Developing a Question Rewrite Model for Disfluency Handling in DISFL-QA Benchmark Dataset

Vinamra Singh
vinamra98@outlook.com

Abstract—**This paper presents the development of a question rewrite model aimed at improving disfluency handling in the DISFL-QA benchmark dataset, a derivative of SQuADv2 that introduces contextual disfluencies to challenge existing Question Answering (QA) systems. We implemented and evaluated sequence-to-sequence models—BART, T5, and Flan-T5—designed to convert disfluent questions into fluent ones. The models were fine-tuned on the DISFL-QA dataset, with Flan-T5 demonstrating the highest performance, achieving BLEU and BERT F1 scores of 90.54 and 0.99, respectively. Our results suggest that the Flan-T5 model is effective for disfluency correction but requires optimization for faster inference. Future work will explore distributed processing techniques to enhance efficiency and further refine hyperparameters.**

Index Terms—**Disfluency, Natural Language Processing, Sequence-to-Sequence Models, Over-fitting, Overconfidence**

I. Introduction

Disfluencies, such as repetitions, corrections, and restarts, are common in spontaneous human speech and present significant challenges for Natural Language Processing (NLP) systems, particularly in tasks like Question Answering (QA) and speech recognition. Despite their prevalence, disfluencies have been an under-studied aspect of NLP, largely due to the lack of large-scale datasets that capture these phenomena. The DISFL-QA benchmark dataset [1], a derivative of the SQuADv2 [2] dataset, was introduced to address this gap. It introduces contextual disfluencies into fluent questions, significantly increasing the complexity of QA tasks. DISFL-QA challenges existing state-of-the-art QA models by requiring a deeper understanding of the context to handle disfluencies effectively. The dataset highlights the critical need for robust NLP models capable of managing the intricacies of natural conversation, as evidenced by the significant performance degradation of QA systems when tested in a zero-shot setting on DISFL-QA.

Previous work on disfluency handling, such as the research conducted on the Switchboard dataset [3], primarily focused on detecting and removing disfluencies from telephonic conversations. The Switchboard dataset, one of the most comprehensive resources for studying spontaneous speech, includes annotations for various disfluency types. While supervised models have been widely used for disfluency detection in Switchboard, recent approaches have explored unsupervised and semi-supervised learning methods, which have shown promise in low-resource settings.

The comparative analysis between DISFL-QA and Switchboard datasets reveals significant differences: DISFL-QA is goal-oriented and designed for QA tasks with a higher prevalence of complex disfluencies like corrections, coreferences [1] and restarts, while Switchboard focuses on general conversational disfluencies with a dominance of repetitions.

A disfluency correction model developed by researchers at IIT Bombay [4], which applies supervised, unsupervised and semi-supervised learning techniques inspired by machine translation and style transfer models. This model achieved BLEU and METEOR scores on the test dataset using three different techniques: unsupervised (79.39 BLEU and 57.25 METEOR), semi-supervised (85.28 BLEU and 58.35 METEOR), and supervised (88.08 BLEU and 59.36 METEOR).

In this work, we implemented sequence-to-sequence models, specifically BART [5], T5 [6], and Flan-T5 [7], inspired by the disfluency correction techniques explored in the DISFL-QA dataset. The decision to use sequence-to-sequence architectures was driven by the nature of the problem, which involves generating fluent questions from disfluent questions. BART was selected due to its strong performance in disfluency correction on the Switchboard dataset, as demonstrated by the IIT Bombay research [4]. T5 was included because it achieved the highest accuracy in the DISFL-QA benchmark [1]. Additionally, Flan-T5 was utilized as it represents an enhanced version of T5, which is expected to provide superior performance. These models were selected based on the literature survey to improve the performance on disfl-qa using LLM's.

II. Experiments

A. Machine Configuration

All models were trained on two Nvidia T4 gpu's (each 16 GB), 56 GB of SSD and 29 GB RAM using kaggle platform.

B. Datasets

The DISFL-QA benchmark dataset [1] is a specialized dataset designed to evaluate the understanding of disfluencies in question answering (QA). Derived from the SQuADv2 [2] dataset, DISFL-QA introduces contextual disfluencies, such as repetitions, corrections, and restarts, into previously fluent questions through human annotation. This makes the dataset significantly more complex than earlier datasets. DISFL-QA

TABLE I
Model Comparisons

Model	Parameters	# Layers	Max tokens	# heads
t5-base	220M	12	512	12
BART-base	140M	12	1024	16
Flan-T5-base	780M	24	512	16

TABLE II
Initial performance of models

Model	Epochs	BLEU score	BERT F1 Score
t5-base	5	90.00	0.98
BART-base	5	89.83	0.98
Flan-T5-base	5	90.54	0.99

TABLE III
Training and Validation Loss, BLEU, and Bert score f1 over Epochs

Epoch	Training Loss	Validation Loss	Bleu	Bert score f1
1	0.831800	0.004827	89.607818	0.9888
2	0.004400	0.004769	89.818564	0.9893
3	0.003100	0.005379	90.199489	0.9900
4	0.002100	0.006171	90.297948	0.9899
5	0.001800	0.007096	90.346927	0.9899

presents several challenges, primarily due to the high prevalence of corrections and restarts, which constitute over 90% of the disfluencies. These types of disfluencies are particularly difficult to detect and correct. Additionally, the dataset includes complex coreferences, further increasing the difficulty level. The dataset comprises over 11,800 annotated questions (train: 7182, dev or val: 1000 and test: 3643), carefully curated to include disfluencies in every instance. It is divided into training, development, and test sets, ensuring that the disfluencies are contextually appropriate.

C. Parameter Configuration

Table I shows the comparison between model parameters used in the experiments.

D. Data Pre-processing

We preprocess the Disfl-QA dataset by extracting and restructuring it into a simplified JSON format that retains only the disfluent-fluent question pairs, omitting the original SQuADv2 dataset's unique IDs. This new structure is more suited for our task, which is better for disfluency correction model rather than question answering model. Subsequently, both the input (disfluent question) and the target (fluent question) are tokenized using the model's tokenizer, preparing the data for further model training and evaluation.

E. Evaluation Metrics

In this study, we employed two evaluation metrics: the BERT F1 score and the BLEU score. The BERT F1 score was chosen due to its ability to capture semantic similarity between the generated and original questions, which is critical for our analysis. The BLEU score was selected based on its established use, drawing inspiration from methodologies presented in [4].

BERT F1 Score. The Bert F1 score [8], leveraging pre-trained BERT [9] embeddings, quantifies semantic similarity between generated and reference texts by evaluating token overlap through cosine similarity. It calculates precision (similarity of generated tokens to reference tokens) and recall (similarity of reference tokens to generated tokens), culminating in an F1 score. This metric indirectly measures model **overconfidence**; overconfident models tend to generate semantically incorrect text, leading to a reduction in the F1 score (specifically reducing precision). Hence, a decline in F1 score may indicate increased overconfidence, providing valuable insights into the model's reliability and accuracy.

BLEU Score. BLEU [10] measures n-gram overlap between generated and reference questions, focusing on word matches rather than intelligibility or grammar. It penalizes unnecessary word repetition and shorter outputs, ensuring generated outputs align closely with the targets based on statistical rules. In this study, we have used SacreBLEU [11] which is similar to BLEU score but better for inter model comparison.

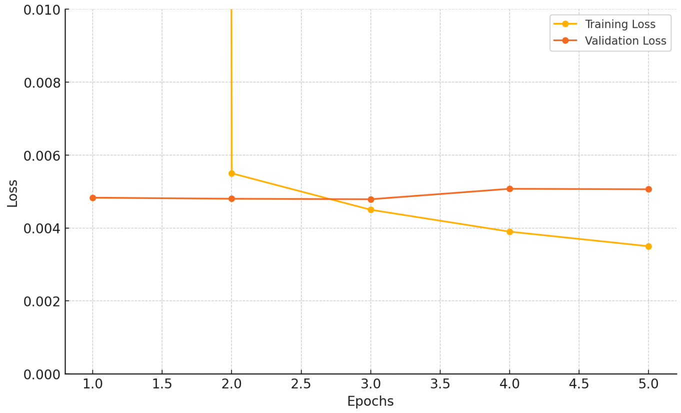

Fig. 1. Training And Validation Loss Over Epochs

III. Experimental Results and Analysis

All experiments were conducted by fine-tuning pre-trained large language models. We begin by fine-tuning three sequence-to-sequence models: T5-base, BART-base, and FlanT5-base, as outlined in Section I. The best performing model is then selected based on the evaluation metrics detailed in Section II-E. As shown in Table II, FlanT5-base demonstrated the highest performance. The next phase involves optimizing its parameters to further enhance the model's performance.

Figure 1 illustrates the performance of the FlanT5 model prior to hyperparameter tuning. After the third epoch, the model begins to overfit, as indicated by increasing validation loss and decreasing training loss. Consequently, *weight_decay* was selected as a parameter for tuning. This parameter, analogous to L2 regularization, is commonly tuned to mitigate **overfitting** in models. Additionally, *learning_rate* was selected for tuning, as it controls the speed of convergence and overfitting, making it a critical factor in improving model performance.

For hyperparameter optimization, we employed the Optuna [12] library, an open-source tool designed for efficient machine learning parameter tuning. In Optuna, a study is created with a trial function, where we define the range of hyperparameter values and the number of trials to be run. Upon completion, the best hyperparameters are selected from all trials. Due to time and computational constraints, only 10% of the training data was used for optimization. Additionally, as the model had already achieved low training and validation loss (see Figure 1), indicating near convergence, using 10% of the data was sufficient and saved considerable time.

After completing hyper-parameter tuning, the optimal values for *weight_decay* and *learning_rate* were applied to fine-tune the final model. Table III presents the training results for each epoch post-tuning. No significant improvement in BLEU and BERT F1 scores was observed during training after hyper-parameter tuning. This is likely due to both the validation and training loss being in the range of 10^{-3} before (Figure 1) and after tuning (Table III), indicating that the model had already converged. Despite this, hyperparameter tuning was pursued to verify our findings and explore any potential performance gains. As per the metrics in Section II-E, the BERT F1 score further confirms that the model is not overconfident, as an F1 score of 0.99 would not be achievable if **overconfidence** were present. Although slight **overfitting** occurs after the second epoch (by comparing training and validation losses in table III), it does not significantly impact validation performance. The high BLEU and BERT F1 scores validate that the model is neither overfit nor overconfident.

As shown in Table III, there is a 9% difference between the BERT F1 and BLEU scores. This discrepancy can be explained by the underlying mechanics of these metrics, as discussed in Section II-E. The BERT F1 score focuses primarily on semantic similarity, giving less weight to factors such as grammatical accuracy and word repetition. In contrast, the BLEU score is a more stringent metric, placing greater emphasis on exact matches between sentences, which results in a lower BLEU score compared to BERT F1.

The model was tested on 10% of the test dataset using the inference code to save time. Due to the model's processing speed, which converts only 26 disfluent questions into fluent questions per minute, for the full test dataset of 3,643 questions, the inference time would be 141 minutes, or approximately 2 hours and 21 minutes. On this subset of test data, the model achieved a BLEU score of 90.26 and a BERT F1 score of 0.99. Given the validation results (Table III) and these test results, it is reasonable to infer that the model's performance on the entire test dataset would be similar to its performance on the 10% subset.

The attention mechanism, visualized using BERT-Viz [13], plays a crucial role in this process by allowing the model to weigh the importance of different words in the input sequence. Figure 2 demonstrates the attention mechanism in selecting the word “countries” in the output based on earlier words such as ”which” and ”I mean,”. The word ”which” introduces the question, while ”I mean” signals the speaker's correction. The model learns to focus on ”I mean” as it signals the actual word of interest (”countries”) after the speaker corrects themselves from ”Norse leader” to ”countries.”

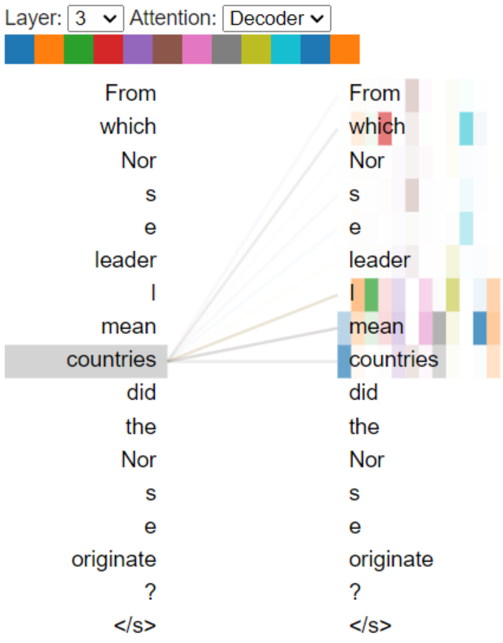
Layer: 3 Attention: Decoder

Fig. 2. Decoder attention head view

IV. Conclusion and Future Work

In this study, we developed and analyzed the question rewrite model which converts disfluent questions to fluent questions using disfl-qa dataset and multiple language models such as T5-base, BART-base and flanT5-base, achieving accuracy comparable to benchmark dataset (Switchboard), as measured by BLEU and BERT F1 Score metrics. However, we observed that model inference times are higher than desired, suggesting the need for distributed mechanisms such as DDP (Distributed Data Parallel) or FSDP (Fully Sharded Data Parallel) to improve efficiency. Also, there is a need of finding more optimum hyperparameters to further improve the models so that small langauge models can be leveraged for faster inference.

References

[1] A. Gupta, J. Xu, S. Upadhyay, D. Yang, and M. Faruqui, “Disfl-qa: A benchmark dataset for understanding disfluencies in question answering,” 2021. [Online]. Available: https://arxiv.org/abs/2106.04016

[2] P. Rajpurkar, R. Jia, and P. Liang, “Know what you don't know: Unanswerable questions for squad,” 2018. [Online]. Available: https://arxiv.org/abs/1806.03822

[3] J. Godfrey, E. Holliman, and J. McDaniel, “Switchboard: telephone speech corpus for research and development,” in *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 1992, pp. 517–520 vol.1.

[4] N. Saini, D. Trivedi, S. Khare, T. Dhamecha, P. Jyothi, S. Bharadwaj, and P. Bhattacharyya, "Disfluency correction using unsupervised and semi-supervised learning," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, P. Merlo, J. Tiedemann, and R. Tsarfaty, Eds. Online: Association for Computational Linguistics, Apr. 2021, pp. 3421–3427. [Online]. Available: https://aclanthology.org/2021.eacl-main.299

[5] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," 2019. [Online]. Available: https://arxiv.org/abs/1910.13461

[6] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2023. [Online]. Available: https://arxiv.org/abs/1910.10683

[7] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei, "Scaling instruction-finetuned language models," 2022. [Online]. Available: https://arxiv.org/abs/2210.11416

[8] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," 2020. [Online]. Available: https://arxiv.org/abs/1904.09675

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1810.04805

[10] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02. USA: Association for Computational Linguistics, 2002, p. 311–318. [Online]. Available: https://doi.org/10.3115/1073083.1073135

[11] M. Post, "A call for clarity in reporting bleu scores," 2018. [Online]. Available: https://arxiv.org/abs/1804.08771

[12] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[13] J. Vig, "A multiscale visualization of attention in the transformer model," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 37–42. [Online]. Available: https://www.aclweb.org/anthology/P19-3007