Does denoising and data augmentation help for rewriting better questions? Fine tuning BART for question rewriting task

Report on
Developing a question rewrite model for the Disfl-QA benchmark dataset.
Author: Sara Binte Zinnat
shazsara20@gmail.com

December 6, 2021

1 Introduction

Natural language processing focuses on giving computers the ability to understand and learn natural languages which are used by humans for communication purpose. A wide variety of tasks form the research dimension of NLP including machine translation, summarization, question answering. In question generation tasks questions are generated based on the input texts or documents. To retrieve the appropriate answers it is worth necessary to emphasize on the fluency of a question. However, due to the typing errors/discontinuity in the thinking process, users may input noisy questions which may provide an irrelevant answer and disrupt the conversational workflow. In order to extracting an accurate question out of noisy inputs, the need of rewriting question occurs. Although majority of the question rewriting tasks focus on conversational question rewriting, the research on rewriting a single and independent fluent question still needs some attention. Gupta et al. [2021] discussed that the lack of data sets containing disfluencies (which refers to the phenomena like repetitions, restarts, or corrections) disrupts some of the NLP tasks. In order to conduct more research on identifying disfluencies in questions, the authors presented a data set Disfl-QA which holds contextual disfluencies in question answering over Wikipedia passage. In this NLP task, I aimed to use this Disfl-QA data set, develop a question rewrite model and analyze whether my developed model is successful to show better performance based on the evaluation metrics. My code is available at `https://github.com/sara-zinnat/question-rewrite`.

2 Background

2.1 Question rewriting task

In DISFL-QA data set each sample contains a disfluent question (Q_d) and it's reference fluent question (Q_r). A sample of the question Q_d and Q_r in DISFL-QA dataset is given in Table 1.

Fluent question	Disfluent question
What is typically used to broadly define complexity measure?	What is defined no is typically used to broadly define complexity measure?

Table 1: Sample question in DISFL-QA dataset

The goal of this Question Rewriting (QR) task is to 1) preprocess the Q_d and Q_r questions, 2) fine tune a pretrained QR model with Q_d and Q_r, 3) predict a fluent question (Q_p) and 4) measure the ROUGE scores between Q_r and Q_p.

2.2 Analysis of the DISFL-QA dataset

In DISFL-QA data set, the authors [Gupta et al., 2021] distributed 11,825 annotated questions into train, dev and test sets where each set holding 7182, 1000 and 3643 questions, respectively. Before selecting any model I prefer to analyze the benchmark data set. Therefore, I computed the number of mismatched words (I termed as noise) that are present in the disfluent question (source) but absent in the fluent question (target). As shown in Figure 1 the mean value of noise is 5.

Again, I investigated the number of new words present in fluent question (target) but absent in disfluent question (source). As shown in Figure 2, the mean of new words is only 1. Without the noise in disfluent question, the words between disfluent and fluent questions are almost same. There are only a few new words that appeared in the fluent question. Therefore, I selected the language model, BART which works better on denoising tasks.

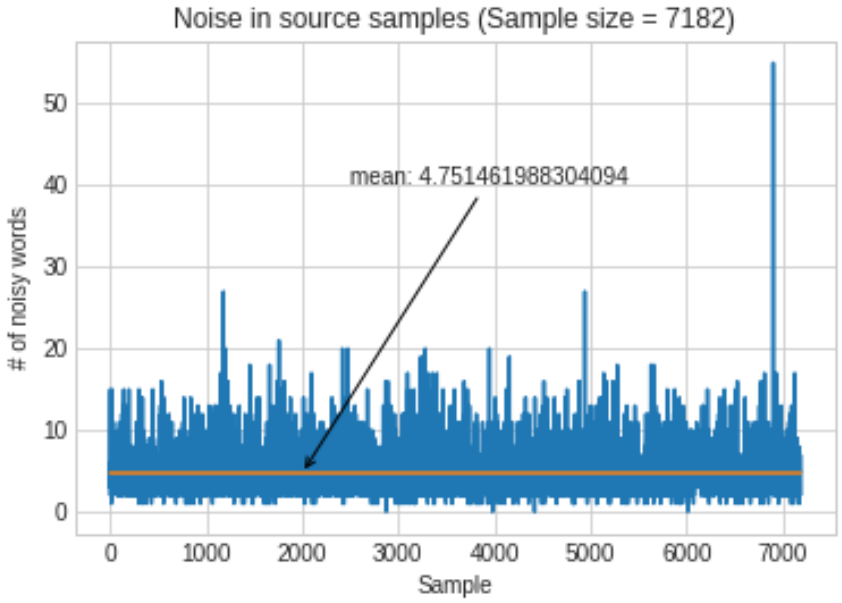

Figure 1: Noise in source (disfluent question) samples.

Analyzing the DISFL-QA dataset, I tried to find out the answers of the following questions: 1) Does denoising a disfluent question help to generate a fluent question? 2) Can data augmentation techniques such as position reordering help for rewriting fluent questions?

2.3 State of the art NLP models

To perform a variety of NLP tasks, the researchers use different language model architectures including Sequence-to-sequence (Seq2seq) model [Sutskever et al., 2014] and Transformer model [Vaswani et al., 2017]. The Seq2seq model is based on Recurrent Neural Network (RNN) concepts such as Long-Short-Term-Memory (LSTM) [Hochreiter and Schmidhuber, 1997] or Gated Recurrent Unit (GRU) [Cho et al., 2014] where the output generated from the previous step is used as input to the present step. There are two components in a Seq2seq model: the encoder and the decoder. The encoder and decoder contain multiple layers of LSTM or GRU models. The Seq2seq model takes a fixed length of input sequence, converts the input to vectors and passed the vectors to the LSTM layers of the encoder. The encoder learns based on the weights and in the final output layer a probability distribution of the entire vocabulary is generated by using softmax activation function. The final output of the encoder is sent to the decoder which uses multiple LSTM layers to generate the output sequence. While generating output the decoder considers the previous outputs and uses softmax function to predict

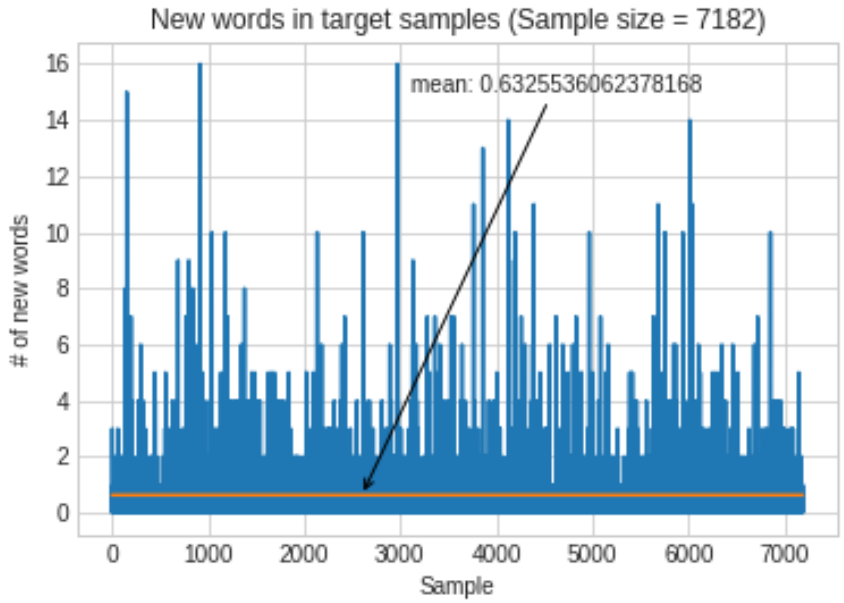

Figure 2: New words in target (reference fluent question) samples.

the final output. Although Seq2seq model is simple, easy to use for smaller data set and requires low memory in a restrictive environment, the performance of the Seq2seq model fall short comparing to the significant performance of the Transformer model.

The Transformer model also has an encoder-decoder structure where the encoder and decoder each has 6 identical layers. Each identical layer is consisted of two sub layers: a multihead attention layer and a feed forward network layer. At each step the multihead attention mechanism or layer considers the input sequence to determine the important parts of the input sequence. The attention layer measures three vectors based on the input which are termed key, query, and value. The dot product of key and query, a scalar, is the relative weighting for a given position. To give every word an attention score, the attention mechanism is applied in parallel at every element in the sequence. Using a softmax the vector representation of the input sequence is then passed to the feed forward layer. Unlike Seq2seq, the Transformer model does not involve any RNN concepts (LSTM, GRU) rather the model uses a feed forward network to remember a relative position of all the words or parts in a sequence.

Again, pretrained models are trained from scratch by using a large amount of data. Thus, training a model from scratch with smaller data set may not result in a good performance. Therefore, for smaller data set it is more appropriate to use the approach of fine-tuning which requires the re-training of a pre-trained model that has already been trained for a given task and makes it perform a second similar task by using the custom data set. Some of the state-of-the-art NLP models are Bidirectional Encoder Representations from Transformers (BERT) [Devlin et al., 2018], Generative Pre-trained Transformer (GPT) [Radford et al., 2019], Pegasus [Zhang et al., 2020], T5 [Raffel et al., 2019], XLNet [Yang et al., 2019].

BERT is based on the concept of the bidirectional training of Transformer. The Transformer encoder in BERT reads the entire sequence of words at once and learn the context of a word based on all of its surroundings (left and right of the word) in all layers. BERT uses two training strategies: 1) Masked Language Model (MLM) where words in each sequence are replaced with a [MASK] token and based on the context the model predicts the original value of the masked words, and 2) Next Sentence Prediction (NSP) where the model receives pairs of sentences as input, and tries to predict whether the second sentence in the input pair is the next sentence in the original document.

Another popular NLP model GPT-2's learning objective is to produce different output for the same input for different tasks, which is referred to as task conditioning. Based on the concept of zero shot learning the model understands the task based on the given instruction. GPT uses multi-layer transformers decoder and fine-tunes the same base model for all end tasks. Transformer block in GPT contains a masked multi-headed self-attention followed by pointwise feed-forward layer and normalization layers in between. GPT is autoregressive which predicts future values based on past values, and uni-directional in nature which is only trained to predict the future left-to-right context.

Other language models such as Pre-training with Extracted Gap-Sentences for Abstractive Summarization (Pegasus) [Zhang et al., 2020] is used for abstractive summarization task whereas T5 is an encoder-decoder model that converts all NLP problems into a text-to-text format and implements the idea of transfer learning.

Therefore, I tried to use such a model that can be fine tuned to denoise the disfluency containing question in DISFL-QA data set. Since BERT and GPT-2 are two powerful state of the art models which are based on Transformer models. Hence, I searched for a model that follows the architecture of Transformer, BERT, GPT and helps to denoise the disfluent question and found out the model BART [Lewis et al., 2020] which pre-trains sequence to sequence models by combining Bidirectional and Auto-Regressive Transformers [Lewis et al., 2020]. Thus, I selected the BART model and planned to fine-tune it by using the given benchmark data set.

3 Related Work

Faruqui and Das [2018] emphasized on generating well formed sentences and created a data set to support question generation task. In a different work, Dong et al. [2017] focused on the learning of paraphrases and implemented a method that can generate paraphrases for different question answering task. Again, Chu et al. [2020] presented a multi-domain question rewriting (MQR) dataset and trained a sequence to sequence model to facilitate question answering task. Some other research works were performed focusing on the conversational question answering and question rewriting tasks [Vakulenko et al., 2021, Gupta et al., 2021]. To smooth the research on question answering and rewriting tasks, different state of the art NLP models have been implemented including BERT, GPT-2, BART which are based on the Transformer architecture [Vaswani et al., 2017, Devlin et al., 2018, Radford et al., 2019, Lewis et al., 2020]. Evaluation metrics such as ROUGE have been widely used to evaluate the performances of the QR models,

4 Experiments

4.1 Experimental Setup

To implement the experiments, I used Google Colab Pro (GCP)[1] and set GPU runtime. I had to mount Google Drive[2] as drive for GCP where the input and our implemented models were saved. I used torch and Transformers libraries packages from Hugging Face [Wolf et al., 2020]. From the transformers library I used BartTokenizer, BartForConditionalGeneration, Trainer, TrainingArguments.

The input sequence in the BART model is preprocessed and tokenized by the BartTokenizer that transformed the input sequence into a vector representation. BartTokenizer use 'facebook/bart-large' pretrained tokenizer to tokenize the input. Here, I used BartTokenizer to use the vocabulary of BART model. This pretrained tokenizer helped me to convert text into vector. To finetune the pretrained model, I used 'facebook/bart-large' for BartForConditionalGeneration library.

To set the training arguments, I used TrainingArguments library. I conducted the fine-tuning for three epochs. Here, training and evaluation batch sizes were set to 16 and 64 respectively. Larger

[1] `https://www.colab.research.google.com/`

[2] `https://drive.google.com/`

batch sizes usually converge faster and provide better performance.

I also set warmup steps as 500, weight decay as 0.01 and learning rate as $5e^{-05}$ to avoid overfitting. Finally, I used the Trainer library to set the trainer for fine-tuning the model. To train the model I used the train and validation dataset. For evaluating the results, I used ROUGE (R1, R2 and RL) [Lin, 2004] metrics.

4.2 My approach

I planned to perform three different experiments using the DISFL-QA dataset. The details of the experiments are as follows,

- **Baseline** - In our first approach, I used the BART model to get the baseline result. I used the test dataset to get the ROUGE scores for the BART model. No training was done here.

  In the second and third approaches, I fine-tuned the BART model with the train and validation dataset.

- **BART-A** - In this approach, I used augmentation on the training dataset so that the word positions in the disfluent questions get attention during training. Since the size of the DISFL-QA data set is small, I used one of the augmentation techniques such as position reordering and reordered the words of the disfluent questions to increase the data set size.

- **BART-B** - In this approach, I didn't use any augmentation technique assuming that the word positions in the reference fluent questions are in the same order as disfluent questions.

4.3 Evaluation Metrics

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) is used as a set of metrics for evaluation of a NLP task. I evaluated the rewrited questions quality by using ROUGE-1, ROUGE-2 and ROUGE-L [Lin, 2004].

5 Results

Table 2 shows the results of the three models on DISFL-QA dataset. From the table we can see that the Baseline model showed the lowest performance comparing to the other two models. Alternatively, BART-B model showed the highest performance. BART-A where I applied augmentation also showed a better performance. Based on the results we can see that the approach of fine-tuning the BART model was successful to remove the noise from the disfluent questions of the benchmark data set.

Model	ROUGE-1			ROUGE-2			ROUGE-L		
	R	**P**	**F1**	**R**	**P**	**F1**	**R**	**P**	**F1**
Baseline	49.92	51.60	49.70	41.35	41.84	40.64	49.07	50.72	48.85
BART-A	84.73	89.96	86.42	79.27	84.59	80.81	84.2	89.38	85.88
BART-B	**85.45**	**92.36**	**88.12**	**80.92**	**88.17**	**83.63**	**85.01**	**91.86**	**87.66**

Table 2: ROUGE scores on the DISFL-QA data set from baseline model (BART) and proposed methods (BART-A and BART-B).

6 Discussion

BART model is broadly used for summarization, machine translation and question answering tasks. Since these tasks use multiple sentences during training phase, this is may be a possible reason that the baseline or pretrained BART model showed the lowest performance. In contrast, the fine tuned

models BART-A and BART-B both outperformed the baseline model. While fine tuning, the BART model learned to generate single sentence or question which might helped BART-A and BART-B to achieve better performance. Again, analyzing the benchmark data set we found that the order or position of the words between the disfluent questions and reference fluent questions was almost similar. Therefore, the reordering of the words in the disfluent questions was not required for the given data set. This is may be a possible reason that BART-B model performed best and BART-A model could not outperformed BART-B model. It is worth noting that, the model selection and fine tuning approach was planned by analyzing the given benchmark data set. If the BART-B model is tested on a different data set, the model may require similar type of data to show such significant performance.

7 Conclusion and future works

In this research my goal was to identify noise in disfluent question and rewrite fluent questions by fine tuning BART model on the benchmark DISFL-QA dataset. Analyzing the experimental results we found that my developed models have showed significant performance to rewrite questions. The future work may include a syntactic analysis of the questions to obtain more accurate result. If the attention matrix can be adjusted with discourse matrix there lies a scope of improving the performance.

References

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Zewei Chu, Mingda Chen, Jing Chen, Miaosen Wang, Kevin Gimpel, Manaal Faruqui, and Xiance Si. How to ask better questions? a large-scale multi-domain dataset for rewriting ill-formed questions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34(05), pages 7586–7593, 2020.

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

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886, 2017.

Manaal Faruqui and Dipanjan Das. Identifying well-formed natural language questions. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 798–803, 2018.

Aditya Gupta, Jiacheng Xu, Shyam Upadhyay, Diyi Yang, and Manaal Faruqui. Disfl-QA: A Benchmark Dataset for Understanding Disfluencies in Question Answering. In *Findings of ACL*, 2021.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 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*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL `https://aclanthology.org/2020.acl-main.703`.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

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. *arXiv preprint arXiv:1910.10683*, 2019.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. Question rewriting for conversational question answering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 355–363, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL `https://aclanthology.org/2020.emnlp-demos.6`.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR, 2020.