Reversed-order Text Generation

Haya Riesel haya.riesel
Asher Guedalia asher.guedalia
Matanel Oren matanel.oren
Oshri Avnery oshri-avitov.avnery

1 Introduction

Left-to-right text generation is probably one of the most fundamental concepts when approaching text generation tasks. It has been used even before the Transformers era, and is the basis for defining the language modeling task. It is surly the most popular these days, since all well-known generation LLMs, such as T5 (Raffel et al., 2020), GPT (Alec Radford, 2018), Llama (Touvron et al., 2023), etc. are trained and evaluated so.

We have found that only a few studies (Nguyen et al., 2023) (Bavarian et al., 2022) asked the question whether there is a better way, in terms of quality and efficiency, to generate text in different scenarios. These studies mainly include methods to generate the text from both sides simultaneously, so as to achieve twice faster generation. But, we have found no work that compares the left-to-right (or forward) generation, to a right-to-left (backward or reversed-order) one.

Although the classic left-to-right generation is straight forward (even literally..), and similar to the way that human-beings generate speech, it might not be the best option for text generation for scenarios where the main idea appears in the end of the text, and especially for languages that put the more important parts later in the sentence.

We find it interesting to compare, in the most limited way possible, these two generation approaches, and perhaps open room for future research to challenge the classical methods, or find the best scenario-approach match (e.g. tasks, datasets or languages).

2 Data

As will be described in the next section, we look for scenarios where the model's input and output are completely separated, therefore we chose the text-summarization task.

We used two primary text summarization datasets: CNN/Daily Mail (See et al., 2017) and XSum (Narayan et al., 2018). CNN/Daily Mail has news articles with human-written summaries, and XSum has various online articles. Both datasets are well known resources for developing and evaluating summarization models.

	Train	Validation	Test
CNN/Daily Mail	287,113	13,368	11,490
XSum	204,045	11,332	11,334

Table 1: The datasets used for the reaserch.

3 Methods

3.1 Encoder-decoder architecture

Text-generation models are roughly typed by two architecture families: encoder-decoder (BART, T5) and decoder-only (GPT, Llama). In encoder-decoders the input text is fed into the encoder, and the decoder generates the output text, while in decoder-only models the input text is fed as the beginning of a continuous text to the decoder, and is continued by it.

We do not find it useful to just generate text from its ending, so we decided to use the encoder-decoder architecture where the input text is fed into the model as-is, and the model could be trained to generate the **output only** right-to-left. In this work we use BART (Lewis et al., 2019) architecture and code.

As a result, we prefer a task where the input and output are not continuous but separated, so we chose text-summarization, as it is one of the most popular and well defined yet challenging tasks in the world of text-generation.

3.2 Warm-start initialization

When approaching the study, we had to deal with the following problem: all of the pretrained models were pretrained using left-to-right, and hence are inherently biased towards this method. The

comparison to models finetuned for reversed-order generation is unfair. On the other hand, pretraining 'from scratch' takes much more time and resources, that we can't afford in the limited frame of this project.

To overcome this challenge, we decided to use warm-start for initializing the model's weights, as in Rothe et al. (2020). This method allows us to use encoder-only model's weights, which has no preference for a one generation direction over the other, to initialize the new model's weights. Furthermore, we found out that no further pretraining is needed to achieve good results on downstream generation tasks, and about 100K of finetuning steps is enough to compete an ordinary finetuned model.

The RoBERTa-base (Liu et al., 2019) architecture and size is quite similar to each part of BART-large, Therefore, it is relatively easy to transform the first's weights to a new model, herein the decoder and the encoder begin with RoBERTa weights. It's important to emphasize that we don't use shared weights method. Note that RoBERTa-base has 12 layer and 12 heads per layer, while BART-large has 12 layers for each part, 16 heads per layer, so the resulting model is a little smaller. In addition, we use RoBERTa tokenizer, as the weights were pretrained on it.

To reduce the effort of adjusting the model's architecture to right-to-left text generation, we leave the model's code as-is and just reverse the label tokens before feeding into the model.

3.2.1 Position embeddings initialization

We noticed that the initialization of the learned position embeddings is more tricky, and decided to add it as a hyper-parameter. We tried four options:

- Copy both encoder's and decoder's from RoBERTa.
- Copy dencoder's and use the reversed decoder's.
- Copy encoder's from RoBERTa and randomly initialize the decoder's.
- Initialize all randomly.

3.3 Experiment setup

We train the model for 3 epochs with default hyper-parameters. For each of the datasets, we initialize the position embeddings using the above 4 options, and train different models to generate summaries left-to-right and right-to-left. Overall 16 runs.

4 Results

Despite the low expectations, the results are pretty good. The reversed model generates grammatically and syntactically correct sentences, containing relevant information from the reference. We show two examples in Table 3.

We used ROUGE metric to evaluate the models. For this metric, we found a noticeable difference between the models performance on the 2 datasets.

For CNN/Dailymail, even though the loss is similar, ROUGE metrics show a consistent gap in favor of the forward models, over configurations and training process. For example, ROUGE-2 curves on the evaluation split during training are shown in Fig. 1. A comparison of the resulting model and a regular finetuned BART is presented in Table 2. The difference is even visible to a human reader, since the generated text is more repetitive and sometimes talks out of context.

For XSum, the results are surprisingly great. The reversed model achieves similar ROUGE to the forward model, and even better for some, as shown in Table 4, and ROUGE-2 curves on the evaluation split are in Fig. 2. To our eyes the reversed outputs really looks better.

However, all of our experiments do not show competitive results to the finetuned BART-large. This might be due to the fact that XSum is smaller, with shorter summaries, such that a model that has not been taught to generate text does not have enough training on such tasks. Note that the model's weights are initialized from pretrained RoBERTa which is trained on MLM and not next token prediction.

One more worth mentioning phenomenon is that the position embeddings initialization really matters. The different initializations imply different results, and hierarchy among them is not consistent across the datasets, meaning, a different initialization was better for each dataset. Nonetheless, for each of the datasets one init configuration occurs that gets stuck in a local minimum, it starts with generating the end-token first, then during evaluation to generate a blank summary, although the loss continues to go down.

We wonder what the cause for the difference between the two datasets is. Our first guess is that reversed models are better in generating shorter texts, but we leave that to future research. We will also be interested to see the effect of the generation direction for other languages, or at least other tasks.

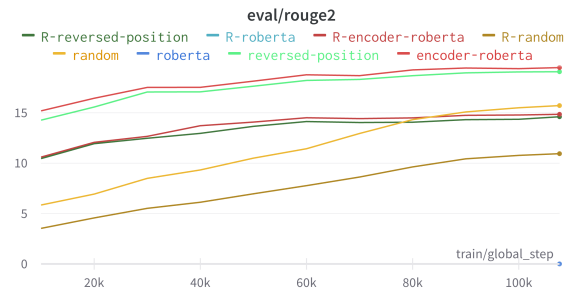

Figure 1: Rouge 2 over the training process on CNN/Dailymail. 'R' stands for a reversed model. Contains the different position embedding initializations, as described in Section 3.2.1

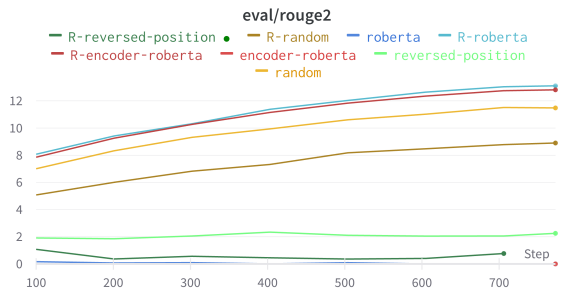

Figure 2: Rouge 2 over the training process on XSum. 'R' stands for a reversed model. Contains the different position embedding initializations, as described in Section 3.2.1

	Loss	Rouge1	Rouge2	RougeLsum
Bart-large-cnn	2.529	42.949	20.815	40.038
Forward-best	1.8294	41.6829	18.9191	39.1257
Reverse-best	1.8824	33.3323	14.4118	31.4998

Table 2: CNN/DailyMail prediction metrics on the finetuned models as described in Table 3

	Loss	Rouge1	Rouge2	RougeLsum
Bart-large-cnn	2.313	45.453	22.346	37.232
Forward-best	2.234	33.002	11.482	26.162
Reverse-best	2.011	32.282	13.112	26.697

Table 4: XSum prediction metrics on the finetuned models as described in Table 3

	Gold	Bart-large-cnn	Forward-best	Reverse-best
CNN/Daily Mail	Once a super typhoon, Maysak is now a tropical storm with 70 mph winds . It could still cause flooding, landslides and other problems in the Philippines.	Maysak g ained super typhoon status just a few days ago. It has since lost steam as it spins west in the Pacific Ocean. It boasts steady winds of..	Maysak gained super typhoon status a few days ago. It's now classified as a tropical storm. Authorities take preemptive steps to keep people safe.	Maysak has steady winds of more than 70 mph (115 kph) as of 5 p.m. (5 a.m. ET) Saturday. It's expected to make landfall Sunday and be out of the Philippines by Monday.
XSum	The pancreas can be triggered to regenerate itself through a type of fasting diet, say US researchers.	A "fasting-mimicking" diet can regenerate a special type of cell in the pancreas, a study in mice suggests.	A vegan diet can reverse type-2 diabetes, a study suggests.	A healthy diet can restore cells in a person's pancreas, a study suggests.

Table 3: Examples of generated text from samples in the test set for each dataset using different models: Bart-large-cnn (BART fine-tuned on the dataset), Forward-best (Best BART with RoBERTa weights fine-tuned on the dataset), and Reverse-best (Best BART with RoBERTa weights fine-tuned on the dataset with reversed labels).

A Github Project

Our code can be found here: https://github.com/ml-oren/reversed-text-gen.

References

Tim Salimans Ilya Sutskever Alec Radford, Karthik Narasimhan. 2018. Improving language understanding by generative pre-training. *OpenAI Blog*.

Mohammad Bavarian, Heewoo Jun, Nikolas Tezak, John Schulman, Christine McLeavey, Jerry Tworek, and Mark Chen. 2022. Efficient training of language models to fill in the middle.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

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

Anh Nguyen, Nikos Karampatziakis, and Weizhu Chen. 2023. Meet in the middle: A new pre-training paradigm. *arXiv preprint arXiv:2303.07295*.

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.

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.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

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. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.