**BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension**

Bidirectional Auto-Regressive Transformer(BART), was developed by facebook AI which uses a new pretrained model for text generation and comprehension that uses both bidirectional and auto-regressive methods and a standard Transformer-based neural machine translation architecture.

**Keywords:** BART,NLP, Encoder, Decoder, Transformers, Machine Learning, Neural Networks, Deep Learning.

**Prerequisite:** Basic understanding of [Transformer](https://en.wikipedia.org/wiki/Transformer_(machine_learning_model)#:~:text=The%20Transformer%20is%20a%20deep,as%20translation%20and%20text%20summarization.) **,** [BERT](https://arxiv.org/abs/1810.04805)and [GPT](https://arxiv.org/abs/1810.04805) model.

**Core Idea:** [BART](https://arxiv.org/pdf/1910.13461.pdf), a denoising autoencoder for pretraining sequence-to-sequence models. BART is trained by (1) corrupting text with an arbitrary noise function, and (2) learning a model to reconstruct the original text. It uses a standard Transformer-based neural machine translation architecture which, despite its simplicity, can be seen as generalizing BERT (due to the bidirectional encoder), GPT (with the left-to-right decoder).

**Key Features**

1. **BART** matches the performance of RoBERTa on GLUE and SQuAD, and achieves new state-of-the-art results on a range of abstractive di-alogue, question answering, and summarization tasks, with gains of up to 3.5 ROUGE.
2. **BART** also provides a 1.1 BLEU increase over a back-translation system for machine translation, with only target language pretraining.

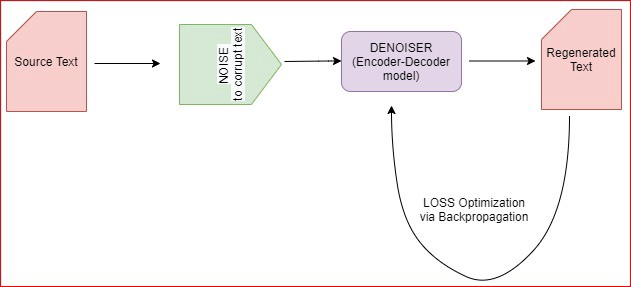
**1 Introduction**

BART is a **denoising autoencoder** that maps a corrupted document to the original document it was derived from. It is implemented as a sequence-to-sequence model with a bidirectional encoder over corrupted text and a left-to-right autoregressive decoder. For pre-training, we optimize the negative log likelihood of the original document.

**2 Background**

**2.1 Architecture**

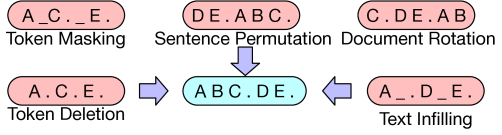
BART uses the standard sequence-to-sequence Transformer architecture from except, following GPT, that we modify ReLU activation functions to GeLUs and initialise parameters from N (0, 0.02). For our base model, we use 6 layers in the encoder and decoder, and for our large model we use 12 layers in each. The architecture is closely related to that used in BERT, with the following differences: (1) each layer of the decoder additionally performs cross-attention over the final hidden layer of the encoder (as in the transformer sequence-to-sequence model); and (2) BERT uses an additional feed-forward network before wordprediction, which BART does not. In total, BART contains roughly 10% more parameters than the equivalently sized BERT model.



it is a transformer based Seq2Seq model that uses corrupted source text and then tries to denoise the source text by regenerating the original text from the decoder and each layer of the decoder attends to the final hidden layer of encoder. It can be seen as a Seq2Seq model modified to work as an auto-encoder. A notable feature in the architecture is the use of GELU instead of RELU activation layer. When compared to BERT, it doesn’t make use of a feed-forward network at the top for word prediction while BERT does.

**2.2** **Pre-training BART**

BART is pre-trained by minimizing the cross-entropy loss between the decoder output and the original sequence.To prepare model for pre-training, firstly, some tokens from the input/source text are corrupted randomly (addition of noise schemes) and while training the regeneration loss is optimized using cross-entropy loss between output and the decoder’s output. Unlike existing denoising auto-encoders, which are tailored to specific noising schemes, BART allows us to apply any type of document corruption. In the extreme case, where all information about the source is lost, BART is equivalent to a language model. Let us also have a look over the noise schemes/transformations that can be used in BART over the source text:



**1.** **Token Masking** **:** Random tokens are sampled and are masked with [MASK] tokens.

**2**. **Token Deletion :** Random tokens are sampled and deleted (similar as masking) and the model adds a new token in their place.

**3.** **Token Infilling** **:** A number of text spans (group of contiguous tokens) is drawn from Poisson’s distribution and each span is replaced by a masked token [MASK].

**4**. **Sentence Permutation :** Random shuffling of document’s sentences.

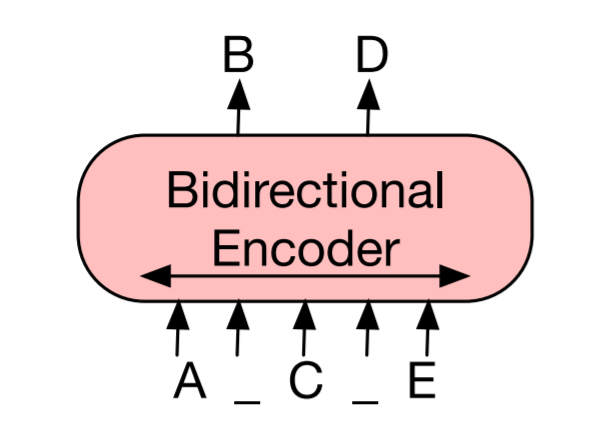
**5**. **Document Rotation** **:** A token is uniformly chosen at random and the document is rotated about that token so that the document begins with that token.

## **Masked Language Modeling (MLM)**

MLM models such as BERT are pre-trained to predict masked tokens. This process can be broken down as follows:

1. Replace a random subset of the input with a *mask token [MASK].* (Adding noise/corruption)
2. The model predicts the original tokens for each of the [MASK]tokens. (Denoising

Importantly, BERT models can “see” the full input sequence (with some tokens replaced with [MASK]) when attempting to predict the original tokens. This makes BERT a bidirectional model, i.e. it can “see” the tokens before and after the masked tokens.



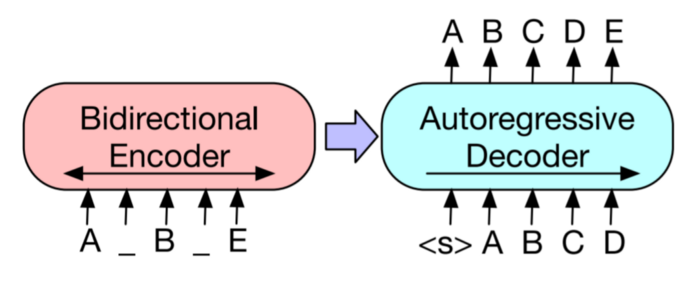
## **Autoregressive Models**

Models used for text generation, such as GPT2, are pre-trained to predict the next token given the previous sequence of tokens. This pre-training objective results in models that are well-suited for text generation, but not for tasks like classification.

## 

## **BART Sequence-to-Sequence**

BART has both an encoder (like BERT) and a decoder (like GPT), essentially getting the best of both worlds.The encoder uses a *denoising* objective similar to BERT while the decoder attempts to reproduce the original sequence *(autoencoder)*, token by token, using the previous (uncorrupted) tokens and the output from the encoder.



**3 Fine-tuning BART**

**3.1 Sequence Classification Tasks**

For sequence classification tasks, the same input is fed into the encoder and decoder, and the final hidden state of the final decoder token is fed into new multi-class linear classifier. This approach is related to the CLS token in BERT; however we add the additional token to the end so that representation for the token in the decoder can attend to decoder states from the complete input.

**3.2 Token Classification Tasks**

For token classification tasks, such as answer endpoint classification for SQuAD, we feed the complete document into the encoder and decoder, and use the top hidden state of the decoder as a representation for each word. This representation is used to classify the token.

**3.3 Sequence Generation Tasks**

Because BART has an autoregressive decoder, it can be directly fine tuned for sequence generation tasks such as abstractive question answering and summarization. In both of these tasks, information is copied from the input but manipulated, which is closely related to the denoising pre-training objective. Here, the encoder input is the input sequence, and the decoder generates outputs autoregressive.

**4 Experiment**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | MNLI  m/mm | SST  Acc | QQP  Acc | QNLI  Acc | STS-B  Acc | RTE  Acc | MRPC  Acc | CoLA  Mcc |
| BERT | 86.6/- | 93.2 | 91.3 | 92.3 | 90.0 | 70.4 | 88.0 | 60.6 |
| UniLM | 87.0/85.9 | 94.5 | - | 92.7 | - | 70.9 | - | 61.1 |
| XLNet | 89.8/- | 95.6 | 91.8 | 93.9 | 91.8 | 83.8 | 89.2 | 63.6 |
| RoBERTa | **90.2/90.2** | 96.4 | 92.2 | 94.7 | **92.4** | 86.6 | **90.9** | **68.0** |
| BART | 89.9/90.1 | **96.6** | **92.5** | **94.9** | 91.2 | **87.0** | 90.4 | 62.8 |

Table 1: Results for large models on GLUE tasks. BART performs comparably to RoBERTa and XLNet, suggesting that BART’s uni-directional decoder layers do not reduce performance on discriminative tasks.

**5 Qualitative Analysis**

BART shows large improvements on summarization metrics, of up to 3.5 points over the prior state-of-theart. To understand BART’s performance beyond automated metrics, we analyse its generations qualitatively.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | ELI5  R1 | ELI5  R2 | ELI5  RL |
| Best Extractive | 23.5 | 3.1 | 17.5 |
| Language Model | 27.8 | 4.7 | 23.1 |
| Seq2Seq | 28.3 | 5.1 | 22.8 |
| Seq2Seq Multitask | 28.9 | 5.4 | 23.1 |
| BART | **30.6** | **6.2** | **24.3** |

Table 2: BART achieves state-of-the-art results on the challenging ELI5 abstractive question answering dataset.

**6 Conclusions**

We introduced BART, a pre-training approach that learns to map corrupted documents to the original. BART performs comparably to RoBERTa on discriminative tasks, and achieves new state-of-the-art results on several text generation tasks. Future work should explore new methods for corrupting documents for pretraining, perhaps tailoring them to specific end tasks.

**7 References**

[**https://medium.com/analytics-vidhya/revealing-bart-a-denoising-objective-for-pretraining-c6e8f8009564**](https://medium.com/analytics-vidhya/revealing-bart-a-denoising-objective-for-pretraining-c6e8f8009564)

[**https://towardsdatascience.com/bart-for-paraphrasing-with-simple-transformers-7c9ea3dfdd8c**](https://towardsdatascience.com/bart-for-paraphrasing-with-simple-transformers-7c9ea3dfdd8c)

[**BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension | Facebook AI Research**](https://ai.facebook.com/research/publications/bart-denoising-sequence-to-sequence-pre-training-for-natural-language-generation-translation-and-comprehension/)

[**https://scontent.fbho4-2.fna.fbcdn.net/v/t39.8562-6/106373513\_3414102562251474\_8005430471454563564\_n.pdf?\_nc\_cat=105&ccb=2&\_nc\_sid=ae5e01&\_nc\_ohc=bCHxFP\_ATJsAX9aBeXm&\_nc\_ht=scontent.fbho4-2.fna&oh=fd08cff2d73197f3704036734bcc390d&oe=5FEB5704**](https://scontent.fbho4-2.fna.fbcdn.net/v/t39.8562-6/106373513_3414102562251474_8005430471454563564_n.pdf?_nc_cat=105&ccb=2&_nc_sid=ae5e01&_nc_ohc=bCHxFP_ATJsAX9aBeXm&_nc_ht=scontent.fbho4-2.fna&oh=fd08cff2d73197f3704036734bcc390d&oe=5FEB5704)