**BART**

<https://www.geeksforgeeks.org/bart-model-for-text-auto-completion-in-nlp/>

BART stands for Bidirectional and Auto-Regressive Transformer.

It is a *denoising autoencoder* that is a pre-trained sequence-to-sequence method, that uses *masked language modelling* for Natural Language Generation and Translation.

It was developed by Lewis et al. in 2019.

BART architecture is similar to an encoder-decoder network except that it uses a combination of similar architecture used in BERT and GPT models.

The BART models can be fine-tuned over small supervised datasets to create domain-specific tasks.

**Denoising autoencoder:**

An autoencoder is a special type of neural network that learns to encode an input sentence into lower dimensional representations and decode the embedded representations back to the corresponding original input sentences.

In a general case, when the input and output sentence of an autoencoder is the same, over a large number of iterations, the autoencoder network directly maps the input token to the output tokens, and the embedded representation that is usually learned between them becomes redundant.

Therefore, we modify the input sentence by randomly deleting word tokens and replacing them with a special <MASK> token, this sentence with the randomly deleted token is called a corrupted or noisy sentence and the supervised output for the corresponding input is the clean sentence with all the original tokens preserved.

By learning to predict the missing or corrupted tokens, the denoising autoencoder learns to extract meaningful features from the input sentence.

A denoising autoencoder is trained on a large corpus of such data so it learns to predict the masked/deleted token in the input sentence which is responsible for the noise in the text, as a result, we get a clean and semantically coherent output, hence the term “denoising” is added to the autoencoder.

**Architecture**

For a given input text sequence, the BERT (Bidirectional Representation for Transformers) encoder network generates an embedding for each token in the input text and an additional sentence-level embedding vector.

The GPT decoder network learns this token-level and sentence-level embedded information and its existing pre-trained weights to generate clean semantically close text sequences.

BART has approximately 140 million parameters which are greater than BERT (110 million parameters) and GPT-1 (117 million) models but outperform them significantly given that BART is a combination of them both.

BART’s primary task is used to generate clean semantically coherent text from corrupted text data but it can also be used for a variety of different NLP sub-tasks like language translation, question-answering tasks, text summarization, paraphrasing, etc.

As BART is an autoencoder model, it consists of an encoder model and a decoder model. For its encoder model, BART uses a bi-directional encoder that is used in BERT, and for its decoder mode, it uses an autoregressive decoder that forms the core aspect of a GPT -1 model.

An autoregressive decoder is a neural network architecture that takes the previous input tokens as well as the current token to predict the next token at every time step. It is important to remember that the input accepted by a decoder is an embedding created by its corresponding encoder network.

It consists of 3 primary blocks:

* Multi-head Attention block
* Addition and Normalization block
* Feed-forward layers

**Multi-head attention block**

This is one of the most important blocks as in this layer multiple levels of masking are performed over the predicting tokens, for example:

* Parallel #1 thread: Entire sentence is replaced by the <MASK> tokens.
* Parallel #2 thread: Multiple bi-gram tokens are replaced by the <MASK> tokens.
* Parallel #3+ thread: Arbitrary words within the sentence are replaced by the <MASK> token.

This masking is done in parallel instead of sequentially to avoid accumulating previous step errors for the same input sentence.

**BART Hugging Face**

<https://huggingface.co/docs/transformers/en/model_doc/bart>

**Overview**

The Bart model was proposed in BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension by Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov and Luke Zettlemoyer on 29 Oct, 2019.

**Abstract**

Bart uses a standard seq2seq/machine translation architecture with a bidirectional encoder (like BERT) and a left-to-right decoder (like GPT).

BART is particularly effective when fine-tuned for text generation but also works well for comprehension tasks.

It matches the performance of *RoBERTa* with comparable training resources on *GLUE* and *SQuAD*, achieves new state-of-the-art results on a range of abstractive dialogue, question answering, and summarization tasks, with gains of up to 6 *ROUGE*.

**Usage tips:**

BART is a model with absolute position embeddings so it’s usually advised to pad the inputs on the right rather than the left.

Sequence-to-sequence model with an encoder and a decoder. Encoder is fed a corrupted version of the tokens, decoder is fed the original tokens (but has a mask to hide the future words like a regular transformers decoder).

A composition of the following transformations are applied on the pretraining tasks for the encoder:

* mask random tokens (like in BERT)
* delete random tokens
* mask a span of k tokens with a single mask token (a span of 0 tokens is an insertion of a mask token)
* permute sentences
* rotate the document to make it start at a specific token

**SQuAD and GLUE Benchmarks**

<https://www.techtarget.com/searchenterpriseai/feature/What-do-NLP-benchmarks-like-GLUE-and-SQuAD-mean-for-developers>

**SQuAD (2016)**

The Stanford Question Answering Data set v1.1 is a collection of 100,000 crowdsourced question/answer pairs drawn from Wikipedia.

SQuAD2.0, introduced in 2018, and builds on this with 50,000 unanswerable questions designed to look like answerable ones. To perform well, the NLP model must determine when the correct answer is not available.

**GLUE (2018)**

The General Language Understanding Evaluation (GLUE) benchmark is a collection of nine different language understanding tasks.

**BART Hugging Face Models**

Model: facebook/bart-large

Tensorflow: Yes

<https://huggingface.co/facebook/bart-large>

Model: facebook/bart-large-cnn

Tensorflow: Yes

<https://huggingface.co/facebook/bart-large-cnn>

Model: Azma-AI/bart-large-text-summarizer

Tensorflow: Yes

<https://huggingface.co/Azma-AI/bart-large-text-summarizer>

Model: facebook/bart-large-xsum

Tensorflow: Yes

<https://huggingface.co/facebook/bart-large-xsum>

Model: philschmid/bart-large-cnn-samsum

Tensorflow: No

<https://huggingface.co/philschmid/bart-large-cnn-samsum>

**Hugging Face Datasets**

Dataset: abisee/cnn\_dailymail

Task: Summarization

Sub-tasks: New article summarization

<https://huggingface.co/datasets/abisee/cnn_dailymail>

Dataset: Samsung/samsum

Task: Summarization

Sub-tasks: Nil

<https://huggingface.co/datasets/Samsung/samsum>

Dataset: McAuley-Lab/Amazon-Reviews-2023

Task: Nil

Sub-tasks: Nil

<https://huggingface.co/datasets/McAuley-Lab/Amazon-Reviews-2023>

Dataset: rkf2778/amazon\_reviews\_mobile\_electronics

Task: Text Classification

Sub-tasks: Nil

<https://huggingface.co/datasets/McAuley-Lab/Amazon-Reviews-2023>

This dataset is not useful for summarization because the review title is missing which primarily works as our summary.

**Hugging Face Summarization (NLP Course)**

<https://huggingface.co/learn/nlp-course/chapter7/5?fw=tf>