BART for Paraphrasing with Simple Transformers

Paraphrasing is the act of expressing something using different words while retaining the original meaning. Let's see how we can do it with BART, a Sequence-to-Sequence Transformer Model.

Thilina Rajapakse

Thilina Rajapakse

Aug 5, 2020·8 min read

Photo by Alexandra on Unsplash

Introduction

BART is a denoising autoencoder for pretraining sequence-to-sequence models. BART is trained by (1) corrupting text with an arbitrary noising function, and (2) learning a model to reconstruct the original text.

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

Don't worry if that sounds a little complicated; we are going to break it down and see what it all means. To add a little bit of background before we dive into BART, it's time for the now-customary ode to Transfer Learning with self-supervised models. It's been said many times over the past couple of years, but Transformers really have achieved incredible success in a wide variety of Natural Language Processing (NLP) tasks.

BART uses a standard Transformer architecture (Encoder-Decoder) like the original Transformer model used for neural machine translation but also incorporates some changes from BERT (only uses the encoder) and GPT (only uses the decoder). You can refer to the 2.1 Architecture section of the BART paper for more details.

Pre-Training BART

BART is pre-trained by minimizing the cross-entropy loss between the decoder output and the original sequence.

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.

Figure 1 (a) from the BART paper

This is suited for tasks like classification where you can use information from the full sequence to perform the prediction. However, it is less suited for text generation tasks where the prediction depends only on the previous words.

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.

Figure 1 (b) from the BART paper

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.

Figure 1 (c) from the BART paper

A significant advantage of this setup is the unlimited flexibility of choosing the corruption scheme; including changing the length of the original input. Or, in fancier terms, the text can be corrupted with an arbitrary noising function.

The corruption schemes used in the paper are summarized below.

1. Token Masking — A random subset of the input is replaced with [MASK] tokens, like in BERT.
2. Token Deletion — Random tokens are deleted from the input. The model must decide which positions are missing (as the tokens are simply deleted and not replaced with anything else).
3. Text Infilling — A number of text spans (length can vary) are each replaced with a single [MASK] token.
4. Sentence Permutation — The input is split based on periods (.), and the sentences are shuffled.
5. Document Rotation — A token is chosen at random, and the sequence is rotated so that it starts with the chosen token.

The authors note that training BART with text infilling yields the most consistently strong performance across many tasks.

For the task we are interested in, namely paraphrasing, the pre-trained BART model can be fine-tuned directly using the input sequence (original phrase) and the target sequence (paraphrased sentence) as a Sequence-to-Sequence model.

This also works for tasks like summarization and abstractive question answering.

Setup

We will use the Simple Transformers library, based on the Hugging Face Transformers library, to train the models.

1. Install Anaconda or Miniconda Package Manager from here.

2. Create a new virtual environment and install packages.

```
conda create -n st python pandas tqdmconda activate st
```

3. If using CUDA:

```
conda install pytorch>=1.6 cudatoolkit=10.2 -c pytorch
```

else:

```
conda install pytorch cpuonly -c pytorch
```

4. Install simpletransformers.

```
pip install simpletransformers
```

Data Preparation

We will be combining three datasets to serve as training data for our BART Paraphrasing Model.

1. Google PAWS-Wiki Labeled (Final)
2. Quora Question Pairs Dataset
3. Microsoft Research Paraphrase Corpus (MSRP)

The bash script below can be used to easily download and prep the first two datasets, but the MSRP dataset has to be downloaded manually from the link. (Microsoft hasn't provided a direct link 😔)

Make sure you place the files in the same directory (data) to avoid annoyances with file paths in the example code.

We also have a couple of helper functions, one to load data, and one to clean unnecessary spaces in the training data. Both of these functions are defined in utils.py.

Some of the data have spaces before punctuation marks that we need to remove. clean_unnecessary_spaces() function is used for this purpose.

Paraphrasing with BART

Once the data is prepared, training the model is quite simple.

Note that you can find all the code in the Simple Transformers examples here.

First, we import all the necessary stuff and set up logging.

Next, we load the datasets.

Then, we set up the model and hyperparameter values. Note that we are using the pre-trained facebook/bart-large model, and fine-tuning it on our own dataset.

Finally, we'll generate paraphrases for each of the sentences in the test data.

This will write the predictions to the predictions directory.

Hyperparameters

The hyperparameter values are set to general, sensible values without doing hyperparameter optimization. For this task, the ground truth does not represent the only possible correct answer (nor is it necessarily the best answer). Because of this, tuning the hyperparameters to nudge the generated text as close to the ground truth as possible doesn't make much sense.

Our aim is to generate good paraphrased sequences rather than to produce the exact paraphrased sequence from the dataset.

If you are interested in Hyperparameter Optimization with Simple Transformers (particularly useful with other models/tasks like classification), do check out my guide here.

Hyperparameter Optimization for Optimum Transformer Models

How to tune your hyperparameters with Simple Transformers for better Natural Langauge Processing.

towardsdatascience.com

The decoding algorithm (and the relevant hyperparameters) used has a considerable impact on the quality and nature of the generated text. The values I've chosen (shown below) are generally suited to produce "natural" text.

For more information, please refer to the excellent Hugging Face guide here.

Try out the model on your own sentences

You can use the script below to test the model on any sentence.

Results

Let's look at some of the paraphrased sequences generated by the model for the test data. For each input sequence, the model will generate three (num_return_sequences) paraphrased sequences.

1.

```
Orignal:A recording of folk songs done for the Columbia society in 1942 was largely arranged by Pjetër Dungu.Truth:A recording of folk songs made for
```

2.

```
Original:In mathematical astronomy, his fame is due to the introduction of the astronomical globe, and his early contributions to understanding the mc
```

3.

```
Original:Why are people obsessed with Cara Delevingne?Truth:Why are people so obsessed with Cara Delevingne?Prediction:Why do people fall in love witl
```

4.

```
Original:Earl St Vincent was a British ship that was captured in 1803 and became a French trade man.Truth:Earl St Vincent was a British ship that was
```

5.

```
Original:Worcester is a town and county city of Worcestershire in England.Truth:Worcester is a city and county town of Worcestershire in England.Predi
```

6. Out of domain sentence

```
Original:The goal of any Deep Learning model is to take in an input and generate the correct output.Predictions >>>
The goal of any deep learning model is to take an input and generate the correct output.The goal of a deep learning model is to take an input and gene
```

As can be seen from these examples, our BART model has learned to generate paraphrases quite well!

Discussion

Potential Problems

The generated paraphrases can sometimes have minor issues, some of which are listed below.

1. The generated sequence is almost identical to the original with only minor differences in a word or two.
2. Incorrect or awkward grammar.
3. Might not be as good on out of domain (from training data) inputs.

Encouragingly, these issues seem to be quite rare and can most likely be averted by using better training data (the same problems can sometimes be seen in the training data ground truth as well).

Wrap Up

Sequence-to-Sequence models like BART are another arrow in the quiver of NLP practitioners. They are particularly useful for tasks involving text generation such as paraphrasing, summarization, and abstractive question answering.

Paraphrasing can be used for data augmentation where you can create a larger dataset by paraphrasing the available data.

Thilina Rajapakse

AI researcher, avid reader, fantasy and Sci-Fi geek, and fan of the Oxford comma. www.linkedin.com/in/t-rajapakse/

Follow

By Towards Data Science

Every Thursday, the Variable delivers the very best of Towards Data Science: from hands-on tutorials and cutting-edge research to original features you don't want to miss. Take a look.

Get this newsletter

By signing up, you will create a Medium account if you don't already have one. Review our Privacy Policy for more information about our privacy practices.

Check your inbox
Medium sent you an email at to complete your subscription.

162

162

3

- Artificial Intelligence
- Machine Learning
- Data Science
- NLP

More from Towards Data Science

Follow

Your home for data science. A Medium publication sharing concepts, ideas and codes.

Read more from Towards Data Science

More From Medium

FuncTools: An Underrated Python Package

Emmett Boudreau in Towards Data Science

Killer Data Processing Tricks For Python Programmers

Emmett Boudreau in Towards Data Science

All The Important Features and Changes in Python 3.10

Martin Heinz in Towards Data Science

A Simple Guide to Beautiful Visualizations in Python

Frank Andrade in Towards Data Science

The Ultimate Interview Prep Guide for Data Scientists and Data Analysts

Tessa Xie in Towards Data Science

How to Study for the Google Data Analytics Professional Certificate

Madison Hunter in Towards Data Science

Why You Should Consider Being a Data Engineer Instead of a Data Scientist.

Terence Shin in Towards Data Science

15 More Surprisingly Useful Python Base Modules

Emmett Boudreau in Towards Data Science