Text Summarization with Pretrained Encoders

This folder contains part of the code necessary to reproduce the results on abstractive summarization from the article Text Summarization with Pretrained Encoders by Yang Liu and Mirella Lapata. It can also be used to summarize any document.

The original code can be found on the Yang Liu's github repository.

The model is loaded with the pre-trained weights for the abstractive summarization model trained on the CNN/Daily Mail dataset with an extractive and then abstractive tasks.

Setup

```
git clone https://github.com/huggingface/transformers && cd transformers
pip install .
pip install nltk py-rouge
cd examples/seq2seq/bertabs
```

Reproduce the authors' ROUGE score

To be able to reproduce the authors' results on the CNN/Daily Mail dataset you first need to download both CNN and Daily Mail datasets from Kyunghyun Cho's website (the links next to "Stories") in the same folder. Then uncompress the archives by running:

```
tar -xvf cnn_stories.tgz && tar -xvf dailymail_stories.tgz
```

And move all the stories to the same folder. We will refer as `$DATA_PATH` the path to where you uncompressed both archive. Then run the following in the same folder as `run_summarization.py`:

```
python run_summarization.py \
    --documents_dir $DATA_PATH \
    --summaries_output_dir $SUMMARIES_PATH \ # optional
    --no_cuda false \
    --batch_size 4 \
    --min_length 50 \
    --max_length 200 \
    --beam_size 5 \
    --alpha 0.95 \
    --block_trigram true \
    --compute_rouge true
```

The scripts executes on GPU if one is available and if `no_cuda` is not set to `true`. Inference on multiple GPUs is not supported yet. The ROUGE scores will be displayed in the console at the end of evaluation and written in a rouge_scores.txt file. The script takes 30 hours to compute with a single Tesla V100 GPU and a batch size of 10 (300,000 texts to summarize).

Summarize any text

Put the documents that you would like to summarize in a folder (the path to which is referred to as $DATA_PATH below) and run the following in the same folder as run_summarization.py:

```
python run_summarization.py \
    --documents_dir $DATA_PATH \
    --summaries_output_dir $SUMMARIES_PATH \ # optional
    --no_cuda false \
    --batch_size 4 \
    --min_length 50 \
    --max_length 200 \
    --beam_size 5 \
    --alpha 0.95 \
    --block_trigram true \
```

You may want to play around with min_length, max_length and alpha to suit your use case. If you want to compute ROUGE on another dataset you will need to tweak the stories/summaries import in utils_summarization.py and tell it where to fetch the reference summaries.