

NTU ADL Homework 2, 2023 Fall.

Model

Model & Preprocessing

- mT5 is a basic Encoder-Decoder transformer architecture.
- mT5's text summarization process involves encoding both source and target texts into continuous vectors through an embedding layer, using Byte-Pair Encoding for multilingual and character-level text. It follows a "text-to-text" approach, employing self-attention mechanisms and Transformer encoders to transform the source text into a summary. Self-supervised training minimizes differences between source and target texts, teaching the model to generate concise summaries. Post-training, the model uses a decoder to produce summaries from source text, emphasizing essential information.

Training

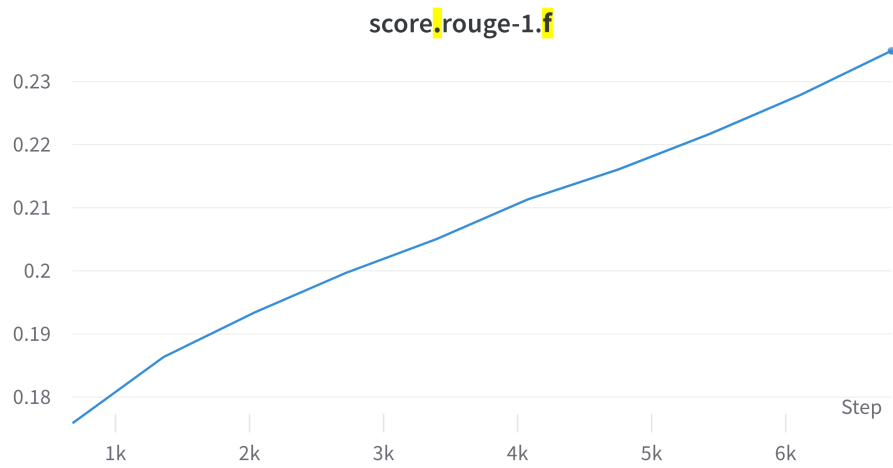
Hyperparameters

- `max_source_length=256`
- `max_target_length=64`
- *After observing the input data and TA's tips, I decide to use 256 as my `max_source_length` and use 64 as my `max_target_length`.*
- `val_max_target_length=32`
- *However, while taking 32 as `val_max_target_length`, the result always include a BOS special token, after changing my `val_max_target_length` to 32 it fixed!*
- *So, I choose 32 as my `val_max_target_length`.*

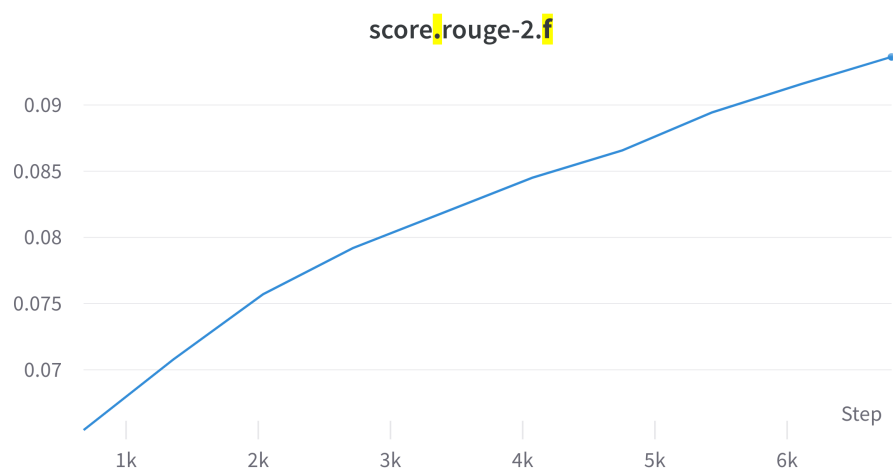
- `source_prefix="summarization: "`
- *This is based on the official repo of mT5, they claim that adding this prefix to the source will improve the performance.*
- `per_device_eval_batch_size=2`
- `per_device_train_batch_size=2`
- *I've tried a lot different size, and find that 2 is the most appropriate number of it.*
- `learning_rate=5e-5,`
- `weight_decay=2e-5,`
- *Not change, same with the original value of the official repo.*
- `epochs=10,`
- `gradient_accumulation_steps=16,`
- *After a lot of experiment, 16 is the base number of all time.*
- `lr_scheduler_type=cosine,`
- `beams=30,`
- `seed=42`

Curves

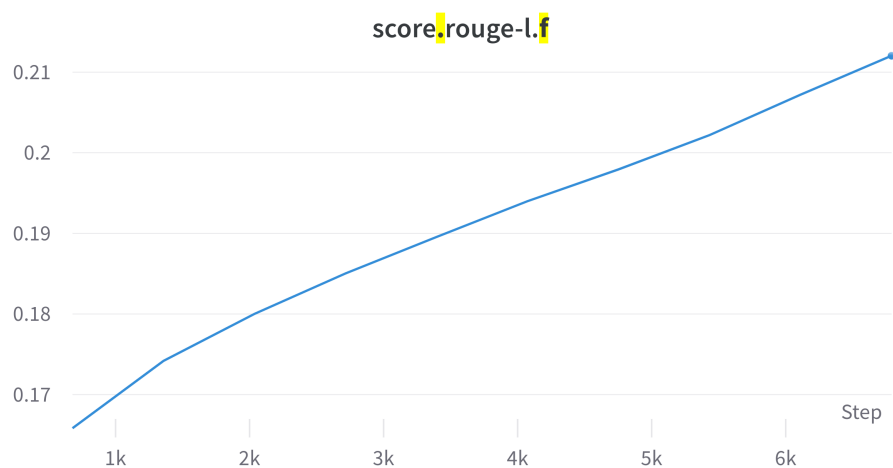
ROUGE1.F



ROUGE2.F



ROUGEL.F



- As you can see, the score is not ending growing up in 10 epochs, so, I should increase the training epochs.

Generation Strategies

Strategies

GREEDY GENERATION:

- Greedy generation is the simplest strategy where the model chooses the word with the highest probability for the next word in a sequence at each step.
- This strategy often leads to coherent but repetitive and predictable text, as it doesn't explore alternative word choices.

BEAM SEARCH:

- Beam Search is a more advanced strategy that aims to improve the diversity and coherence of generated text.
- It maintains a set of multiple candidate sequences (the "beam") and extends them by predicting the next word for each candidate. At each step, it selects the top-k candidates with the highest probabilities.
- This strategy explores multiple possibilities and can lead to more diverse and contextually relevant outputs.

TOP-K SAMPLING:

- Top-k sampling is a probabilistic strategy where the model selects from the top-k most likely words at each generation step, based on their probabilities.
- It adds an element of randomness and allows for a variety of outputs, but still restricts the choices to a predefined number of words. This can help balance diversity and coherence.

TOP-P SAMPLING:

- Top-p sampling, also known as nucleus sampling, is a strategy that selects words from the most

probable words whose cumulative probability is below a certain threshold p .

- This approach adapts to the changing distribution of word probabilities and allows for a dynamic set of candidates, which makes it more adaptable and potentially more diverse than top-k sampling.

TEMPERATURE:

- Temperature is a parameter that can be adjusted to control the randomness of the model's output during generation.
- A higher temperature (e.g., 1.0) makes the model's predictions more random, resulting in more diverse but potentially less coherent text. A lower temperature (e.g., 0.2) makes the predictions more deterministic and focused on the most probable words, leading to more conservative and deterministic text.

Hyperparameters:

- I tried 2 settings of each strategy (unless greedy strategy), the hyperparameters are down below.

SETTING 1:

beams=10, top_k=30, top_p=0.8, temperature=0.6:

```
Greedy: {'rouge-1': {'r': 0.1926826060755947, 'p': 0.26426854813736156, 'f': 0.21383303442621515}, 'rouge-2': {'r': 0.0787865551005832, 'p': 0.08959518395100159, 'f': 0.07575187552278989}, 'rouge-l': {'r': 0.1737868597528498, 'p': 0.23794286298283254, 'f': 0.19255267330460288}}
Beams: {'rouge-1': {'r': 0.21547020975686888, 'p': 0.2683679969716713, 'f': 0.22922276399997707}, 'rouge-2': {'r': 0.08688454415067896, 'p': 0.1090239228383944, 'f': 0.09201975574580165}, 'rouge-l': {'r': 0.19324618958401412, 'p': 0.24070316628276733, 'f': 0.20547214831058617}}
Top k: {'rouge-1': {'r': 0.17138211452680374, 'p': 0.2157957709381367, 'f': 0.18286611076027056}, 'rouge-2': {'r': 0.055220093470994776, 'p': 0.0657185213454485, 'f': 0.05713312943645179}, 'rouge-l': {'r': 0.1532108512948312, 'p': 0.19286556880140685, 'f': 0.16317815565819724}}
Top p: {'rouge-1': {'r': 0.18728794725088824, 'p': 0.2519170635108028, 'f': 0.20588632557671505}, 'rouge-2': {'r': 0.0673454434023263, 'p': 0.08519867135173967, 'f': 0.0720444539227493}, 'rouge-l': {'r': 0.16976924819378156, 'p': 0.2284848955705184, 'f': 0.18653540931766893}}
Temperatures: {'rouge-1': {'r': 0.19071897341694244, 'p': 0.24645907548934, 'f': 0.2122075154948009}, 'rouge-2': {'r': 0.06921650956366027, 'p': 0.09864610596989905, 'f': 0.07589481845963318}, 'rouge-l': {'r': 0.17321839327369015, 'p': 0.2408846437127971, 'f': 0.19283428158826956}}
```

SETTING 2:

beams=20, top_k=15, top_p=0.4, temperature=0.3:

```
Greedy: {'rouge-1': {'r': 0.1926826060755947, 'p': 0.26426854813736156, 'f': 0.21383303442621515}, 'rouge-2': {'r': 0.0787865551005832, 'p': 0.08959518395100159, 'f': 0.07575187552278989}, 'rouge-l': {'r': 0.1737868597528498, 'p': 0.23794286298283254, 'f': 0.19255267330460288}}
Beams: {'rouge-1': {'r': 0.21547020975686888, 'p': 0.2683679969716713, 'f': 0.22922276399997707}, 'rouge-2': {'r': 0.08688454415067896, 'p': 0.1090239228383944, 'f': 0.09201975574580165}, 'rouge-l': {'r': 0.19324618958401412, 'p': 0.24070316628276733, 'f': 0.20547214831058617}}
Top k: {'rouge-1': {'r': 0.17138211452680374, 'p': 0.2157957709381367, 'f': 0.18286611076027056}, 'rouge-2': {'r': 0.055220093470994776, 'p': 0.0657185213454485, 'f': 0.05713312943645179}, 'rouge-l': {'r': 0.1532108512948312, 'p': 0.19286556880140685, 'f': 0.16317815565819724}}
Top p: {'rouge-1': {'r': 0.18728794725088824, 'p': 0.2519170635108028, 'f': 0.20588632557671505}, 'rouge-2': {'r': 0.0673454434023263, 'p': 0.08519867135173967, 'f': 0.0720444539227493}, 'rouge-l': {'r': 0.16976924819378156, 'p': 0.2284848955705184, 'f': 0.18653540931766893}}
Temperatures: {'rouge-1': {'r': 0.19071897341694244, 'p': 0.24645907548934, 'f': 0.2122075154948009}, 'rouge-2': {'r': 0.06921650956366027, 'p': 0.09864610596989905, 'f': 0.07589481845963318}, 'rouge-l': {'r': 0.17321839327369015, 'p': 0.2408846437127971, 'f': 0.19283428158826956}}
```

- I choose beams as my final strategy, according to the result, it's performance is better than others.