Report

Model

In this project, we use the mT5-small (Multilingual T5 with fewer parameters) model to summarize Chinese news. It was created by Google and is part of their T5 family and was designed to handle multiple languages

Architecture

mT5 follows the same architecture as the original T5 model, an encoder-decoder transformer structure, which relies heavily on self-attention mechanisms to process data sequences. It is pre-trained on a large-scale multilingual dataset from the Common Crawl, covering 101 languages. It uses a denoising objective similar to the one used in T5, where the model is trained to predict missing text parts.

- Tokenization:
  mT5 uses the SentencePiece model for tokenization, which treats the text as a sequence of Unicode characters and learns subword units (like byte-pair encoding) for efficient representation. This is particularly important for handling the many languages in its training data, some of which may have very different scripts and morphological structures.
- Span Corruption:
  During pre-training, random text spans are replaced with a unique identifier (a sentinel token), and the model learns to fill in these gaps. This process is a form of self-supervised learning that helps the model understand the context and improve its generative capabilities.

Capabilities

Due to its pre-training, mT5 can perform various text-based tasks without needing task-specific architecture changes. With task-specific prefixes, it can translate, summarize, answer questions, and more across different languages.

How this project works for text summarization

1. **Preprocessing**: Tokenize and encode document into subwords, add "summarize:" prefix, and limit the total number of tokens to a maximum length of 1024 with padding or truncation.

2. **Decoding**: Decoder predicts summary tokens sequentially, using encoded input and prior tokens.
3. **Attention Mechanism**: Self-attention in the encoder and decoder focuses on relevant input parts for summarization.
4. **Generation Strategies**: Inference uses beam search to maintain multiple probable summary sequences for coherence.

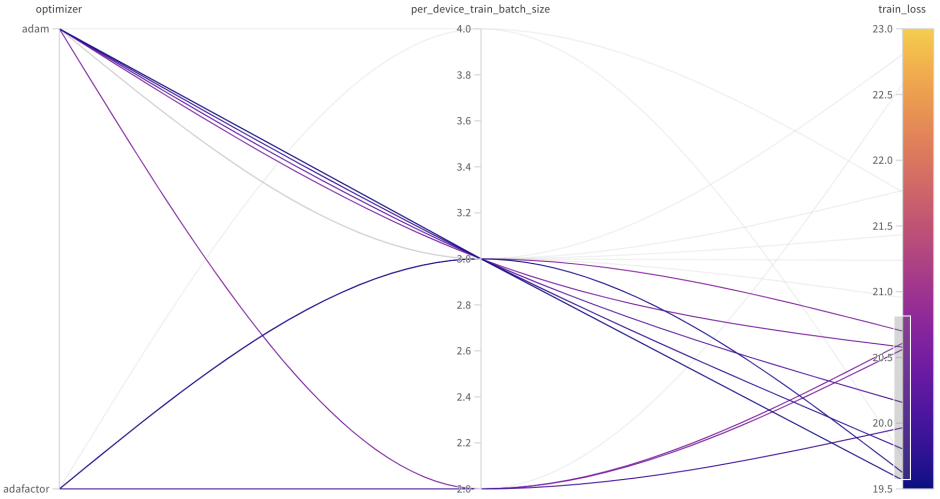
Training

Hyperparameter Sweeping

The goal was to identify the optimal set of hyperparameters that minimize the training loss of the mT5-small model while keeping the gradient accumulation steps and number of training epochs fixed.

- Methodology:
  A random search method was employed for hyperparameter optimization, facilitated by Weights & Biases. With the fixed training epoch (1) and gradient accumulation steps (4), searching for 17 permutations was configured to minimize the training loss.
- Hyperparameters Explored and Results:

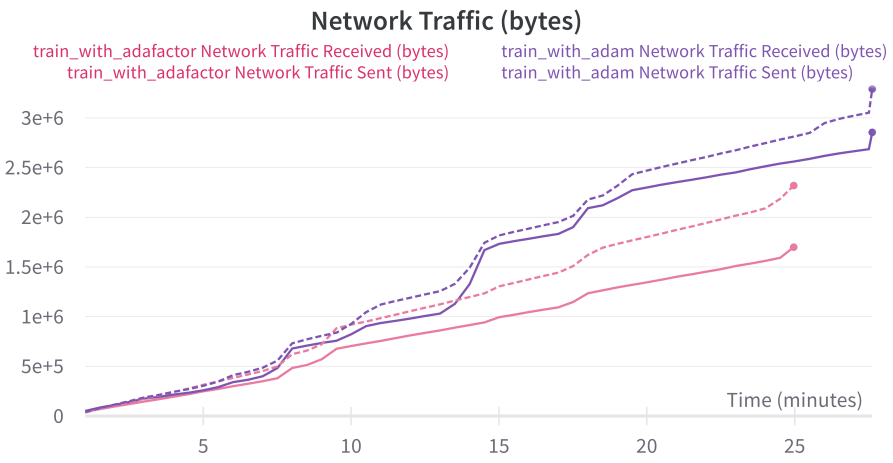
  Two primary hyperparameters were varied during the sweep:



1. **Per Device Train Batch Size**: 2-4 (Quantized uniform distribution)
   - This parameter was found to be a good predictor of training loss.

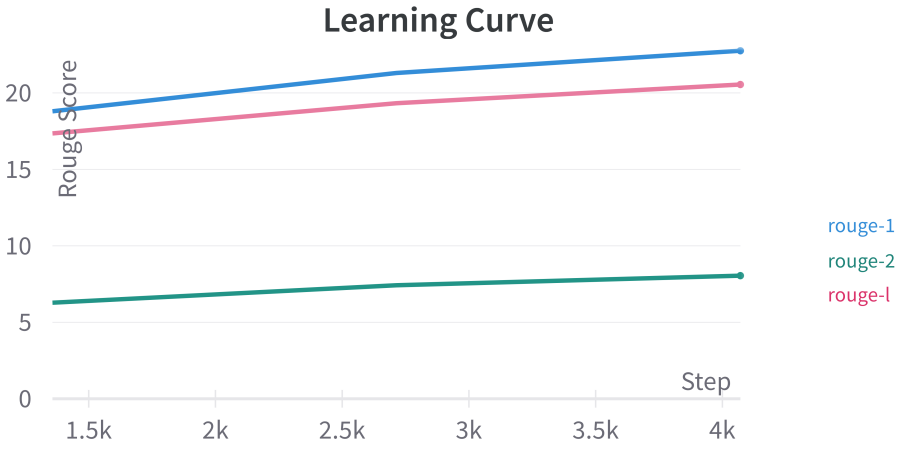
- A batch size of 2 to 3 was identified as optimal, balancing the computational load and model performance.

2. **Optimizer**: Adam, Adafactor
    - No significant difference in training loss was observed between the Adam and Adafactor optimizers.
    - Adafactor was determined to be the more efficient optimizer in computing power, making it the preferred choice for the training of the mT5 model.



Learning Curves

- We later use Adafactor optimizer with batch size 2 to train our model, receiving accuracy: {'rouge-1': 25.82, 'rouge-2': 10.14, 'rouge-l': 23.08}



Generation Strategies

For suitable implementation, we tested out various generation strategies:

1. **Greedy Search:**
    - Greedy search selects the most probable next word at each step in the sequence.
    - This method can lead to repetitive and less diverse text because it never explores second-best options.
2. **Beam Search:**
    - Beam search expands on greedy search by considering a fixed number of the most probable sequences (the "beam size") and explores further steps for each of these sequences.
    - This approach balances between finding the most probable sequence and maintaining multiple candidate sequences, leading to more coherent outputs than greedy search.
3. **Top-k Sampling:**
    - Top-k sampling randomly picks the next word from the top k most likely candidates instead of the most likely next word.
    - By limiting the sample pool to the top k, it introduces randomness but avoids highly improbable words, leading to more diverse and interesting text.
4. **Top-p (Nucleus) Sampling:**
    - Top-p sampling chooses the next word from a subset of the vocabulary where the cumulative probability exceeds a threshold p.
    - It helps to focus on a "nucleus" of plausible options, generating text that is both diverse and high-quality.
5. **Temperature:**
    - Temperature modifies the probability distribution used for generation.
    - Higher temperature increases diversity; lower temperature favors accuracy.

Here’s the results and summarized insights of scoring patterns:

Generation Strategy	ROUGE-1	ROUGE-2	ROUGE-L
Greedy	24.57	9.03	22.05
Beam Search (3)	25.82	10.14	23.08
Beam Search (5)	25.57	10.11	22.94
Top-K Sampling (10)	21.48	6.60	19.19
Top-K Sampling (50)	18.70	6.18	15.98
Top-P Sampling (0.3)	23.66	9.62	21.69
Top-P Sampling (0.7)	22.27	8.21	19.52
Temperature (0.5)	23.78	8.66	21.75
Temperature (2.0)	11.41	1.87	9.62

1. **Beam Search Improves Quality**: The beam search strategy with both beam sizes of 3 and 5 outperforms the other methods in terms of ROUGE scores, suggesting that beam search tends to generate higher quality summaries. This is likely due to its mechanism of keeping multiple hypotheses at each step and choosing the one with the highest overall score.
2. **Sampling Methods Lower Scores**: Top-K and Top-P sampling methods generally result in lower ROUGE scores compared to beam search and greedy methods. This suggests that while sampling methods introduce diversity in the generated text, they may not always align well with the reference summaries. The temperature-controlled sampling, especially with a high temperature of 2.0, significantly decreases performance, indicating that too much randomness can degrade the quality of the generated summaries.

Based on the discovery above, we chose the beam search method with the highest score (beam size = 3) as the generation strategy of the model.