ADL HW2 Report

Student ID: m11203404
Name: 陳旭霖

Q1: Model

- Model

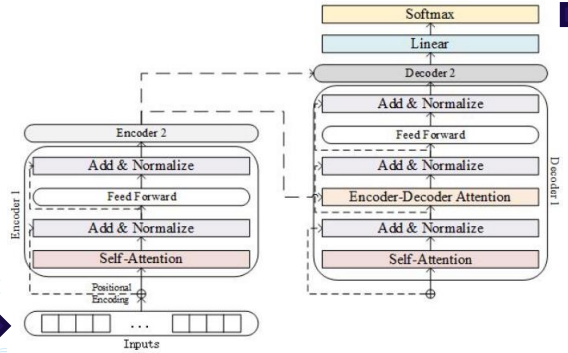
Q: Describe the model architecture and how it works on text summarization.

A: Because I had to handle the task of Chinese summarization, I used the mt5-small model released by Google, which is a multilingual variant of T5. The architecture of mt5 is similar to T5, both utilizing an encoder-decoder framework. However, mt5 replaces the activation function with GeGLU and does not use Dropout during training on unlabeled data.

In downstream tasks like summarization, mt5 uses prefixes during training to enable the model to automatically determine the type of text to generate. Hence, it is necessary to include the prefix 'summarize' during both training and inference stages.

output

門票秒殺！ YOASOBI開演唱會 台粉求票「願改名」



input

Summarize:日本人氣天團「YOASOBI」明年1月21號來台開唱.....回應歌迷期待。

Q1: Model

- Preprocessing

Q: Describe your preprocessing (e.g. tokenization, data cleaning and etc.)

A: When I use the tokenizer, I read the MT5 pre-trained model. Both input and output undergo truncation, with a maximum input length of 256 and a maximum output length of 64. During the tokenization process, the tokenizer automatically converts Chinese words into corresponding IDs based on the dictionary.

```
# preprocess
tokenizer = AutoTokenizer.from_pretrained(model_name)
def preprocess_function(examples):
    inputs = [prefix + doc for doc in examples["text"]]
    model_inputs = tokenizer(inputs, max_length=256, truncation=True)

    labels = tokenizer(examples["summary"], max_length=64, truncation=True)

    model_inputs["labels"] = labels["input_ids"]
    return model_inputs

tokenized_train = train_data.map(preprocess_function, batched=True)
tokenized_valid = valid_data.map(preprocess_function, batched=True)
data_collator = DataCollatorForSeq2Seq(tokenizer=tokenizer, model=model_name)
```

Q2: Training

- Hyperparameter

Q: Describe your hyperparameter you use and how you decide it.

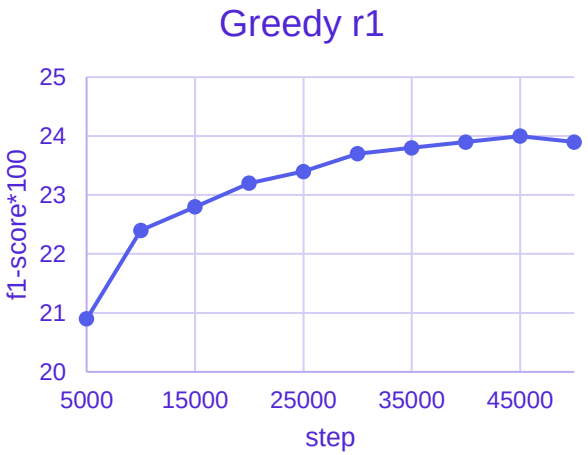
A: To reduce training time, the input maximum length is set to 256, the minimum output length is 64, and we use fp16 for computational precision.

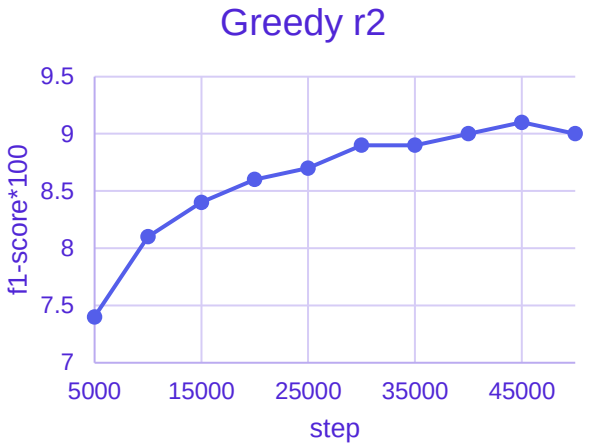
1. Epoch: 40
2. Batch_size: 16
3. Input_max_length: 256
4. output_max_length: 64
5. Learning_rate: 0.00002
6. Weight_decay: 0.01
7. Warmup_steps: 100
8. Optimizer: adafactor
9. fp16

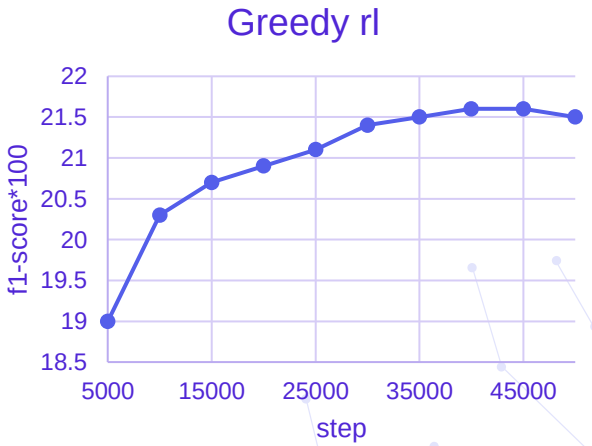
Q2: Training

- Learning Curves

Q: Plot the learning curves (ROUGE versus training steps)







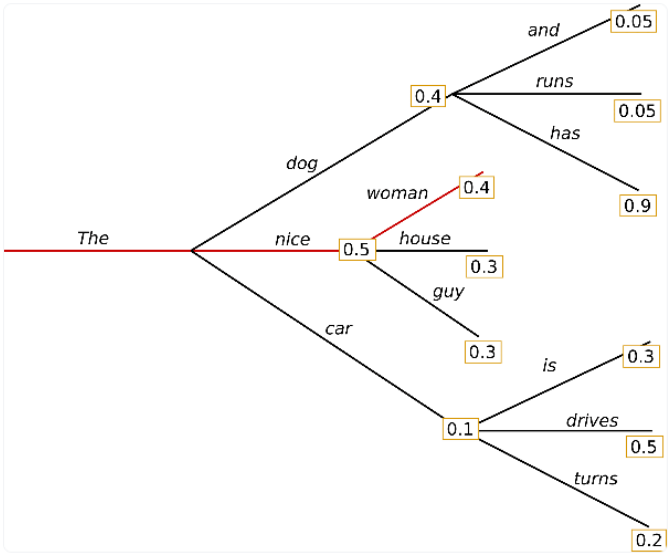
Q3: Generation Strategies

- Stratgies

Q: Describe the detail of the following generation strategies.

A:

1. **Greedy**: It is the most naive decoding technique, as it directly chooses the next word with the highest probability.



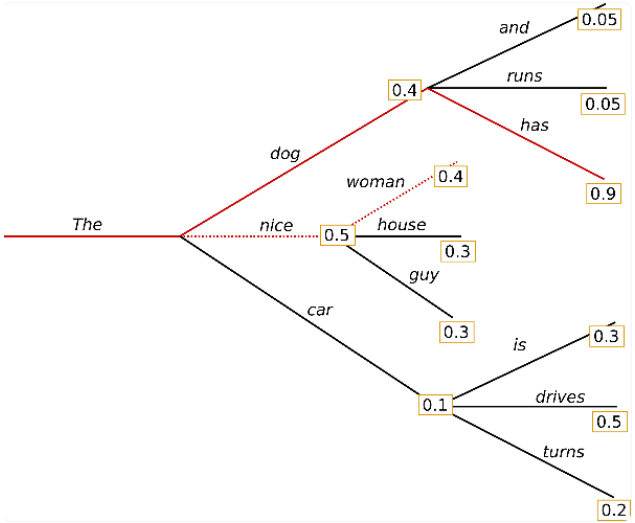
Q3: Generation Strategies

- Stratgies

Q: Describe the detail of the following generation strategies.

A:

2. **Beam search**: To avoid being influenced by the probability of a single word, multiple beams are first sampled, and then the beam with the highest overall probability is selected.



Q3: Generation Strategies

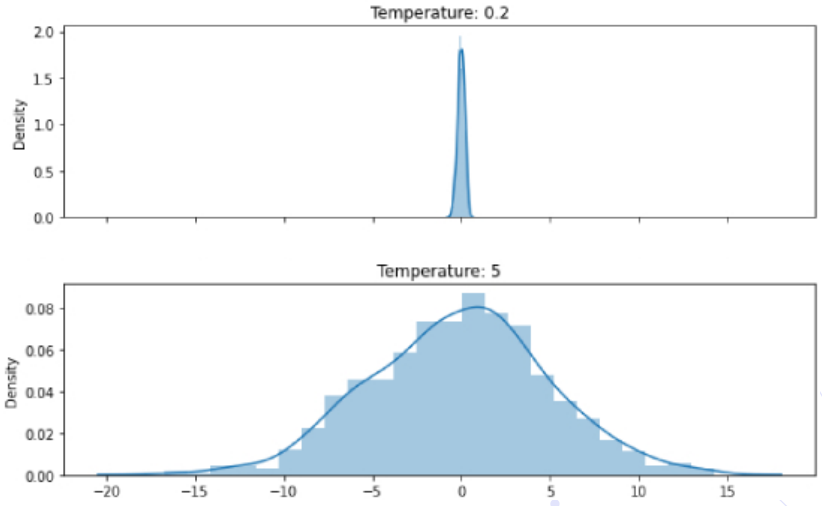
- Stratgies

Q: Describe the detail of the following generation strategies.

A:

3. **Temperature**: It will randomly select the next word based on the probabilities associated with each possible word. The Temperature technique allows the adjustment of the distribution of these probabilities. For example, it can sharpen the distribution, making it easier to select high-probability words when choosing randomly.

$$p(x_i) = \frac{e^{\frac{x_i}{T}}}{\sum_{j=1}^{V} e^{\frac{x_j}{T}}}$$



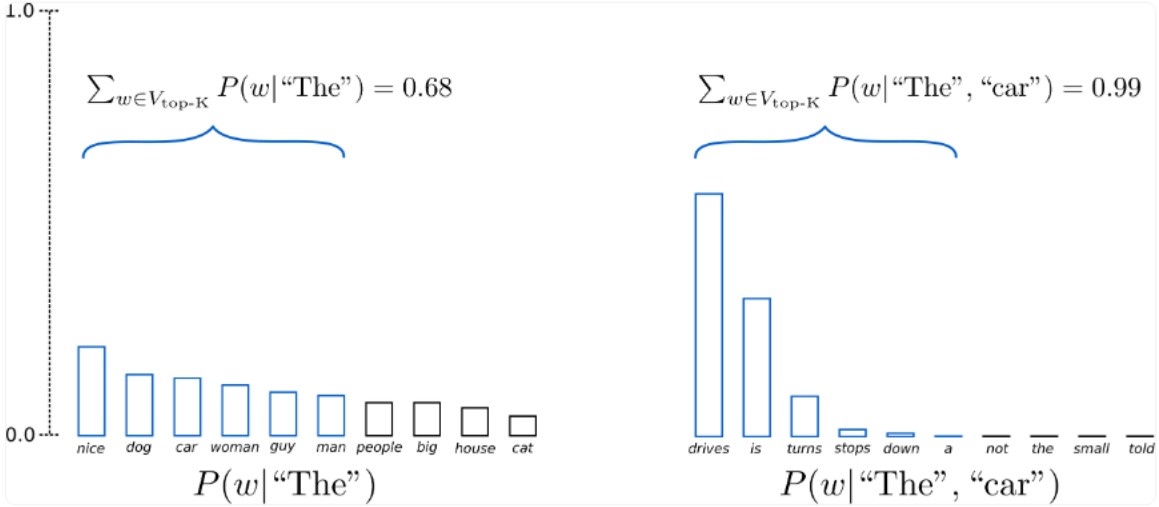
Q3: Generation Strategies

- Stratgies

Q: Describe the detail of the following generation strategies.

A:

4. **Top-k**: During the process of selecting the next word, the top N words with the highest probabilities will be filtered out first. Then, words from this filtered group will be sampled based on their individual probabilities.



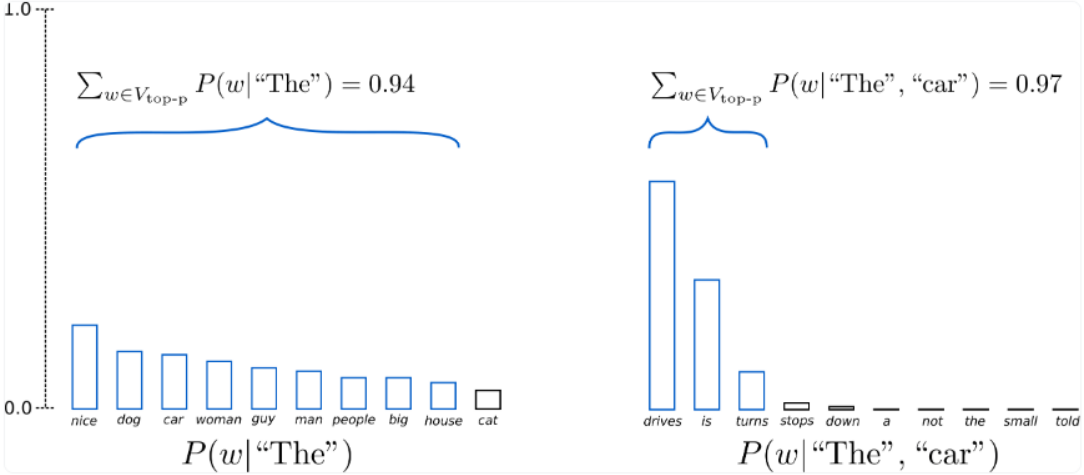
Q3: Generation Strategies

- Stratgies

Q: Describe the detail of the following generation strategies.

A:

5. **Top-p**: Top-p sampling selects words from a reduced set where the cumulative probability surpasses a specified threshold, denoted as p. The probability mass is then reallocated among this chosen set of words. This approach allows for the set's size, or the number of words within it, to dynamically expand or contract based on the probability distribution of the subsequent word.



Q3: Generation Strategies

- Hyperparameters

Q: Try at least 2 settings of each strategies and compare the result.

A:

beam				
n	phase	r1	r2	rl
3	k	25.1	9.9	22.6
5	k	25.1	10.1	22.6
8	k	25.1	10.2	22.6

topk				
k	phase	r1	r2	rl
8	k	21.7	7.4	19.2
16	k	20.7	6.9	18.2
32	k	19.8	6.4	17.5

temperature				
t	phase	r1	r2	rl
0.5	K	22.7	8.1	20.3
0.7	K	20.7	7.1	18.5
1.0	K	14.1	4.2	12.6

topp				
p	phase	r1	r2	rl
0.98	K	14.6	4.4	13.1
0.94	K	15.5	4.8	13.8
0.86	k	16.7	5.4	14.9

Q3: Generation Strategies

- Hyperparameters

Q: What is your final generation strategy? (you can combine any of them)

A: I chose beam search as the generation strategy because its performance is the best among all generation strategies.

beam				
n	phase	r1	r2	rl
3	k	25.1	9.9	22.6
5	k	25.1	10.1	22.6
8	k	25.1	10.2	22.6