# [ADL 2023 Fall]
# HW2 Report

Student ID: R12945039　Name: 楊瀚博

## Q1: Model

### ■ Model

MT5-small (Multilingual Text-to-Text Transfer Transformer, small version) is a multilingual model based on the Transformer architecture.

The model takes input texts, which can be in various languages, and converts it into a **numerical representation** (tokens to embeddings). Afterwards, the input embeddings are then passed through a stack of **encoder layers**, which consists of multiple **self-attention** mechanisms and **feed-forward neural networks**. These layers allow the model to learn contextual information from the input texts, capturing relationships between words. For text summarization, the model uses a **decoder**, which consists of **self-attention** and **cross-attention** mechanisms, to generate summaries of the input texts.

During training, the model is fine-tuned based on **cross-entropy** loss function. It is provided with training data consisting of **pairs of source texts** and corresponding **reference summaries**. The model learns to generate summaries that are close to the reference summaries by minimizing the cross-entropy loss between the generated output and the target output.

At each step of the inference process, the model uses the **outputs** from the **encoder** and the inputs from the decoder as **cross-attention** inputs, generating **one token at a time**. It predicts the probability distribution (**condition** on the **previous generated tokens**) over the vocabulary tokens and selects the best next one using various decoding strategies. This process (feeding the model with the **generated token sequence**) is **repeated** until the model generates the **end of sentence (EOS)** token. Finally, we decode the output tokens into words to get the predicted summaries.

## Preprocessing

### *Data Cleaning*

Before sending training data pairs into the tokenizer, we first remove pairs where at least one record (main-text or title) is absent.

```
def preprocess_function(examples):
    # remove pairs where at least one record is None
    inputs, targets = [], []
    for i in range(len(examples[text_column])):
        if examples[text_column][i] and examples[summary_column][i]:
            inputs.append(examples[text_column][i])
            targets.append(examples[summary_column][i])
```

### *Tokenization*

Subsequently, we preprocess the data pairs on the basis of the following steps:

1. Prefix the input with a prompt ***"summarize: "*** so that MT5 model knows this is a summarization task. (Some models capable of multiple NLP tasks require prompting for specific tasks.)
2. Use the keyword ***"text_target"*** argument when tokenizing **labels**.
3. Truncate sequences to be no longer than the maximum length set by the ***"max_length"*** parameter.

```
inputs = [prefix + input for input in inputs]
model_inputs = tokenizer(
    inputs,
    max_length=data_args.max_source_length,
    padding=padding,
    truncation=True,
)

# Tokenize targets with the `text_target` keyword argument
labels = tokenizer(
    text_target=targets,
    max_length=max_target_length,
    padding=padding,
    truncation=True,
)
```

# Q2: Training

## ■ Hyperparameter

| | |
|---|---|
| **Loss Function** | *Cross-Entropy* |
| **Optimization Algorithm** | *AdaFactor* |
| **Learning Rate** | *3e-4* |
| **Warmup Ratio** | *0.05* |
| **Batch Size per Device** | *4* |
| **Gradient Accumulation Steps** | *16* |
| **Effective Batch Size** | *64* |
| **Epochs** | *25* |
| **Max Source Length** | *256* |
| **Max Target Length** | *64* |
| **Padding to Max Length** | *True* |

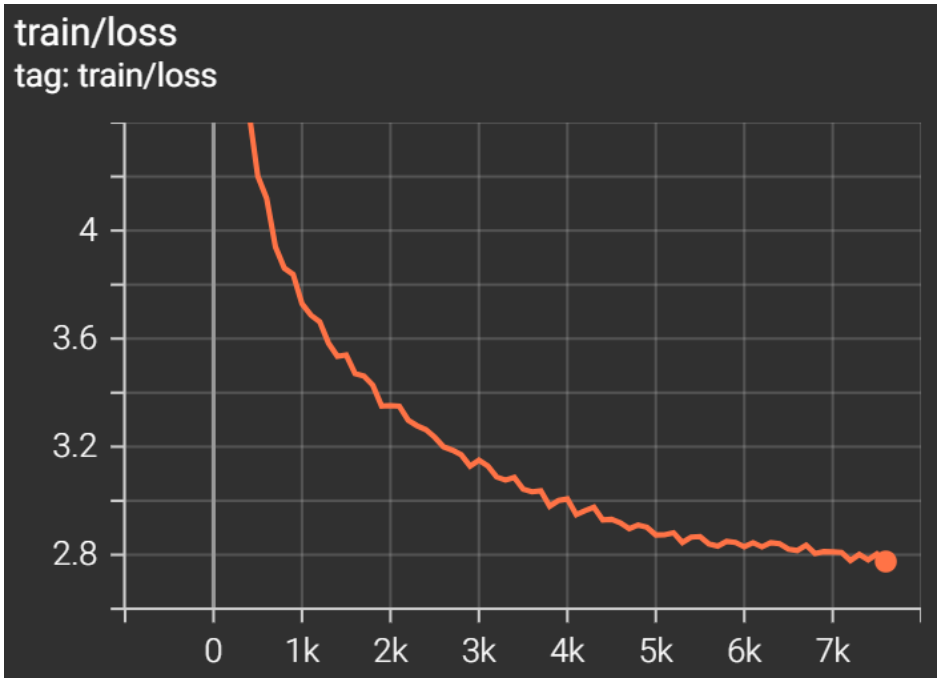
We set the **learning rate** to **3e-4** based on the **trial & error** process. When the learning rate is set to **5e-5**, which is the default value of MT5-small model, both the training and the validation loss **stop dropping** after 8$^{th}$ epoch, indicating that the model converges too slowly or gets stuck in a local minimum. On the other hand, when the learning rate is set to **8e-4**, both the training and validation loss start to **oscillate**, indicating a too high learning rate. The final learning rate, 3e-4, makes the model learn efficiently.

Moreover, we increase the batch size to 64 so that the learning process could be **smoother** and **more stable**. Accordingly, we also **increase** the number of **epochs** for training, hence the model could **converge** eventually.

## ■ Learning Curves

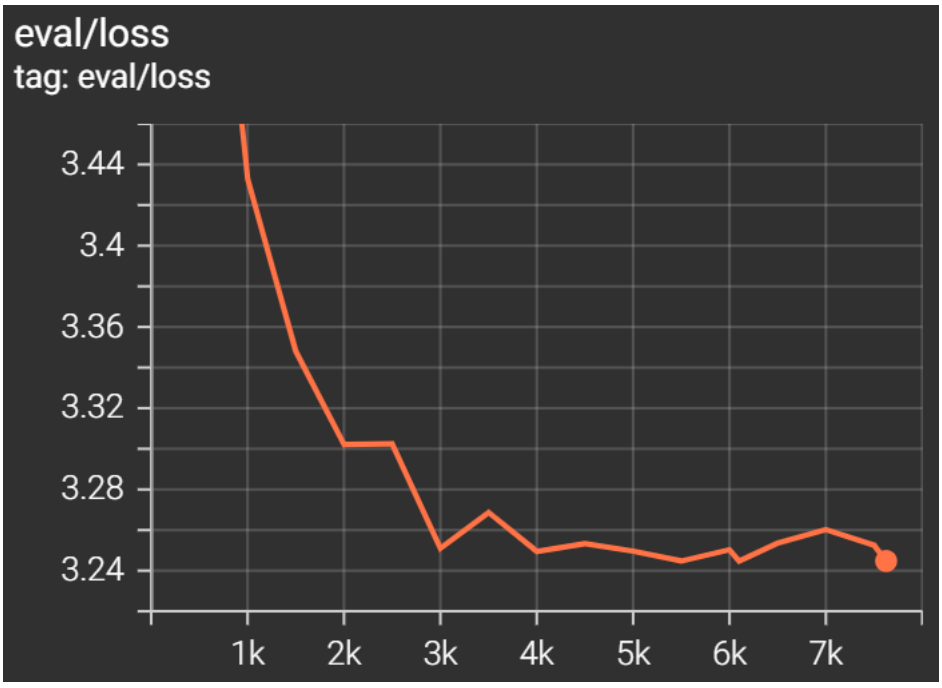
We randomly split the ***train.jsonl*** file into "train" and "validation" datasets (9:1) for training and evaluation. These curves are plotted by the TensorBoard packages using the logs files.

***(Note**: The rouge score in the curves showed below is calculated based on the **"evaluate"** and **"rouge-score"** package, **not** the **"tw_rouge"** package.)*



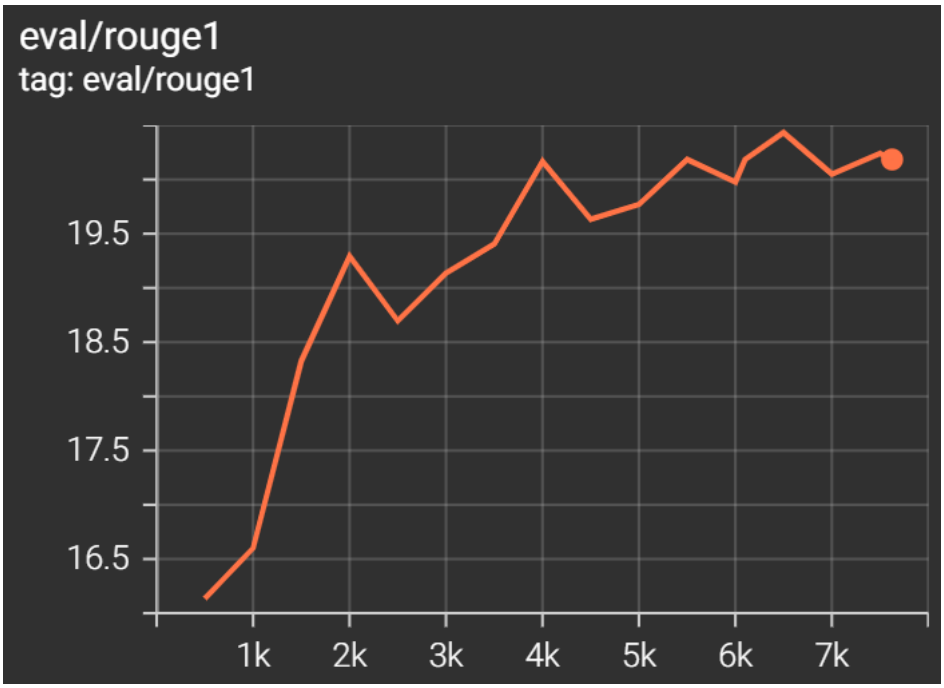

***Loss Curve on Train Dataset***

*(Vertical Axis: **Loss Value**, Horizontal Axis: **Training Steps**)*



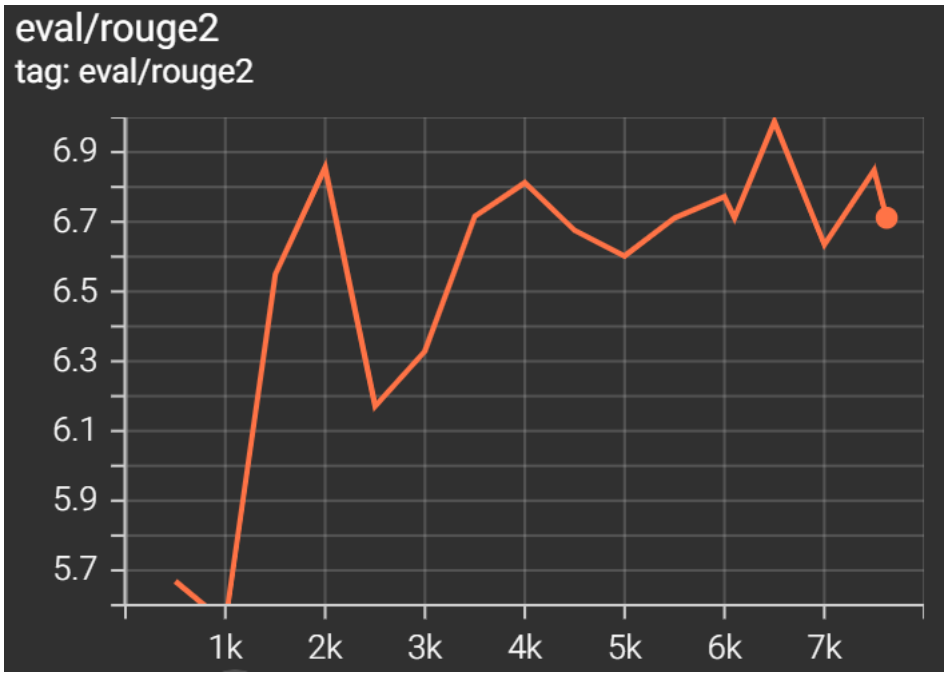

***Loss Curve on Validation Dataset***

*(Vertical Axis: **Loss Value**, Horizontal Axis: **Training Steps**)*



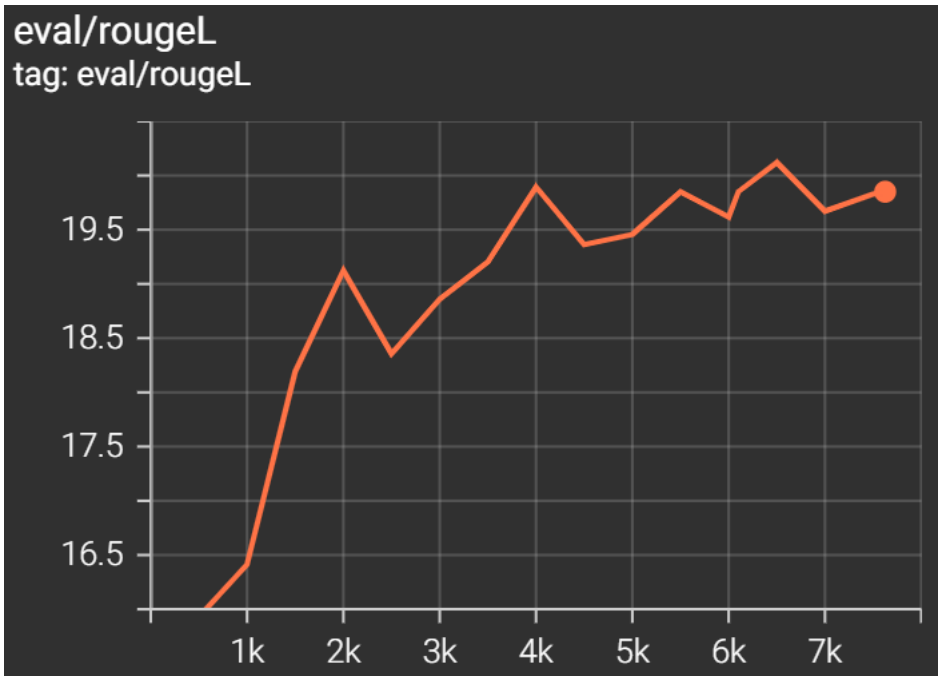

***Rouge-1 Score on Validation Dataset***

*(Vertical Axis: **Rouge-1 Score**, Horizontal Axis: **Training Steps**)*




***Rouge-2 Score on Validation Dataset***

*(Vertical Axis: **Rouge-2 Score**, Horizontal Axis: **Training Steps**)*




***Rouge-L Score on Validation Dataset***

*(Vertical Axis: **Rouge-L Score**, Horizontal Axis: **Training Steps**)*

# Q3: Generation Strategies

## ■ Strategies

### *Greedy*

Greedy decoding is a straightforward strategy where the model selects the token with the **highest probability** at **each step** of the decoding process. Simplicity and speed make greedy an efficient choice for generating text, but it may lead to **suboptimal results** as it can prematurely commit to a word, ignoring the possibility of better choices later in the sequence. This can result in less diverse and less fluent text generation.

### *Beam Search*

Beam search explores multiple token choices at each step and **maintains** a beam of the top ***num_beams*** **candidates**. These candidates are chosen based on their combined probabilities. Beam search keeps the most likely ***num_beams*** of hypotheses at each time step and **eventually choosing** the hypothesis that has the **overall highest probability**.

### *Top-k Sampling*

Top-k sampling is a **probabilistic** decoding strategy that **randomly samples** from the **top-k most likely tokens** at each decoding step. The value of k is a hyperparameter that controls the diversity of the output. Top-k sampling introduces **randomness** and **diversity** into the generated text, making it less deterministic and more creative. Nevertheless, it might sometimes generate **less contextually relevant** text.

### *Top-p Sampling*

Top-p sampling is another **probabilistic** strategy that samples tokens from the set of tokens with **cumulative probabilities** that exceed a certain **threshold**, **p**. It ensures that the chosen tokens are among the most probable, but the **number** of **tokens** can **vary** depending on their probabilities. Top-p sampling allows for more dynamic control over the number of tokens considered, making it adaptable to different contexts. It encourages **diversity** while maintaining **quality**.

### *Temperature*

Temperature is a **hyperparameter** used with **sampling strategies**, including top-k and top-p sampling. It controls the level of randomness in the sampling process. Mathematically, the temperature parameter **adjusts** the **SoftMax** function used in **sampling**.

When temperature is set to a **low value** (e.g., close to 0), the SoftMax function sharpens the probability distribution, **emphasizing** the **most probable tokens**. This makes the text generation more deterministic.

Conversely, when temperature is set to a **high value** (e.g., > 1), the SoftMax function softens the probability distribution, giving **more tokens** a **chance** to be **selected**, even those with lower probabilities. This results in more randomness and diversity in the generated text.

## ■ Hyperparameters

### *Greedy (default)*

- Performance on ***public.jsonl***

| *Rouge-1* | *Rouge-2* | *Rouge-L* |
|---|---|---|
| *25.1853* | *9.3628* | *22.5589* |

- Samples of Generated Title

```
168  {"title": "孤味是女性的孤味 女性的孤味", "id": "21877"}
169  {"title": "天竺鼠車車爆紅!港貨發現活生生生生生生生 網怒轟買家「無人性」", "id": "21878"}
170  {"title": "MLB/洋基獲薪資仲裁 布勒、巴恩斯仍未共識", "id": "21879"}
171  {"title": "雪白迷人!2021中南部賞梅秘境 全台最大賞梅秘境", "id": "21880"}
172  {"title": "影/選對瀏覽器、電視看 Netflix 體驗高畫質的美好", "id": "21881"}
```

### *Greedy (no_repeat_ngram_size = 3)*

- Performance on ***public.jsonl***

| *Rouge-1* | *Rouge-2* | *Rouge-L* |
|---|---|---|
| *25.0778* | *9.1741* | *22.2414* |

- Samples of Generated Title

```
168  {"title": "孤味是女性的孤味 女性的「孤味」", "id": "21877"}
169  {"title": "天竺鼠車車爆紅!港貨發現活生生生天竺豹扣起代為飼養", "id": "21878"}
170  {"title": "MLB/洋基獲薪資仲裁 布勒、巴恩斯仍未共識", "id": "21879"}
171  {"title": "雪白迷人!2021中南部賞梅秘境 全台最大賞梅景點一次看", "id": "21880"}
172  {"title": "影/選對瀏覽器、電視看 Netflix 體驗高畫質的美好", "id": "21881"}
```

### *Beam Search (num_beams = 3)*

- Performance on ***public.jsonl***

| *Rouge-1* | *Rouge-2* | *Rouge-L* |
|---|---|---|
| *26.2021* | *10.4652* | *23.3775* |

- Samples of Generated Title

```
168  {"title": "「孤味」在台語中演繹女性的獨特滋味", "id": "21877"}
169  {"title": "天竺鼠車車爆紅 網怒轟買家「無人性」", "id": "21878"}
170  {"title": "MLB/洋基獲薪資仲裁 布勒、巴恩斯仍未共識", "id": "21879"}
171  {"title": "2021中南部賞梅秘境! 台中「梅花森林、梅花隧道」免費美拍", "id": "21880"}
172  {"title": "用電腦觀看1080P影片 讓你看看畫質上的差別", "id": "21881"}
```

## *Beam Search (num_beams = 5)*

- Performance on ***public.jsonl***

| *Rouge-1* | *Rouge-2* | *Rouge-L* |
|---|---|---|
| *26.3910* | *10.6199* | *23.6194* |

- Samples of Generated Title


```
{"title": "【一劍浣春秋】孤味是什麼?", "id": "21877"}
{"title": "天竺鼠車車爆紅 網怒轟買家「無人性」", "id": "21878"}
{"title": "MLB/洋基獲薪資仲裁 布勒、巴恩斯仍未共識", "id": "21879"}
{"title": "2021中南部賞梅秘境!台中「梅花森林、梅花隧道」免費美拍", "id": "21880"}
{"title": "影/選對瀏覽器 體驗高畫質的美好", "id": "21881"}
```


## *Top-k Sampling (k = 10)*

- Performance on ***public.jsonl***

| *Rouge-1* | *Rouge-2* | *Rouge-L* |
|---|---|---|
| *22.2741* | *7.5292* | *19.7261* |

- Samples of Generated Title


```
{"title": "《孤味》讓男性更具體的「孤味」", "id": "21877"}
{"title": "天竺鼠車車爆紅!港貨「活生生的天」遭控 網怒轟買家「無人性」", "id": "21878"}
{"title": "MLB/MLB/洋基被裁為高教 前球星依然有共識", "id": "21879"}
{"title": "2021中南部賞梅秘境!2021中南部最佳秘境、特色美景全攻略", "id": "21880"}
{"title": "你會用電腦觀看1080P的影片! 教你如何讓你更了解這些小撇步!", "id": "21881"}
```


## *Top-k Sampling (k = 50)*

- Performance on ***public.jsonl***

| *Rouge-1* | *Rouge-2* | *Rouge-L* |
|---|---|---|
| *20.4494* | *6.6737* | *18.0812* |

- Samples of Generated Title


```
{"title": "從孤味看見她 陳伯昌看見老公的孤味", "id": "21877"}
{"title": "天命神遊港貨貨 網怒轟:無人性", "id": "21878"}
{"title": "MLB/法官賈吉獲10萬美元 雙手也無夢被封", "id": "21879"}
{"title": "不只是雪白片林! 2020中南部雪景美拍全攻略", "id": "21880"}
{"title": "免開啟就能看高清影片!測出高畫質的美好", "id": "21881"}
```

## *Top-p Sampling (p = 0.7)*

- Performance on ***public.jsonl***

| *Rouge-1* | *Rouge-2* | *Rouge-L* |
|---|---|---|
| *23.2630* | *8.2273* | *20.6688* |

- Samples of Generated Title


```
{"title": "孤味與女性相愛:女人的孤味 還是孤味?", "id": "21877"}
{"title": "天竺鼠被放入禁運黑名單 網怒轟「無人性」", "id": "21878"}
{"title": "MLB/洋基雙手獲薪資仲裁 布勒也未共識", "id": "21879"}
{"title": "2021中南部賞梅秘境!台中「梅花森林、雪白隧道」免飛台中,還有「鮭魚葡萄」仙境", "id": "21880"}
{"title": "影/在家也能看 Netflix 影片", "id": "21881"}
```


## *Top-p Sampling (p = 0.9)*

- Performance on ***public.jsonl***

| *Rouge-1* | *Rouge-2* | *Rouge-L* |
|---|---|---|
| *21.3869* | *7.3673* | *19.0536* |

- Samples of Generated Title


```
{"title": "孤味是男性的滋味", "id": "21877"}
{"title": "日動畫天竺鼠車車爆紅 網怒轟買家「無人性」", "id": "21878"}
{"title": "MLB/洋基奪新人王 法官賈吉獲10萬美元", "id": "21879"}
{"title": "美拍全台最美梅景!2021中南部賞梅秘境「山城雪景」賞全攻略", "id": "21880"}
{"title": "免開啟就能看高清影片!測出高畫質的美好", "id": "21881"}
```


## *Temperature (T = 0.5)*

- Performance on ***public.jsonl***

| *Rouge-1* | *Rouge-2* | *Rouge-L* |
|---|---|---|
| *24.0860* | *8.7386* | *21.5045* |

- Samples of Generated Title


```
{"title": "孤味的男味 女主角的孤味", "id": "21877"}
{"title": "天竺鼠車車爆紅 網怒轟買家無人性", "id": "21878"}
{"title": "MLB/洋基「法官」獲101萬美元 巴恩斯仍有共識", "id": "21879"}
{"title": "賞梅秘境!2021中南部賞梅秘境「雪白雪景」 台中新社「梅花隧道」全攻略", "id": "21880"}
{"title": "影/想用電腦看電影也能看 1080P影片!測你會怎麼看?", "id": "21881"}
```

### *Temperature (T = 1.5)*

- Performance on ***public.jsonl***

| *Rouge-1* | *Rouge-2* | *Rouge-L* |
|---|---|---|
| *14.8802* | *3.8200* | *13.0641* |

- Samples of Generated Title


```
{"title": "孤味從演繹上女性精神靈魂化身新風潮", "id": "21877"}
{"title": "海軍偷買到香港快來港天免費貓,她衝退錢搶到開貨?", "id": "21878"}
{"title": "MLB/敲名人堂「法官」奇 不僅前一豪都沒有共識", "id": "21879"}
{"title": "15間「中南部賞梅」秘境懶人包!跟著「全美雪必看」!", "id": "21880"}
{"title": "想想進去Apple、Google、Apple等開網 看電影可能是電影外觀差在哪?", "id": "21881"}
```


### *Result Comparison*

According to the rouge score performance and the title generation results showed above, we could compare these text decoding strategies and hyperparameters:

- Greedy Decoding:

  By default, Greedy Decoding may lack diversity and cause repetition issue, the use of ***"no_repeat_ngram_size" = 3*** helps to mitigate this problem.

- Beam Search:

  Increasing ***"num_beams"*** from ***3*** to ***5*** will explore more alternative possibilities during decoding, leading to a better output.

- Top-k Sampling:

  Setting a higher ***"top_k"*** value, such as ***50***, will allow a larger pool of tokens to be considered at each step, **increasing diversity** and **creativity**. A smaller ***"top_k"***, like ***10***, limits the choice of tokens, making the output more focused but potentially ***repetitive***.

- Top-p Sampling:

  A ***"top_p"*** value of ***0.9*** allows more tokens with higher probabilities to be considered, resulting in **more diversity** and **creativity** in the output. Lowering ***"top_p"*** to ***0.7*** restricts the pool of tokens to those with very high probabilities, potentially making the output **more deterministic**.

- Temperature:

  A lower temperature value, such as ***0.5***, makes the output more **deterministic**, favoring the most probable tokens. Increasing the temperature to **1.5** introduces more randomness, potentially leading to more diverse and creative output, while, in this case, the generated outputs become **too creative** and **lack of fluent texts**.

We could end up with a brief summary:

- Greedy decoding is suitable for tasks where deterministic, accurate, and focused output is required.
- **Beam search** is a good choice for generating **fluent**, **coherent text**, such as in translation or **summarization**.
- Top-k and Top-p Sampling are ideal for tasks where diversity and creativity are desired, such as creative writing.
- Temperature provides fine-tuned control, making it adaptable to a wide range of use cases.

### *Final Generation Strategy*

Based on the experiments results, we choose **beam search *(num_beams = 5)*** as the **final strategy** to generate titles. The performance is showed in the following table.

| *Rouge-1* | *Rouge-2* | *Rouge-L* |
|---|---|---|
| *26.3910* | *10.6199* | *23.6194* |

(We also tried the combination of beam search and sampling strategies with different temperature settings, but the model did not perform better on the public dataset.)