ADL2022 HW3 Report

資工四 b08902149 徐晨祐

Q1: Model

1. Model

- **Describe the model architecture and how it works on text summarization.**
  I used **T5** as the pretrained model. T5 is an **encoder-decoder Transformer model** pretrained on a multi-task mixture of supervised tasks and unsupervised tasks (All the tasks are converted into text-to-text format).
  T5 uses an abstractive summarization algorithm to generate new sentences from given text. The algorithm first build an internal semantic representation and use natural language generation technique to create a summary (not just picking up sentences directly from the original text)

2. Preprocessing

- **Describe your preprocessing (e.g. tokenization, data cleaning and etc.)**
  I tokenized the `maintext` field and the `title` field. The maximum length of each `maintext` is set to 256 and that of each `title` is set to 64. I used -100 as the padding token id when padding each `title`to its maximum length.Describe your hyperparameter you use and how you decide it.

Q2: Training

1. Hyperparameter

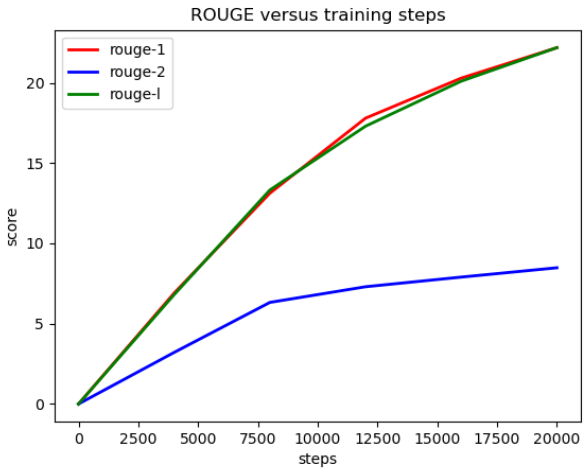
- **Describe your hyperparameter you use and how you decide it.**
  - **Maximum length of tokenized data**
    The maximum length of each `maintext` is set to 256 and that of each `title` is set to 64. The configuration came from TAs' advice so I chose it.
  - **For Optimizer:**

    ```
    weight_decay: 0.0
    lr: None
    ```

    I used `transformers.Adafactor` as my optimizer. The hyperparameters above are the same as the default configuration, I find it effective for training so I used it.
  - **Beam Search:**
    The beam size is set to 7 because it maximize the ROUGE score of my model.

2. Learning Curves

- **Plot the learning curves (ROUGE versus training steps)**



Q3: Generation Strategies

1. Stratgies

- **Describe the detail of the following generation strategies:**
    - **Greedy:** Chooses the most probable word as our result when predicting each word.
    - **Beam Search:** Keeps track of the k-most propable sequences and choose a better one. (k is a hyperparamter)
    - **Top-k Sampling:** Samples the word as our result via distribution but restricted to the top-k propable words when predicting each word.
    - **Top-p Sampling:** Chooses a word from the smallest possible set of words whose cumulative probability exceeds the probability p as our result when predicting each word.
    - **Temperature:** Modify the probability of each word by the following function.

    $$P(x_i|x_{:i-1}) = \frac{exp(u_i/t)}{\Sigma_j\ exp(u_j/t)}$$

    When the range of t (temperature) is $0 < t \leq 1$, temperature will be used to increase the probability of probable tokens while reducing the one that is not. After modifying the probabilities, we sample a word via the new distribution.

2. Hyperparameters

- **Try at least 2 settings of each strategies and compare the result.**
    - **greedy:**

```
{
  "rouge-1": {
    "r": 0.20551761545352792,
    "p": 0.29542129183777777,
    "f": 0.2342291782970164
  },
  "rouge-2": {
    "r": 0.07559887537909897,
    "p": 0.10474380095340757,
    "f": 0.0848342572574422
  },
  "rouge-l": {
    "r": 0.18527374001450989,
    "p": 0.2659040743375466,
    "f": 0.21092596299441815
  }
}
```

- **beam_search; num_beams = 5**

```
  {
    "rouge-1": {
      "r": 0.21706447744469354,
      "p": 0.30691504757772015,
      "f": 0.24620997672681175
    },
    "rouge-2": {
      "r": 0.08465548917593531,
      "p": 0.11831934611959294,
      "f": 0.09531206047714297
    },
    "rouge-l": {
      "r": 0.1958559012063075,
      "p": 0.27655740203024115,
      "f": 0.2218904010879465
    }
  }
```

- **beam_search; num_beams = 7**

```
  {
    "rouge-1": {
      "r": 0.2178477452949044,
      "p": 0.3026019083951786,
      "f": 0.24544516783504253
    },
    "rouge-2": {
      "r": 0.0862245486364595,
      "p": 0.11856444946335135,
      "f": 0.0965703573793699
    },
    "rouge-l": {
      "r": 0.1973533870427799,
      "p": 0.2738950094133531,
      "f": 0.22215173445132555
    }
  }
```

- **top_k; k = 3**

  ```
  {
    "rouge-1": {
      "r": 0.20551761545352792,
      "p": 0.29542129183777777,
      "f": 0.2342291782970164
    },
    "rouge-2": {
      "r": 0.07559887537909897,
      "p": 0.10474380095340757,
      "f": 0.0848342572574422
    },
    "rouge-l": {
      "r": 0.18527374001450989,
      "p": 0.2659040743375466,
      "f": 0.21092596299441815
    }
  }
  ```

- **top_k; k = 10**

  ```
  {
    "rouge-1": {
      "r": 0.20551761545352792,
      "p": 0.29542129183777777,
      "f": 0.2342291782970164
    },
    "rouge-2": {
      "r": 0.07559887537909897,
      "p": 0.10474380095340757,
      "f": 0.0848342572574422
    },
    "rouge-l": {
      "r": 0.18527374001450989,
      "p": 0.2659040743375466,
      "f": 0.21092596299441815
    }
  }
  ```

- **top_p; p = 0.4**

  ```
  {
    "rouge-1": {
      "r": 0.20551761545352792,
      "p": 0.29542129183777777,
      "f": 0.2342291782970164
    },
  ```

```
  "rouge-2": {
    "r": 0.07559887537909897,
    "p": 0.10474380095340757,
    "f": 0.0848342572574422
  },
  "rouge-l": {
    "r": 0.18527374001450989,
    "p": 0.2659040743375466,
    "f": 0.21092596299441815
  }
}
```

- **top_p; p = 0.8**

```
{
  "rouge-1": {
    "r": 0.20551761545352792,
    "p": 0.29542129183777777,
    "f": 0.2342291782970164
  },
  "rouge-2": {
    "r": 0.07559887537909897,
    "p": 0.10474380095340757,
    "f": 0.0848342572574422
  },
  "rouge-l": {
    "r": 0.18527374001450989,
    "p": 0.2659040743375466,
    "f": 0.21092596299441815
  }
}
```

- **temperature = 0.8**

```
{
  "rouge-1": {
    "r": 0.20551761545352792,
    "p": 0.29542129183777777,
    "f": 0.2342291782970164
  },
  "rouge-2": {
    "r": 0.07559887537909897,
    "p": 0.10474380095340757,
    "f": 0.0848342572574422
  },
  "rouge-l": {
    "r": 0.18527374001450989,
    "p": 0.2659040743375466,
      "f": 0.21092596299441815
    }
  }
```

- **temperature = 1.2**

  ```
  {
    "rouge-1": {
      "r": 0.20551761545352792,
      "p": 0.29542129183777777,
      "f": 0.2342291782970164
    },
    "rouge-2": {
      "r": 0.07559887537909897,
      "p": 0.10474380095340757,
      "f": 0.0848342572574422
    },
    "rouge-l": {
      "r": 0.18527374001450989,
      "p": 0.2659040743375466,
      "f": 0.21092596299441815
    }
  }
  ```

  I don't really know why the result when using `top_k`, `top_p` and `temperature` are all the same. (I have made sure my code is correct)

- **What is your final generation strategy? (you can combine any of them)**
  I chose beam search with `num_beam = 7` because it reached a better result.