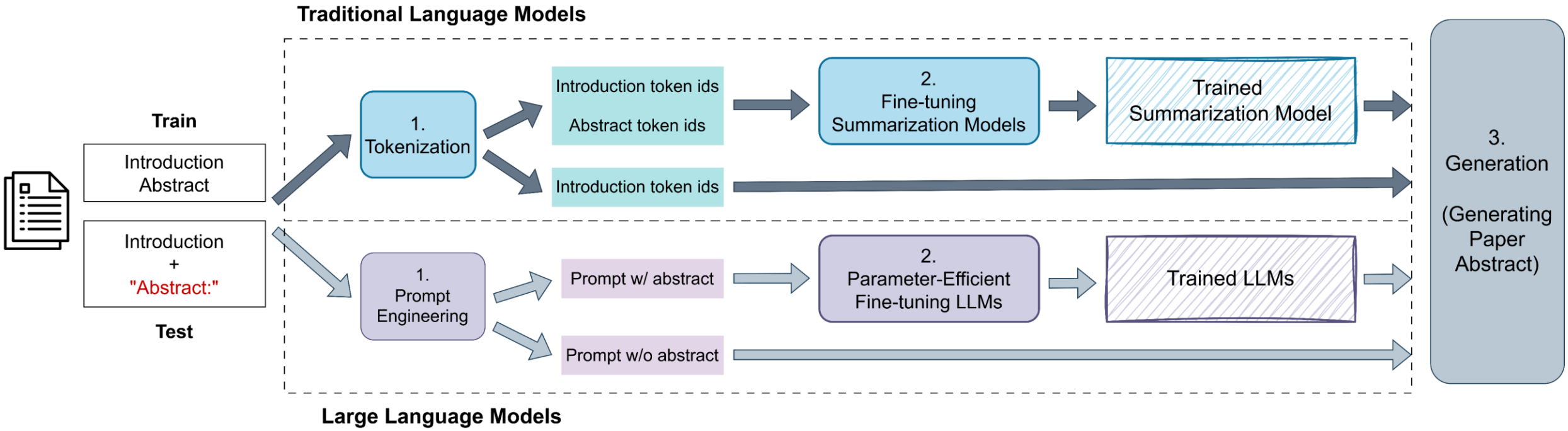
HW2: Paper Abstract Generation

Start Date: 2025/03/18 now
Deadline: 2025/04/08 23:59
[TA] Tzu-Ling Lin (林子淩)

Task Overview



Lab Objective

- In this homework, we aim to **generate paper abstracts from paper introduction bodies**. You should try to improve the quality of generated abstract to increase the performance.
- The data files can be found in the E3 homework section.
- We will have a **two-stages submission**. The first stage is optional and is to check the current performance. The final result should be submitted to the second stage submission, which will be used to calculate your final score.

Dataset Files

- **train.json**: a text file with 408 json lines, where each line represents an individual row of data as follows:
  - **paper_id:** the index of paper.
  - **introduction**: parsed paper introduction in string format.
  - **abstract**: parsed paper abstract in string format.
- **test.jsonl**: a text file with 103 json lines, where each line represents an individual row of data as follows:
  - **paper_id:** the index of paper.
  - **introduction**: parsed paper introduction in string format.
- **sample_submission.json**: a sample submission file with 103 json lines, where there is a header and 103 rows of predictions.
  - **paper_id:** the index of paper.
  - **abstract**: parsed paper abstract in string format.

Submissions for first-stage **(Optional)**

hard **deadline: 03/25, 03/28, 04/01, 04/04, 04/06 23:59**

- Submit the prediction result named **'{student_id}.json'** to **E3**. Please ensure the format follows the **sample_submission.json**.
- It is recommended to submit the prediction within two checkpoint deadlines to check the current performance.

```
{"paper_id": 0, "abstract": "In the realm of reinforcement learning and Monte Carlo methods, a formidable challenge looms: the
{"paper_id": 0, "abstract": "In the realm of reinforcement learning and Monte Carlo methods, a formidable challenge looms: the
{"paper_id": 0, "abstract": "In the realm of reinforcement learning and Monte Carlo methods, a formidable challenge looms: the
{"paper_id": 0, "abstract": "In the realm of reinforcement learning and Monte Carlo methods, a formidable challenge looms: the
{"paper_id": 0, "abstract": "In the realm of reinforcement learning and Monte Carlo methods, a formidable challenge looms: the
{"paper_id": 0, "abstract": "In the realm of reinforcement learning and Monte Carlo methods, a formidable challenge looms: the
{"paper_id": 0, "abstract": "In the realm of reinforcement learning and Monte Carlo methods, a formidable challenge looms: the
{"paper_id": 0, "abstract": "In the realm of reinforcement learning and Monte Carlo methods, a formidable challenge looms: the
{"paper_id": 0, "abstract": "In the realm of reinforcement learning and Monte Carlo methods, a formidable challenge looms: the
{"paper_id": 0, "abstract": "In the realm of reinforcement learning and Monte Carlo methods, a formidable challenge looms: the
```

Submissions for 04/08 23:59 (Deadline)

- Submit the prediction result named **'{student_id}.json'** to **E3**. Please ensure the format follows the **sample_submission.json** or you will get zero point.
- Submit **'{student_id}.zip'** to **E3**. After unzipping, it should appear a folder {student_id} with the following structure:
  - {student_id}
    - {student_id}_code (folder with codes that can reproduce your final result)
      - Model_checkpoint: provide the model checkpoint that can reproduce the test result.
      - requirements.txt: list the required libraries.
    - {student_id}.json: your final result (generated abstracts)
    - readme.md: explain how to reproduce your final .json result.
    - No need to include the datasets. Therefore, please clearly specify the expected location of the datasets. (if included, -5 points)

Evaluation Metrics – Rouge Score

- **ROUGE** counts the number of overlapping units such as n-gram, word sequences, and word pairs between the computer-generated text to be evaluated and the references created by humans.

prediction: `the cat was found under the bed`

reference: `the cat was under the bed`

#	1-gram	reference 1-gram	2-gram	reference 2-gram
1	the	the	the cat	the cat
2	cat	cat	cat was	cat was
3	was	was	was found	was under
4	found	under	found under	under the
5	under	the	under the	the bed
6	the	bed	the bed	
7	bed			
count	7	6	6	5

$$Rouge\_1(X1, Y) = \frac{6}{6} = 1.0 \qquad Rouge\_2(X1, Y) = \frac{4}{5} = 0.8$$

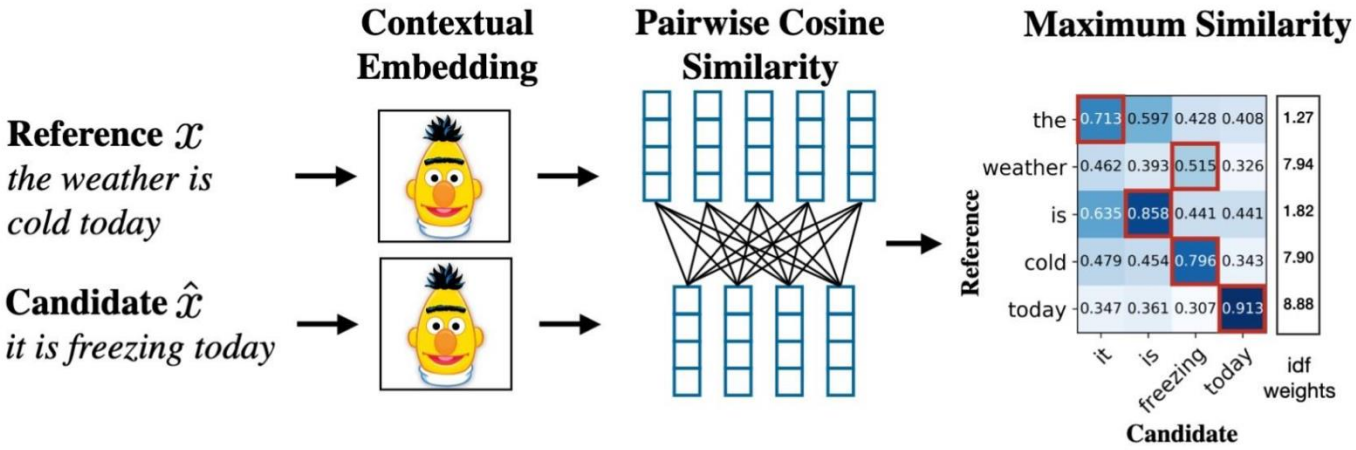
$$F_{\mathrm{ROUGE}} = 2 * \frac{R_{\mathrm{ROUGE}} P_{\mathrm{ROUGE}}}{R_{\mathrm{ROUGE}} + P_{\mathrm{ROUGE}}}$$

$$R_{\mathrm{ROUGE}} = \frac{\#\ of\ matched\ N-grams}{\#\ of\ N-grams\ in\ reference}$$

$$P_{\mathrm{ROUGE}} = \frac{\#\ of\ matched\ N-grams}{\#\ of\ N-grams\ in\ prediction}$$

Evaluation Metrics – BERTScore

- **BERTScore** computes a similarity score for each token in the candidate sentence with each token in the reference sentence using contextual embeddings.



$$F_{\text{BERT}} = 2\frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}$$

$$P_{\text{BERT}} = \frac{1}{|\hat{x}|}\sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \mathbf{x}_i^\top \hat{\mathbf{x}}_j$$

$$R_{\text{BERT}} = \frac{1}{|x|}\sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j$$

Grading Policy

- ROUGE (50 points) / BERTScore (50 points)
  - Top 10%: 100%
  - Top 25%: 90%
  - Top 50%: 80%
  - Top 75%: 75%
  - Other: 70%
  - Below baseline: 0%

Baseline

- Here is the baseline performance. Please try to beat it!

Metric	Score
ROUGE-1	0.47
ROUGE-2	0.12
ROUGE-L	0.22
BERTScore F1	0.85

Reference

- Huggingface: https://huggingface.co/docs
- BART summarization: https://medium.com/@sandyeep70/demystifying-text-summarization-with-deep-learning-ce08d99eda97
- LoRA & QLoRA finetuning Llama: https://www.llama.com/docs/how-to-guides/fine-tuning/
- PEFT finetuning: https://github.com/huggingface/peft
- Unsloth finetuning guide: https://docs.unsloth.ai/get-started/fine-tuning-guide

Reference

- Evaluate: https://huggingface.co/docs/evaluate/a_quick_tour

```
>>> import evaluate


>>> metric_rouge = evaluate.load("rouge", rouge_types=["rouge1", "rouge2", "rougeL"])
>>> metric_bertscore = evaluate.load("bertscore")

>>> rouge = metric_rouge.compute(predictions=preds, references=titles, use_stemmer=True)
>>> bertscore = metric_bertscore.compute(predictions=preds, references=titles, lang="en")
```

Rules

- Do not plagiarize. Write your own codes.
- Do not use additional datasets.
- Do not use API in this homework, train by your own.
- Please follow the required submission format; otherwise will result in point deductions.
- No late submission for first-stage submission.
- Late submission for final submission: score*0.8

Contact & Information

- Deadline: 2025/04/08 23:59
- Please post you question on the **E3 forum**
- [TA] Tzu-Ling Lin (林子淩): tzulinglin.11@nycu.edu.tw
- [TA hours] 14:00-15:00 Wed. (Please make an appointment through email first.)