ADL 2025 Homework 3

Retrieval-Augmented Generation

Deadline: 2025/11/03 23:59:59

Update Logs

-

Links

- Homework 3 files
- HW3 討論區

Task & Guides

Task

RAG system

Your tasks:

- Fine-tune retriever & reranker
- Design prompt to optimize the generation performance
- (Bonus) RAG with RL

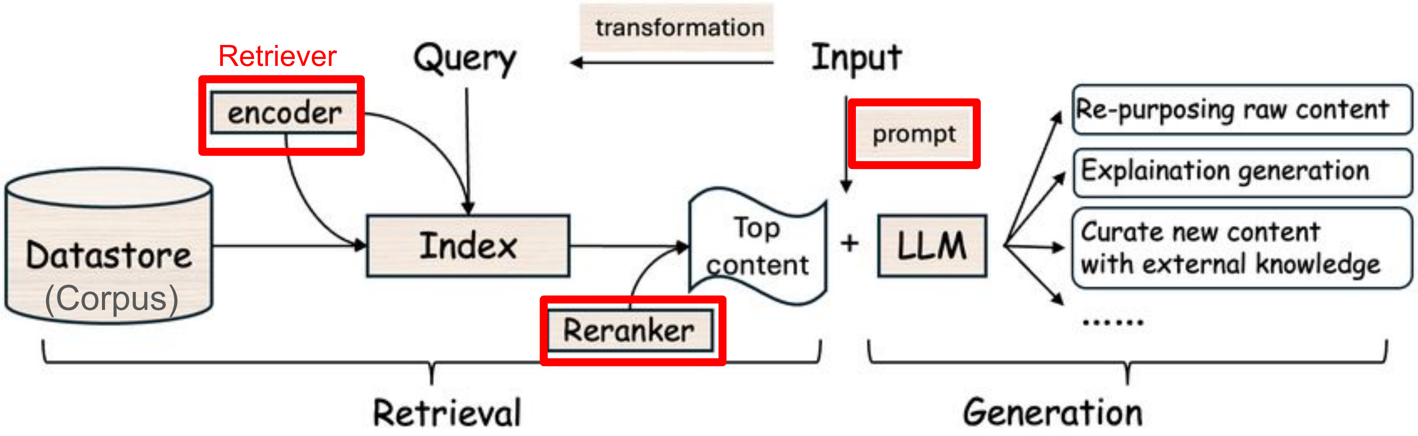
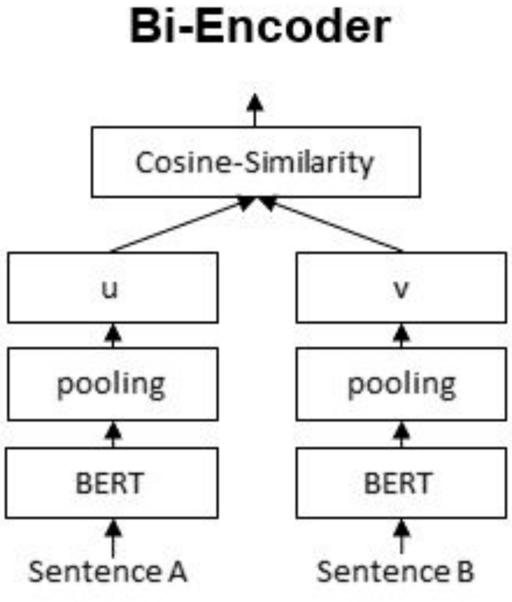


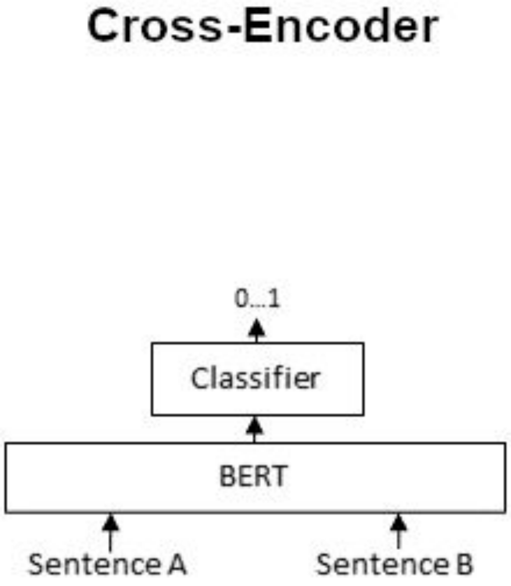
Figure source: https://arxiv.org/abs/2406.06475

Task

Bi-Encoder & Cross-Encoder



retriever



reranker

Figure source: https://sbert.net/examples/cross_encoder/applications/README.html

Task

Model

- Model checkpoint
  - Retriever: intfloat/multilingual-e5-small
  - Reranker: cross-encoder/ms-marco-MiniLM-L-12-v2
  - LLM: Qwen/Qwen3-1.7B (bf16)
- You **CAN ONLY** fine-tune retriever & reranker from the above checkpoint
- You **CANNOT** make any adjustments to the LLM

Task

Files

- data/
    - corpus.txt
    - train.txt
    - test_open.txt
    - qrels.txt
- save_embeddings.py
- inference_batch.py
- utils.py
- readme.md
- requirements.txt

Guides

Dataset

- **Corpus**: passages to be retrieved

  ```
  {"text": "...", "title": "...", "aid": "25749059", "bid": 5, "id": "25749059@5"}
  ```

- **qrels**: mappings of queries and their positive passages

  ```
  {"qid1": {"passageId1": 1},...}
  ```

- Each query has a specific positive passage

Guides

Dataset

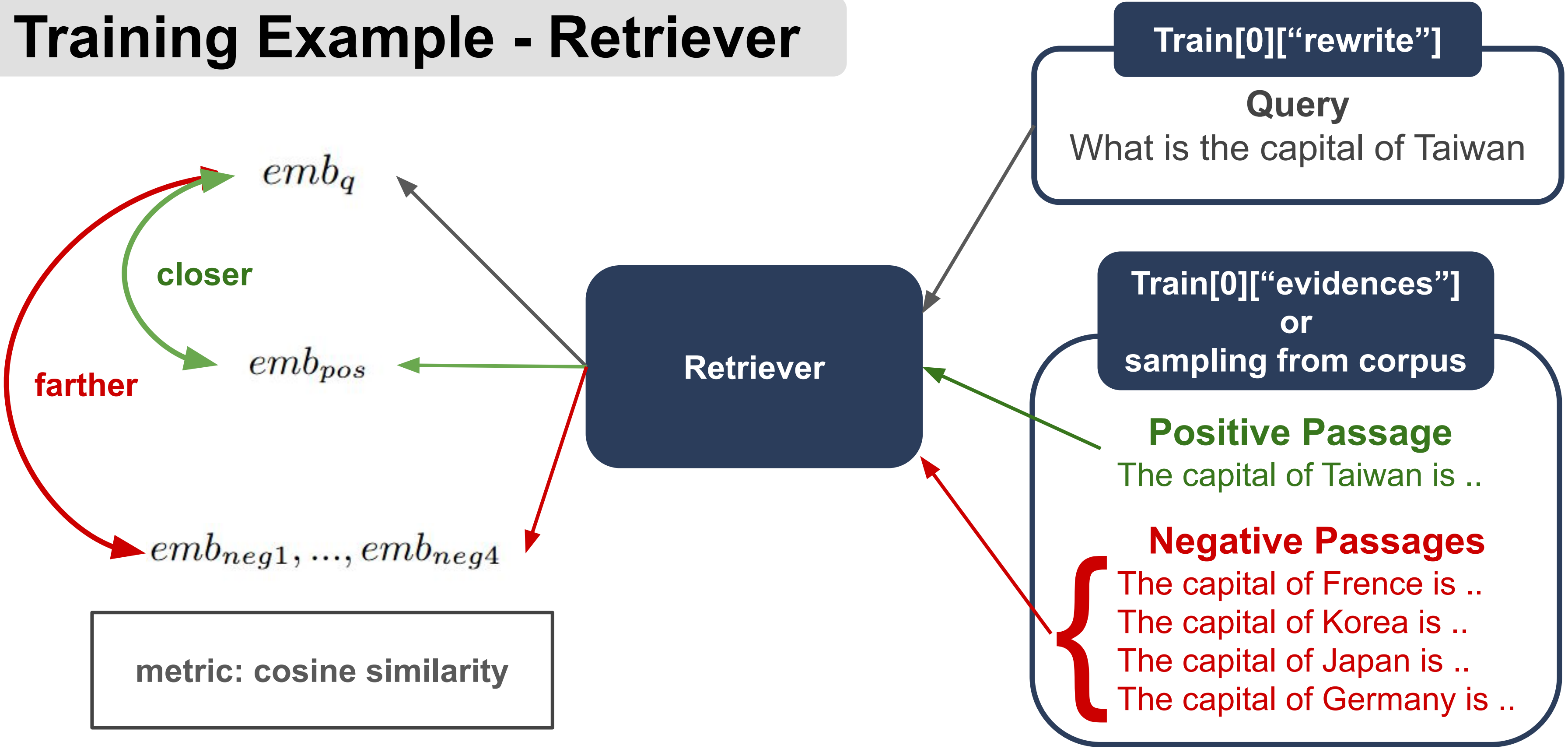
- **train / test_open**: train and public test data

  ```
  {"qid": "...", "rewrite": "...", "evidences": [...], "answer": {"text": "CANNOTANSWER", "answer_start": 0}, "retrieval_labels": [0, 0, 0, 0, 0]}
  ```

  - rewrite: query content
  - evidences: passages from BM25 negative sampling
  - retrieval_labels: corresponding true/false label for passages in “evidences”
- The answer can be **an exact span of positive passage** or **CANNOTANSWER** in both train and test data.
- There can be no positive passage in “evidences” column in `test_open.txt`, however, you can find the positive passage id in `qrels.txt`

Guides

Training Example - Retriever



emb_q

closer

farther

emb_{pos}

$emb_{neg1}, ..., emb_{neg4}$

metric: cosine similarity

Retriever

Train[0][“rewrite”]

Query

What is the capital of Taiwan

Train[0][“evidences”]
or
sampling from corpus

Positive Passage

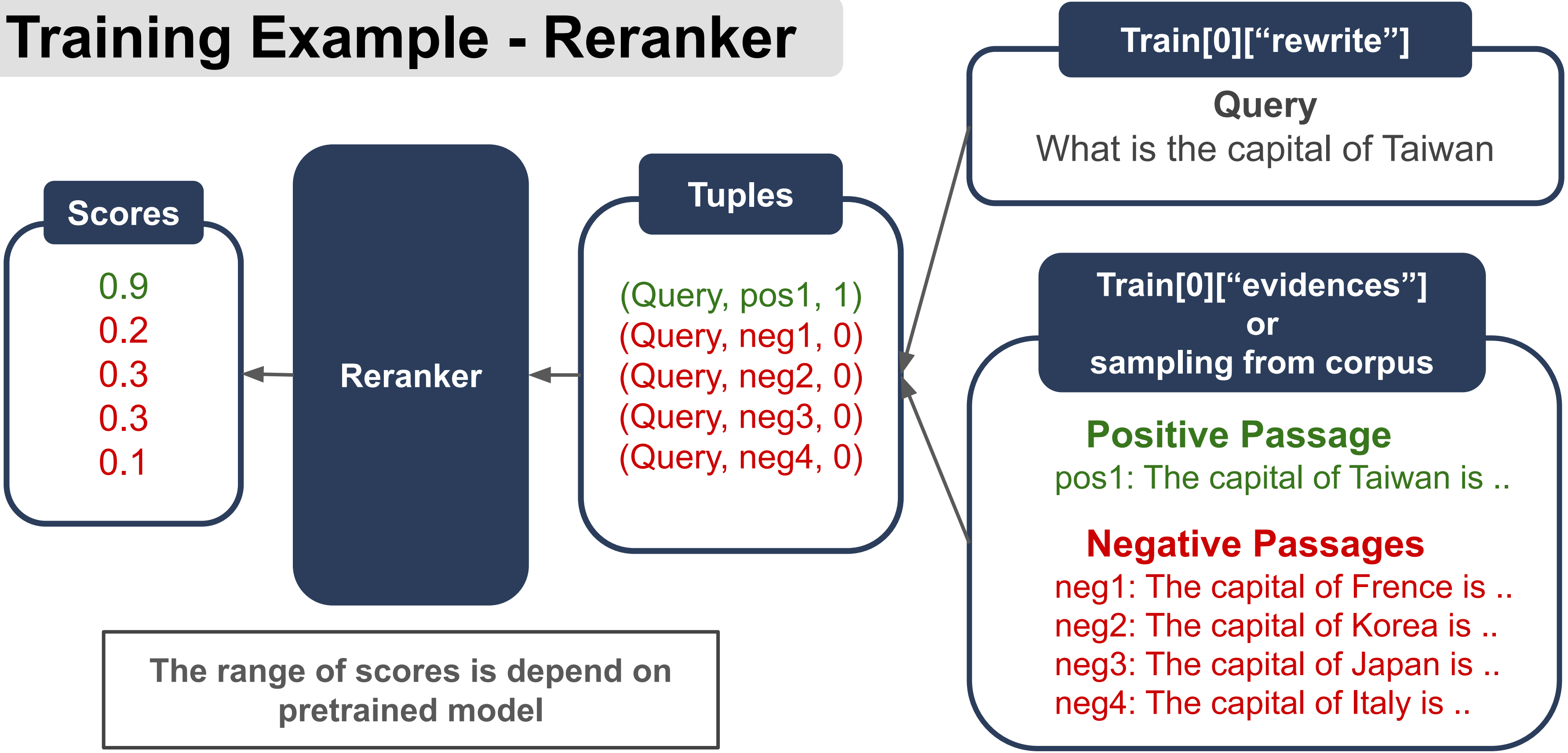
The capital of Taiwan is ..

Negative Passages

The capital of Frence is ..
The capital of Korea is ..
The capital of Japan is ..
The capital of Germany is ..

Guides

Training Example - Reranker



Train[0][“rewrite”]

Query

What is the capital of Taiwan

Train[0][“evidences”]
or
sampling from corpus

Positive Passage

pos1: The capital of Taiwan is ..

Negative Passages

neg1: The capital of Frence is ..
neg2: The capital of Korea is ..
neg3: The capital of Japan is ..
neg4: The capital of Italy is ..

Tuples

(Query, pos1, 1)
(Query, neg1, 0)
(Query, neg2, 0)
(Query, neg3, 0)
(Query, neg4, 0)

Reranker

Scores

0.9
0.2
0.3
0.3
0.1

The range of scores is depend on pretrained model

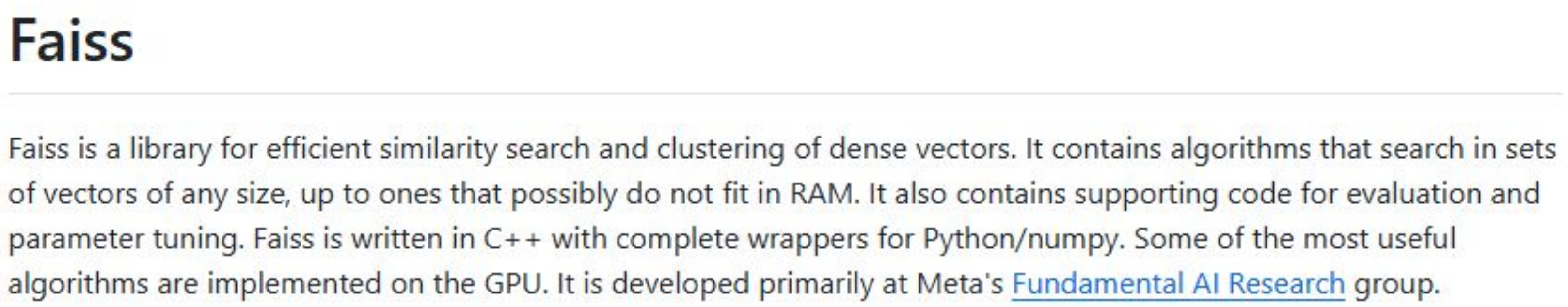
Guides

save_embeddings.py

- Building sqlite DB to store passages, Faiss vector DB to store embeddings

```
$ python save_embeddings.py --retriever_model_path [your_model_path] --build_db
```

- `build_db` flag is for building sqlite passage DB, you can turn it off for only updating your vector DB



Faiss

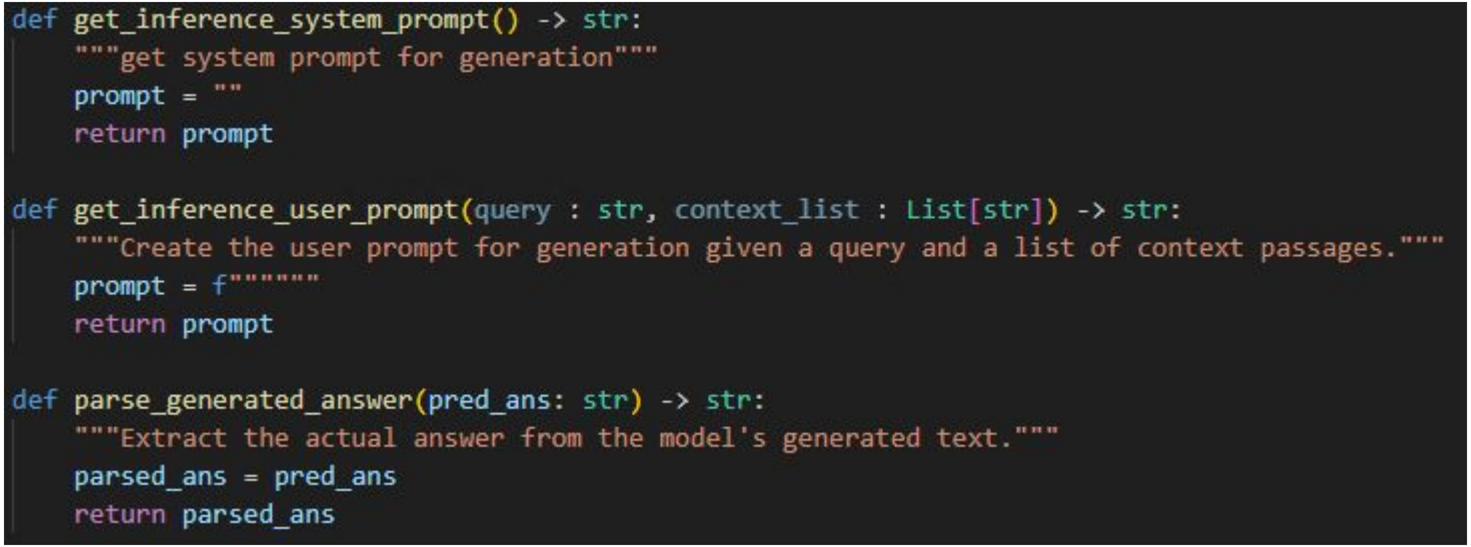
Faiss is a library for efficient similarity search and clustering of dense vectors. It contains algorithms that search in sets of vectors of any size, up to ones that possibly do not fit in RAM. It also contains supporting code for evaluation and parameter tuning. Faiss is written in C++ with complete wrappers for Python/numpy. Some of the most useful algorithms are implemented on the GPU. It is developed primarily at Meta's Fundamental AI Research group.

https://github.com/facebookresearch/faiss

Guides

Prompt Optimization

- Modifying `utils.py` to optimize the generator performance



```python
def get_inference_system_prompt() -> str:
    """get system prompt for generation"""
    prompt = ""
    return prompt

def get_inference_user_prompt(query : str, context_list : List[str]) -> str:
    """Create the user prompt for generation given a query and a list of context passages."""
    prompt = f""""""
    return prompt

def parse_generated_answer(pred_ans: str) -> str:
    """Extract the actual answer from the model's generated text."""
    parsed_ans = pred_ans
    return parsed_ans
```

Guides

inference_batch.py

- Pipeline: Load data ➡ Retrieve ➡ Rerank ➡ Generate ➡ Evaluate
- Import prompt and parse function from `utils.py`
- Run `save_embeddings.py` to build DB before executing this script
- Evaluation metric:
  - Recall@k (k=10): $\text{Recall@k} = \frac{1}{|Q|}\sum_{i=1}^{|Q|} \mathbf{1}(\text{hit}_i \leq k)$
  - MRR@k (k=10): $\text{MRR@k} = \frac{1}{|Q|}\sum_{i=1}^{|Q|} \frac{\mathbf{1}(\text{rank}_i \leq k)}{\text{rank}_i}$
  - Sentence similarity from bi-encoder: `sentence-transformers/all-MiniLM-L6-v2`
- **DO NOT** modify this file during testing

```
$ python inference_batch.py --retriever_model_path [your_model_path]
--reranker_model_path [your_model_path] --test_data_path [test_data_path]
```

Rules & Submission

Deadline: 2025/11/03 23:59:59

Rules

Environment

The following packages should suffice for this assignment. Please email us for approval if you need additional ones. However, you can use other packages to plot or do testing in your own local

- python == 3.12
- packages
    - transformers==4.56.1
    - torch==2.8.0
    - datasets==4.0.0
    - tqdm==4.67.1
    - faiss-gpu-cu12==1.12.0
    - sentence-transformers==5.1.0
    - python-dotenv==1.1.1
    - accelerate==1.10.1
    - gdown

Rules

What You CANNOT Do

- Use external training data.
- Fine-tune the model on test data.
- Try to find private test data.
- Any means of cheating or plagiarism, including but not limited to:
  - Directly apply others' published / unpublished code…, including the codes written your classmates or public on the internet
  - Give/get trained model/predictions to/from others.
  - Give/get report answers or plots to/from others.
  - Publish your code before deadline.
- Violations may cause zero/negative score and punishment from school.

Submission

File Layout

- Zip your folder, which should be named as your student id (lower-cased) (ex. r13000000) and **submit the .zip to NTU COOL**.

```
r13000000.zip
|---r13000000/  (Should contain this folder, not separate files)
|    |- download.sh
|    |- utils.py
|    |- run.sh
|    |- report.pdf
|    |- README.md
|    |-  code/script (all the code/script you used to train, predict, or plot report figures should be included)
```

Submission

download.sh

- `download.sh` should download and create a folder called `models`
    - **!!DO NOT** download original model checkpoint!!
- **DO NOT** modify your file after deadline, or it will be seen as cheating.
- Keep the URLs in `download.sh` valid for at least 3 weeks after deadline.
- You can download at most 2G, and `download.sh` should finish within 1 hour. (At csie dept with maximum 10MB/s bandwidth)
- **DO NOT** do things more than downloading. Otherwise, your `download.sh` may be killed.
- **DO NOT** pip install ANYTHING in your `download.sh`, you are not allowed to modify the testing environment
- We will execute `download.sh` before inference scripts.

Submission

File Layout (after downloading)

- After we run `download.sh`, the `retriever/` and `reranker/` should be in `models/`

```
r13000000.zip
|---r13000000/  (Should contain this folder, not separate files)
     |- download.sh
     |- utils.py
     |- run.sh
     |- report.pdf
     |- README.md
     |- code/script (all the code/script you used to train, predict, or plot report figures should be included)
     |- models/
         |- retriever/
             |- model.safetensors…
         |- reranker/
             |- model.safetensors…
```

Submission

README.md

- `README.md` should contain step-by-step instructions on how to setup your environments and how to train your model with your `codes/scripts`.
- You will lose **2%** score if you have no or empty `README.md`.
- If necessary, you will be required to reproduce your results based on the `README.md`.
- If you cannot reproduce your result, you may lose points.

Submission

Execution Environment

- We will run the testing codes on the computer with
  - OS: Ubuntu 20.04
  - Hardware: 32GB RAM, RTX 3070 **8GB** VRAM, and 20GB disk space available
- Same conda environment with python version and packages in p.9

Submission

Grading

- Model Performance (10%)
  - Public baseline:
    - Recall@10 = **0.780** ↑ (2%)
    - MRR@10 = **0.695** ↑ (2%)
    - Sentence_similarity = **0.340** ↑ (2%)
  - Private baseline:
    - Recall@10 = ? ↑ (1%)
    - MRR@10 = ? ↑ (1%)
    - Sentence_similarity = ? ↑ (2%)
- Report (10% + 2%)
- Format
  - You may lose 2% score if your file structure is wrong

Submission

Grading - inference_batch.py

- TA will use our own `save_embeddings.py` and `inference_batch.py` (which is the same as we published), so you don’t have to upload these files

```
$ python save_embeddings.py --retriever_model_path ./models/retriever --build_db
```

```
$ python inference_batch.py --retriever_model_path ./models/retriever
--reranker_model_path ./models/retanker --test_data_path ./data/test_open.txt
```

Submission

Late Submission

- Deadline: **2025/11/03 23:59:59**
- Late submission penalties:
  - **0 day < late submission ≦ 1 day**: original score * **0.90**
  - **1 day < late submission ≦ 3 day:** original score * **0.70**
  - **3 day < late submission**: original score * **0.0**
- We only consider your last submission in NTU COOL.
  - Update your submission after deadline implies that you will get penalty.

Report

Report

Q1: Retriever & Reranker Tuning (5%)

Retriever Training Process (2.5%) and Reranker Training Process (2.5%). Both should include, but are not limited to, the following:

- Clearly describe how the training data is constructed (e.g., anchor, positive sampling, and negative sampling strategies).
- Specify and explain the loss function used for training.
- List the hyperparameters adopted in your experiments.
- Provide a training loss curve with at least 5 data points to illustrate the training process. (You can use any package to plot)

Report

Q2: Prompt Optimization (3%)

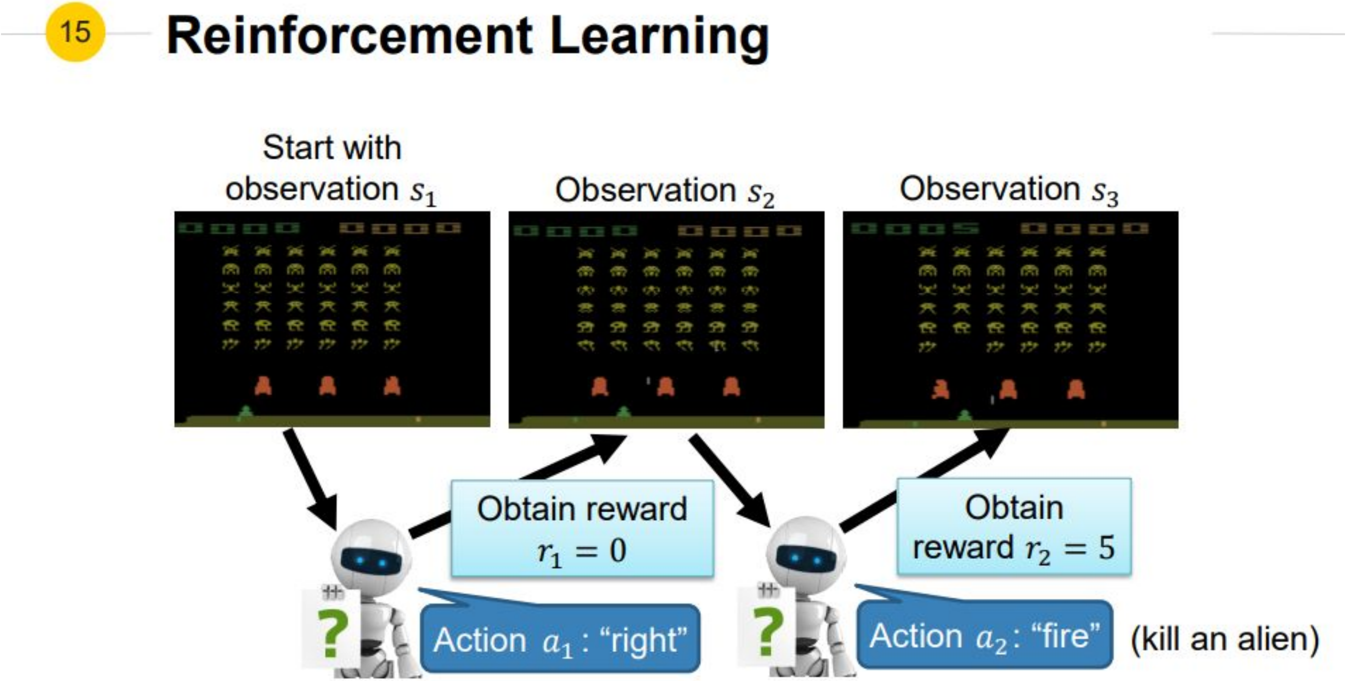
- Provide a detailed explanation of how you designed your prompt. (1.5%)
- Present at least three different prompts you experimented with, along with their inference performance, to demonstrate the effectiveness of your optimization approach. (1.5%)

Q3: Additional Analysis (2%)

- Show any interesting analysis using plots or experimental data obtained during inference. (2%)
- For example:
  - A comparison of model performance when the correct answer is retrieved versus when it is not
  - A comparison of performance with and without a reranker
  - ...
- Scores will be assigned based on the richness and depth of your analysis.

Report

Bonus: RL in the loop (2%)



15 Reinforcement Learning

Start with observation s_1

Observation s_2

Observation s_3

Obtain reward $r_1 = 0$

Obtain reward $r_2 = 5$

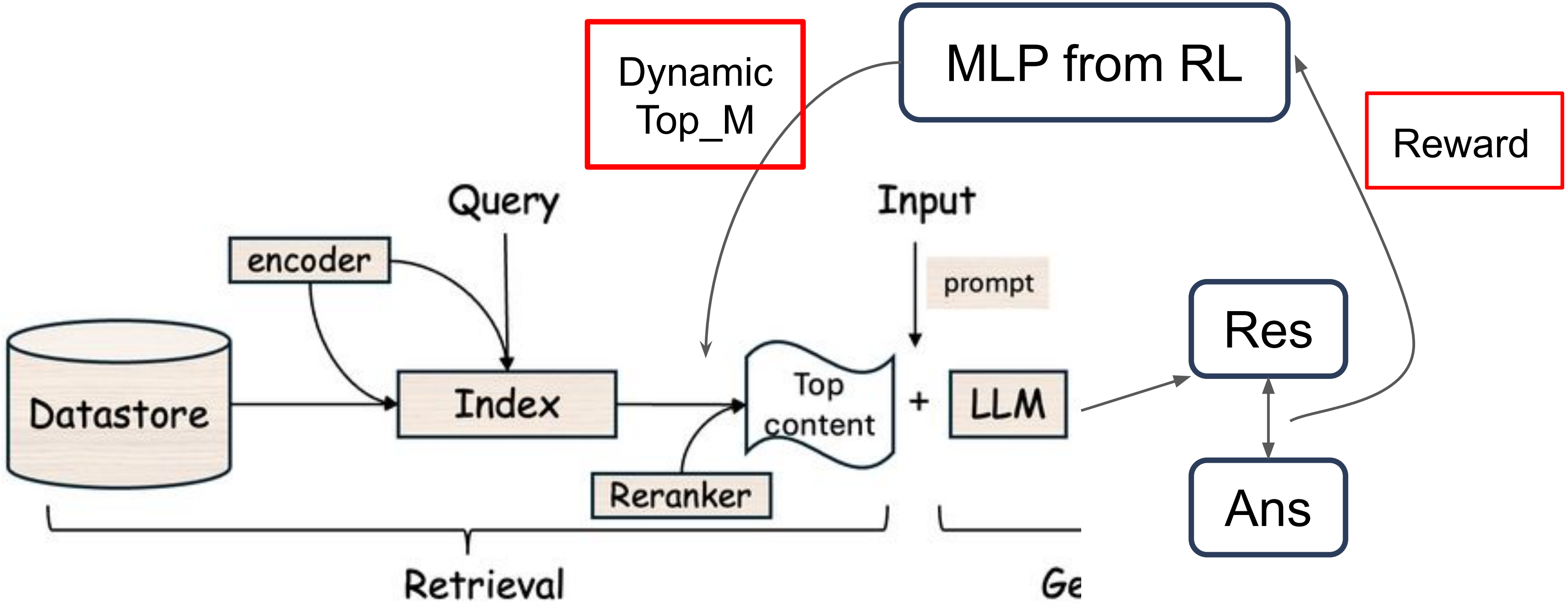
Action a_1: “right”

Action a_2: “fire”

(kill an alien)

Report

Bonus: RL in the loop (2%)



Report

Bonus: RL in the loop (2%)

Use **Reinforcement Learning** to train a model deciding the number of passages to include in the prompt.

- Describe your training method and experimental setting to compare with the original results.
- Submit your training & inference script.
- It is recommended to use the *stable-baselines3* and *gymnasium* packages, as they provide a simple framework for training and environment setup.

Reference

- https://stable-baselines3.readthedocs.io/en/master/
- https://gymnasium.farama.org/index.html
- https://arxiv.org/abs/2406.06475
- https://github.com/facebookresearch/faiss
- https://aclanthology.org/W04-1013.pdf
- https://en.wikipedia.org/wiki/Mean_reciprocal_rank
- https://sbert.net/examples/cross_encoder/applications/README.html

Any Question

- NTU COOL discussion
  - We kindly ask you to check the discussion forum first before emailing us.
- Email:
  - adl-ta@csie.ntu.edu.tw
  - Please prefix your email subject with **[ADL2025 HW3]** for faster response.
- TA hours at CSIE Building (德田館) R524
  - Thursday 11:00 ~ 12:00 (TA: 彭紀綸)
  - Friday 14:00 ~ 15:00 (TA: 陳剛頡)