Hallucineers Team Technical Report

Submission Details

Task 1:

- **Submission File Name**: `submission_1.json`

  Description: Rerank range score = 0.5, retrieve range scores = 0.05, alpha = 0.3, model_1_weight = 0.5, model_2_weight = 0.5

- **Submission File Name**: `submission_2.json`

  Description: Rerank range score = 2.0, retrieve range scores = 0.1, alpha = 0.3, model_1_weight = 0.5, model_2_weight = 0.5

- **Submission File Name**: `submission_3.json`

  Description: Rerank range score = 3.0, retrieve range scores = 0.1, alpha = 0.3, model_1_weight = 0.5, model_2_weight = 0.5

Task 2:

- **Submission File Name**: `submission1.json`
- **Description**: Combined the answers from two SFT models: `qwen_3e` and `qwen_14e`.
- **Submission File Name**: `submission2.json`
- **Description**: An ensemble of the **three GPRO-trained models** using a voting mechanism to determine the final answer.
- **Submission File Name**: `submission3.json`
- **Description**: An ensemble combining the **three GPRO models and the `qwen_14e` SFT model** (total of 4 models), also using a voting mechanism.

Environment Setup

1. **Prerequisites**:

- Python 3.8 or higher
- CUDA-compatible GPU (recommended)
- pip (Python package manager)

2. **Create and activate a virtual environment (recommended)**:

```
python -m venv venv
source venv/bin/activate  # On Windows: venv\Scripts\activate
```

3. **Install dependencies**:

```
cd retrieval
pip install -r requirements.txt
```

Running the Code

Task 1: Legal Document Retrieval

1. Data Processing

Process the main competition data:

```
python process_data.py
```

Process additional Zalo data:

```
python process_data.py --data-path ../ALQAC_2025_data/additional_data/zal
```

2. BM25 Model Training

Train BM25 on main data:

```
python bm25_train.py
```

Train BM25 on Zalo data:

```
python bm25_train.py --data-path ../ALQAC_2025_data/additional_data/zalo
```

3. Generate Training Pairs

Create pairs for main data with re-ranking:

```
python bm25_create_pairs.py --rerank
```

Create pairs for Zalo data:

```
python bm25_create_pairs.py --rerank --data_path ../ALQAC_2025_data/addit
```

4. Fine-tune Sentence Transformers

Train on Zalo data with Vietnamese embedding model:

```
python train_sentence_transformer.py --pretrained_model AITeamVN/Vietname
--step 200 --round 1 --saved_model sbert_saved_model --eval_size 0.0 \
--pair_data_path pair_data_zalo/bm_25_pairs_top50 --batch_size 16 --data_
```

Train with word segmentation:

```
python train_sentence_transformer.py --pretrained_model huyydangg/DEk21_h
--step 200 --round 1 --wseg --saved_model sbert_wseg_saved_model --eval_s
--pair_data_path pair_data/bm_25_pairs_top50 --batch_size 16 --data_path
```

5. Train Reranker

Install FlagEmbedding library:

```
pip install -U FlagEmbedding[finetune]
```

Prepare data for reranker:

```
python bm25_create_pairs.py --rerank --data_path ../ALQAC_2025_data/addit
python bm25_create_pairs.py --rerank --data_path ../ALQAC_2025_data --sav
```

Train reranker on Zalo data first:

```
torchrun --nproc_per_node 2 \
    -m FlagEmbedding.finetune.reranker.encoder_only.base \
```

```
    --model_name_or_path AITeamVN/Vietnamese_Reranker \
    --cache_dir ./cache_zalo/model \
    --train_data ./pair_data_zalo/rerank_data_top20.jsonl \
    --cache_path ./cache_zalo/data \
    --train_group_size 8 \
    --query_max_len 256 \
    --passage_max_len 2048 \
    --output_dir ./zalo-legal-reranker \
    --learning_rate 6e-5 \
    --num_train_epochs 4 \
    --per_device_train_batch_size 6
```

Then fine-tune on main data:

```
torchrun --nproc_per_node 2 \
    -m FlagEmbedding.finetune.reranker.encoder_only.base \
    --model_name_or_path phonghoccode/ALQAC_2025_Reranker_top20_zalo \
    --cache_dir ./cache/model \
    --train_data ./pair_data/rerank_data_top20.jsonl \
    --cache_path ./cache/data \
    --output_dir ./legal-reranker \
    --learning_rate 6e-5 \
    --num_train_epochs 3 \
    --per_device_train_batch_size 4
```

7. Evaluation

Evaluate the retrieval system's performance using the following command:

```
python eval.py \
    --raw_data ../ALQAC_2025_data \
    --model_1_weight 0.5 \
    --model_2_weight 0.5 \
    --hybrid \
    --combine_type weighted_sum \
    --rerank_range_score 0.5 \
    --retrieve_range_scores 0.05 \
    --alpha 0.3 \
    --eval_size 0.2 \
    --encode_legal_data \
    --create_law_mapping
```

8. Grid search

Grid search the best parameters using the following command:

```
python eval.py \
    --raw_data ../ALQAC_2025_data \
    --model_1_weight 0.5 \
    --model_2_weight 0.5 \
    --find-best-score \
    --hybrid \
    --combine_type weighted_sum \
    --alpha 0.3 \
    --eval_size 0.2 \
    --encode_legal_data \
    --encode_question_data \
    --create_law_mapping
```

9. Inference

Run inference on new questions using the following command:

```
python inference.py \
    --raw_data ../ALQAC_2025_data \
    --model_1_weight 0.5 \
    --model_2_weight 0.5 \
    --hybrid \
    --combine_type weighted_sum \
    --rerank_range_score 0.5 \
    --retrieve_range_scores 0.05 \
    --alpha 0.3 \
    --encode_legal_data \
    --encode_question_data \
    --create_law_mapping \
    --output_file predictions
```

Task 2: Legal Question Answering

- **Data Preparation (Generating the "think" component):**
  To augment the dataset with the reasoning part, run the notebook:
  `alqac-gen-data-train.ipynb`

- **SFT Training:**
  To perform Supervised Fine-Tuning (SFT) on the full dataset, run:
  `alqac-train-data-train-qwen-fulldata.ipynb`

- **GPRO Training:**
  To train the model using GPRO with the three different configurations, run these notebooks:

  - `alqac-train-grpo-config1.ipynb`
  - `alqac-train-grpo-config2.ipynb`
  - `alqac-train-grpo-config3.ipynb`

- **Inference:**

  - To perform inference on the test set using the SFT model, run:
    `alqac-inference-test-qwen-fulldata.ipynb`
  - To perform inference on the test set using the GPRO models, run:
    `alqac-inference-test-qwen-grpo-fulldata.ipynb`

- **Generate Submission Files:**
  To create the three final submission files by ensembling the model outputs, run:
  `submit.ipynb`