Topic Modeling Documentation

Author: Ankit Agrawal, Email: ankitagr.nitb@gmail.com

Overview:

In this project, we address the challenge of topic modeling on abstracts of papers from the CShorten/ML-ArXiv-Papers dataset. Due to the lack of high-quality labeled data (and the idea of creating topics dynamically), we employ a knowledge distillation approach. A large teacher model (Llama-3-70B) generates labels for the dataset, and a smaller student model (Llama-3-8B) is fine-tuned on this data. The fine-tuned student model is then used for inference.

Approach:

Knowledge Distillation:

We use a knowledge distillation approach for the following reasons:

- To generate high-quality training data.
- To enable the use of a smaller, fine-tuned model for inference that performs similarly to the large teacher model.

Files: (Huggingface hub)

Dataset:
Train/test dataset generated using Teacher Model (Llama-3-70b_Instruct):
Train: ankitagr01/dynamic_topic_modeling_arxiv_abstracts (15k samples)
Test: ankitagr01/dynamic_topic_modeling_arxiv_abstract_1k (1k samples)

Finetuned Model:
ankitagr01/llama_3_8b_ft_topic_new

Challenges:

1. **Data for training:** Existing public datasets are either designed for fixed classes of topics or contain only single keyword topics (Suitable for traditional approaches like LDA, tf-idf).

2. **Defining proper prompt:** Crafting effective prompts for generating responses from the teacher model is crucial.

3. **GPU allocation:** Long wait time for large GPUs (A100-80GB, H100-80GB) allocation.

4. **Train/Eval time:** Was restricted due to availability of GPUs and training/eval time. Around 1.5 hours to load Llama-3-70b. Similarly, loading time for llama-3-8b.

5. **Llama-3 instability:** Llama-3 models are very instable, till now not much better results have been achieved on finetuning llama-3 models. ((https://www.reddit.com/r/LocalLLaMA/comments/1cwwgkz/is_llama_3_just_not_a_good_model_for_finetuning/)

6. **Unsloth library:** Had problems with using the unsloth library due to dependencies issues. (This took a lot of my time to try and fix, but unfortunately with the GPU clusters I am using, it was not very supported. Hence, I decided later not to proceed with it currently.)

'Unsloth is free and Apache 2 open source licensed, is 2.2x faster, uses 70% less VRAM, has 0% degradation in accuracy for QLoRA (4bit) and LoRA (16bit) finetuning.'' inference 2x faster natively.'

Datasets:

Used CShorten/ML-ArXiv-Papers dataset which consists of Abstract from different research papers from Arxiv.
Topics were generated for the abstracts using the teacher model.

Total data used:
Train + val: 15000 samples
Test: 1000 samples.

Training:

Used ROUGE as a **custom metric** during the finetuning process.
GPU used: H100 80GB
Epoch: 3

{'train_runtime': 4318.4911, 'train_samples_per_second': 1.526, 'train_steps_per_second': 0.047, 'train_loss': 1.5356604936076146, 'epoch': 2.97}

Evaluation:

We use BLUE-3 and ROUGE scores as our evaluation metrics to assess the performance of our models.

Choice of models:

- **Teacher model:** Llama-3-70B-Instruct
- **Student model:** Llama-3-8B-Instruct

We evaluate the following versions of the student model against the teacher model predictions:

- Llama-3-8B-Instruct (Pre-trained)
- Llama-3-8B-Instruct (Few-shot Prompt-tuning)
- Llama-3-8B-Instruct (Fine-tuned)

Results:

The table below presents the performance (BLEU and ROUGE scores) of our models:

Model	BLEU-3	ROUGE-1 (F1)	Inference time per sample
Llama-3-8B-Instruct (Pre-trained)	42.11	51.58	0.25s
Llama-3-8B-Instruct (Few-shot)	39.83	**53.91**	0.3s
Llama-3-8B-Instruct (Fine-tuned)	**44.44**	53.13	0.4s

We see the performance of all 3 model variants are similar to each other.

Recently, with all the finetuned versions of Llama-3 models, significant performance improvements have not been achieved. (https://www.reddit.com/r/LocalLLaMA/comments/1cwwgkz/is_llama_3_just_not_a_good_model_for_finetuning/)

One solution is using the Unsloth library, which has been shown to improve performance, however, I was not able to use it currently in my GPU cluster because of dependency issues.

Resource Analysis

GPU used: A100-80GB, H100-80GB

GPU Utilization

- **Teacher Model (Llama-3-70B-Instruct)**
  - **GPU Memory Required**: 42.8 GB
  - **Inference Time**: 1.2 s/sample
  -

- **Student Model (Llama-3-8B-Instruct)**
    - **Fine-tuning**:
        - **GPU Memory Required**: 33.8 GB (batch size 4)
        - **Total Training Time**: 90 mins (15k samples)
    - **Inference**:
        - **GPU Memory Required**: 32.7 GB
        - **Inference Time**: 0.4 s/sample

Improvements:

1. **Hyperparameter Tuning**: Further tuning to optimize model performance.
2. **Prompt Tuning**: Experimenting with zero-shot and few-shot prompting techniques.
3. **Human Evaluation**: Incorporating human feedback for model evaluation and retraining, or generating training data using human annotations.
4. **Large train dataset**: Right now only training with 15k training sample, which is insufficient for Llama-3 finetuning.
5. **Using custom embedding-based metrics** for training and eval. (Similarity score or distance between embeddings between predicted and ground-truth topic)
6. **Unsloth:** Using Unsloth package for faster inference and stable finetuning.
7. **Using GGUF** version of Llama-3-8b for finetuning. This has proven to achieve better finetuning results.
8. **Using Flash attention.**
9. **Other LLM models**: Evaluate with other choice of open-source free LLMs to check which is better (Gemma, Mistral, etc)
10. **Diverse Domain Training**: Expanding the training dataset to include various domains (legal documents, books, newspapers, tweets, reviews) for better generalization.
11. **Scalability Enhancements**: Implementing multiprocessing or using multiple GPUs (e.g., DataParallel or DistributedDataParallel) to scale.
12. **Documentation Tools**: Utilizing Doxygen for better documentation.

Scalability:

1. **Support for large documents:**
   a. **Chunking:** Splitting documents into chunks, generating topics for each, and combining them for the final topic.
   b. **Large Context Models:** Use LLM with large context sizes.

2. **Multi-lingual support:** Training the model on datasets in multiple languages to enhance robustness.

3. **Multi-modal support:** Topic modeling can also be applied to images/videos/audio or a combination of them. Combining different modalities can help to make the context very rich.

Tech-stack:

- **Programming Language:** Python
- **Deep Learning Framework:** PyTorch
- **Tools:**
  - Gafarna
  - WandB
  - Slurm
  - Huggingface
  - **Development Environment:** VSCode with SSH connection

TODO:

- Add training plots, logs from wandb.
- Instructions to run.
- Docker