perplexity

Think of yourself as a great software developer, writer and presenter/toastmaster who understands the nuances of English and Simplified Chinese and can speak convincingly like a native speaker. Always print/output your response using WhatsApp-style of text formatting to improve readability for your users. You will answer with appropriate technical depth, yet in a language that most people can understand. You can present your thoughts concisely and clearly, and think and brainstorm creatively. You will carefully review and evaluate each reported problem/bug or message and then think deeply and carefully about a solution before recommending it. You will try to simulate and test any generated code or script before replying.

Background:
As an "old" and thus very seasoned/experienced medical doctor, I have accumulated many medical diagnosis reports over many years and I would like to use them to train a suitable open source LLM from the huggingface repository to help with my practice, meaning specialized "medically trained" LLM for private use.

What I need you to do:
carefully review the content of this link https://chatgpt.com/share/67ac8f75-ca14-8000-a0d1-0746fbc03ad3
use the information contained in the link together with further web searches to confirm and validate the information, then do a deep dive research on how to choose a suitable LLM of medium size (32GB to 70GB), which suffix type ("SFT", instilled, "instruct", "base", "GRPO", etc.) should I choose. Also also give me a detailed step-by-step guide on the training methodology to follow for training the chosen LLM. The guide should be very detailed enough and yet easy to understand and thus follow without further questions. As your final output, the guide should be in a presentation style document with 2000 to 3000 words and formatted in WhatsApp style text formatting.

Choosing and Training a Suitable LLM for Medical Practice

Introduction

As an experienced medical professional, you've accumulated a wealth of medical diagnosis reports over the years. Your goal is to leverage this valuable data to train an open-source LLM from the Hugging Face repository for specialized medical use in your practice. This guide will help you select an appropriate medium-sized LLM (32GB to 70GB) and provide a detailed step-by-step training methodology.

1. Choosing the Right LLM

When selecting an LLM for medical applications, consider the following factors:

- *Size*: 32GB to 70GB (as per your requirement)
- *Domain specificity*: Pre-trained on medical data
- *Performance*: Accuracy and efficiency
- *Ethical considerations*: Bias mitigation and safety
- *Scalability*: Ability to handle your expected query volume
- *Suffix type*: Indicates the model's specialization

Based on these criteria, here are some suitable options:

1. *BioGPT* (base)
    - Size: ~45GB
    - Pre-trained on biomedical literature
    - Good performance on medical tasks
2. *PubMedBERT* (base)
    - Size: ~40GB
    - Trained on PubMed abstracts and full-text articles
    - Excellent for biomedical NLP tasks

3. *ClinicalBERT* (SFT)
    - Size: ~35GB
    - Fine-tuned on clinical notes
    - Specialized for clinical applications
4. *BioBERT* (base)
    - Size: ~50GB
    - Pre-trained on biomedical corpora
    - Strong performance in biomedical text mining

Recommended choice: ClinicalBERT (SFT)

Rationale:

- *Size*: Fits within your specified range
- *Suffix*: SFT (Supervised Fine-Tuning) indicates it's already fine-tuned for specific tasks
- *Specialization*: Focused on clinical applications, aligning with your medical practice needs

2. Understanding Model Suffixes

- *Base*: Pre-trained model without task-specific fine-tuning
- *SFT* (Supervised Fine-Tuning): Fine-tuned on labeled data for specific tasks
- *Instruct/Instilled*: Trained to follow instructions or prompts
- *GPTO* (Generative Pre-trained Transformer): Optimized for text generation tasks

For medical applications, SFT models are often preferred as they're already fine-tuned on relevant data [1] [2].

3. Step-by-Step Training Methodology

Step 1: Data Preparation

a) *Collect and organize your medical diagnosis reports*

- Ensure patient privacy by removing identifiable information
- Standardize the format of your reports

b) *Clean and preprocess the data*

- Remove irrelevant information
- Correct spelling and formatting errors
- Tokenize the text (split into individual words or subwords)

c) *Split the data*

- Training set (70-80%)
- Validation set (10-15%)
- Test set (10-15%)

Step 2: Set Up Your Environment

a) *Install required libraries*

```
pip install transformers datasets torch
```

b) *Import necessary modules*

```
from transformers import AutoTokenizer, AutoModelForCausalLM, TrainingArguments, Trainer
from datasets import load_dataset
```

c) *Load the pre-trained ClinicalBERT model and tokenizer*

```
model_name = "emilyalsentzer/Bio_ClinicalBERT"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(model_name)
```

Step 3: Prepare the Dataset

a) *Load your preprocessed data*

```
dataset = load_dataset("path/to/your/data")
```

b) *Tokenize the dataset*

```
def tokenize_function(examples):
    return tokenizer(examples["text"], padding="max_length", truncation=True)

tokenized_datasets = dataset.map(tokenize_function, batched=True)
```

Step 4: Fine-Tuning

a) *Set up training arguments*

```
training_args = TrainingArguments(
    output_dir="./results",
    num_train_epochs=3,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    warmup_steps=500,
    weight_decay=0.01,
    logging_dir="./logs",
)
```

b) *Create Trainer instance*

```
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_datasets["train"],
    eval_dataset=tokenized_datasets["validation"]
)
```

c) *Start training*

```
trainer.train()
```

Step 5: Evaluation

a) *Evaluate on test set*

```
test_results = trainer.evaluate(tokenized_datasets["test"])
print(test_results)
```

b) *Analyze performance metrics*

- Perplexity
- Accuracy
- F1 score (for classification tasks)

Step 6: Iterative Improvement

a) *Error analysis*

- Identify common mistakes or biases in the model's outputs

b) *Hyperparameter tuning*

- Adjust learning rate, batch size, or number of epochs

c) *Data augmentation*

- Add more diverse medical reports if available

Step 7: Deployment

a) *Save the fine-tuned model*

```
model.save_pretrained("./my_medical_model")
tokenizer.save_pretrained("./my_medical_model")
```

b) *Set up inference pipeline*

```
from transformers import pipeline
```

```
medical_nlp = pipeline("text-generation", model="./my_medical_model")
```

c) *Test with sample queries*

```
result = medical_nlp("Patient presents with symptoms of")
print(result)
```

4. Best Practices and Considerations

- *Ethical considerations*:
- Ensure patient privacy and data security
- Be aware of potential biases in the training data
- Implement safeguards against generating harmful or incorrect medical advice
- *Continuous learning*:
- Regularly update the model with new medical reports and findings
- Stay informed about advancements in medical AI and LLMs
- *Human oversight*:
- Always have a medical professional review the model's outputs
- Use the LLM as a supportive tool, not a replacement for expert judgment
- *Documentation*:
- Keep detailed records of the training process and model versions
- Document any limitations or known issues with the model

5. Potential Challenges and Solutions

- *Challenge*: Limited computational resources
  Solution: Use cloud-based GPU services or distributed training
- *Challenge*: Overfitting to specific medical cases
  Solution: Implement regularization techniques and diverse data augmentation
- *Challenge*: Handling rare medical conditions
  Solution: Incorporate few-shot learning techniques and expert knowledge

Conclusion

By following this guide, you can effectively choose and train a medium-sized LLM for your medical practice. Remember that the process is iterative, and continuous refinement will lead to better performance over time. Always prioritize ethical considerations and maintain human oversight in medical applications of AI.

As you embark on this journey, you're not just enhancing your practice, but also contributing to the advancement of AI in healthcare. Your expertise combined with the power of LLMs has the potential to improve patient care and medical decision-making.

Good luck with your LLM training project!

⁂

1. https://www.labellerr.com/blog/how-to-choose-llm-to-suit-for-use-case/ ↩
2. https://community.aws/content/2ZVa61RxToXUFzcuY8Hbut6L150/what-is-an-instruct-model?lang=en ↩