Generative AI for Text Summarization using LLMs

...

April, 2024

Presenter

Gagandeep Dua, Vice President

JPMorgan Chase

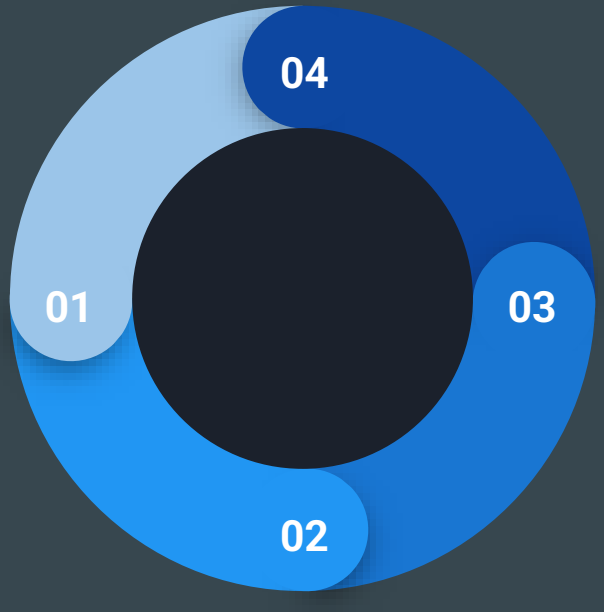
Agenda

1. LLM App Project Lifecycle
2. Text Summarization Use Case and Scope
3. Selecting the Right LLM
4. Training & Adaptation
    a. Summarization using Foundational Models
    b. Fine Tuning
    c. Evaluation using Rouge Metrics
5. Q&A

LLM App Project Lifecycle

Define the Use Case , Scope

Select right LLM



Deployment and Monitoring

Training & Adaptation

- Pretraining
- Prompt Engineering
- Fine Tuning
- Evaluation and Iteration

Define the Use Case and Scope

LLM Use Cases

1. Text / Content Generation
2. Text Summarization
3. Language Translation
4. Text Classification
5. Question Answering
6. Named Entity Recognition
7. Automated Text Summarization

Scope of Text Summarization

1. Research Reports
2. Blogs
3. Financial Statements
4. Dialogues
5. News
6. Large Documents
7. Domain - Finance, Legal, etc.

Selecting the Right LLM

Encoder Only

- Text Classification
- Sentiment Analysis
- Named Entity Recognition

- BERT
- DistilBERT
- RoBERTa

Decoder Branch

- Text Completion
- Text Generation
- Translation
- Q&A

- GPT Family
- Jurassic
- LLaMA

Encoder / Decoder

- Translation
- Text Summarization
- Question & Answering

- T5
- Pegasus
- BART

Training & Adaptation - Hugging Face Overview

Hugging Face Hub

- Models
- Datasets
- Metrics
- Docs

Hugging Face Libraries

- Tokenizers
- Transformers
- Datasets
- Accelerate

Training & Adaptation: Summarization using Foundational Models

Dialogue

```
Sandra: Hey Louis, have you had a chance to work with Hugging Face for NLP tasks?
Louis: Yes, I've been experimenting with their models for text generation. It's been quite impressive.
Sandra: That's great to hear! Which model have you found most useful?
Louis: I've been using GPT-3 for various natural language processing tasks. It's incredibly versatile.
Sandra: Interesting! I've been exploring BERT for sentiment analysis. Have you tried it?
Louis: Not yet, but I've heard good things about BERT's performance for classification tasks. How's your experience been?
Sandra: It's been excellent so far. BERT's pre-trained embeddings are really helpful for understanding text semantics.
Louis: Sounds promising. We should collaborate on some projects using these models!
Sandra: Absolutely! Let's discuss some ideas and get started soon.
Louis: Sounds like a plan. Looking forward to it!
```

Reference Summary

Sandra asks Louis about Hugging Face for NLP.
Louis praises their text generation. Sandra talks BERT for sentiment analysis.
Louis suggests collaboration. They plan to start soon.

Loading Pegasus Model

```
pipe = pipeline("summarization", model="google/pegasus-cnn_dailymail")
```

```
pipe_out = pipe(dialogue)
print("Summary:")
```

Generated Summary using Pegasus

```
Summary:
I've been experimenting with Hugging Face for NLP tasks.
I've been using GPT-3 for various natural language processing tasks.
I've heard good things about BERT's performance for classification tasks.
```

Generated Summary using T5-Large

```
Generated Summary:
Louis: i've been using GPT-3 for various natural language processing tasks.
he says pre-trained embeddings are really helpful for understanding text semantics.
"let's collaborate on some projects using these models!"
```

Training & Adaptation: Fine Tuning Pegasus with Samsum Dataset

3

Set Configuration Parameters

```
from transformers import TrainingArguments, Trainer
training_args = TrainingArguments(
    output_dir='pegasus-samsum', num_train_epochs=1, warmup_steps=500,
    per_device_train_batch_size=1, per_device_eval_batch_size=1,
    weight_decay=0.01, logging_steps=10, push_to_hub=False,
    evaluation_strategy='steps', eval_steps=500, save_steps=1e6,
    gradient_accumulation_steps=16)
```

Setting Trainer

```
trainer = Trainer(model=model, args=training_args,
                  tokenizer=tokenizer, data_collator=seq2seq_data_collator,
                  train_dataset=dataset_samsum_pt["train"],
                  eval_dataset=dataset_samsum_pt["validation"])
```

Generated Summary

```
Generated Summary:
Louis has been experimenting with Hugging Face for NLP tasks.
He has been using GPT-3 for various natural language processing tasks.
Sandra has been exploring BERT for sentiment analysis.
```

Reference Summary

Sandra asks Louis about Hugging Face for NLP.
Louis praises their text generation. Sandra talks BERT for sentiment analysis.
Louis suggests collaboration. They plan to start soon.

T & A : Rouge Metrics for Foundational Vs. Fine Tuned Pegasus

Rouge Metrics with Foundational Pegasus

```
[87] rouge_metric.add(prediction=pipe_out, reference=reference)
     score = rouge_metric.compute()
     rouge_dict = dict((rn, score[rn].mid.fmeasure) for rn in rouge_names)
     records = []
     records.append(rouge_dict)
```

```
model_name = ["google/pegasus"]
pd.DataFrame.from_records(records, index=model_name)
```

	rouge1	rouge2	rougeL	rougeLsum
google/pegasus	0.246154	0.095238	0.184615	0.184615

Rouge Metrics with Fine Tuned Pegasus

```
[62] rouge_metric.add(prediction=summary, reference=reference)
     score = rouge_metric.compute()
     rouge_dict = dict((rn, score[rn].mid.fmeasure) for rn in rouge_names)
     records = []
     records.append(rouge_dict)
```

```
model_name = ["pegasus-trained"]
pd.DataFrame.from_records(records, index=model_name)
```

	rouge1	rouge2	rougeL	rougeLsum
pegasus-trained	0.372881	0.210526	0.338983	0.338983

Text Summarization Use Cases - Which you'll experiment?

1. Research Reports
2. Blogs
3. Financial Statements
4. Dialogues
5. News
6. Large Documents
7. Domain - Finance, Legal, etc.

Q&A

Thank You !!

APPENDIX

Encoder-Decoder Transformer Architecture

1. Input text is tokenized and converted to token embeddings
2. Token embeddings combined with Positional embeddings
3. Input tokens + Positional encodings are passed to Multi-headed self-attention layer.
4. Multi-head Attention Layer encodes each word ‘s relationship with every other word in the same sentence , paying more attention to the most relevant ones.
5. Encoder’s output is fed to each decoder layer

Source: Vaswani et al., 2017