Generative AI Project—Part 2

Muthu Arumugam

To understand how AI projects work, see Part 1.

This part covers a sample project using Python. The source code is published in GitHub and you can download or clone or contribute more examples if you are interested. Link: Source Code

Let’s start with an example where we see 2 people talking and we wanted the model to summarize by going through a few dialogues.

- Use case: Summarize
- Model: Google FLAN-T5 (Model Card)
- Dataset: A Real-Life Scenario Dialogue Summarization Dataset
- Fine-tuning: Not needed

We are getting into writing a Python code to experiment with how a model responds to summarizing some of it from sample data. Several steps are involved.

Step 1: Install pre-requisites

- Mac or Windows
- Python 3.x
- Download source code — https://github.com/muthuka/llm-summarize-demo
- Python libs — torch, torchdata, transformers, datasets

Step 2: Load data set

```
from datasets import load_dataset
from transformers import AutoModelForSeq2SeqLM
from transformers import AutoTokenizer
from transformers import GenerationConfig

huggingface_dataset_name = "knkarthick/dialogsum"
dataset = load_dataset(huggingface_dataset_name)

example_indices = [50, 500]
dash_line = '-'.join('' for x in range(100))

for i, index in enumerate(example_indices):
    print(dash_line)
    print('Example ', i + 1)
    print(dash_line)
    print('INPUT DIALOGUE:')
    print(dataset['test'][index]['dialogue'])
    print(dash_line)
    print('BASELINE HUMAN SUMMARY:')
    print(dataset['test'][index]['summary'])
    print(dash_line)
    print()
```

Step 3: Load the mode and tokenize. Check if encode/decode works.

```
from datasets import load_dataset
from transformers import AutoModelForSeq2SeqLM
from transformers import AutoTokenizer
from transformers import GenerationConfig

huggingface_dataset_name = "knkarthick/dialogsum"
dataset = load_dataset(huggingface_dataset_name)

example_indices = [50, 500]
dash_line = '-'.join('' for x in range(100))

for i, index in enumerate(example_indices):
    print(dash_line)
    print('Example ', i + 1)
    print(dash_line)
    print('INPUT DIALOGUE:')
    print(dataset['test'][index]['dialogue'])
    print(dash_line)
    print('BASELINE HUMAN SUMMARY:')
    print(dataset['test'][index]['summary'])
    print(dash_line)
    print()

model_name = 'google/flan-t5-base'
model = AutoModelForSeq2SeqLM.from_pretrained(model_name)
tokenizer = AutoTokenizer.from_pretrained(model_name, use_fast=Tr
sentence = "What time is it, Tom?"
sentence_encoded = tokenizer(sentence, return_tensors='pt')
sentence_decoded = tokenizer.decode(
    sentence_encoded["input_ids"][0],
    skip_special_tokens=True
)

print('ENCODED SENTENCE:')
print(sentence_encoded["input_ids"][0])
print('\nDECODED SENTENCE:')
print(sentence_decoded)
```

Step 4a: Try zero-shot learning by making your prompt say:

```
Summarize the following conversation.
{dialog}

Summary:
```

Since this is a zero-shot, we wanted the model to predict the summary for us.

```
from datasets import load_dataset
from transformers import AutoModelForSeq2SeqLM
from transformers import AutoTokenizer
from transformers import GenerationConfig

huggingface_dataset_name = "knkarthick/dialogsum"
dataset = load_dataset(huggingface_dataset_name)

example_indices = [50, 500]
dash_line = '-'.join('' for x in range(100))

model_name = 'google/flan-t5-base'
model = AutoModelForSeq2SeqLM.from_pretrained(model_name)
tokenizer = AutoTokenizer.from_pretrained(model_name, use_fast=Tr

for i, index in enumerate(example_indices):
    dialogue = dataset['test'][index]['dialogue']
    summary = dataset['test'][index]['summary']

    prompt = f"""
Summarize the following conversation.

{dialogue}

Summary:
    """

    # Input constructed prompt instead of the dialogue.
    inputs = tokenizer(prompt, return_tensors='pt')
    output = tokenizer.decode(
        model.generate(
            inputs["input_ids"],
            max_new_tokens=50,
        )[0],
        skip_special_tokens=True
    )
    print(dash_line)
    print('Example ', i + 1)
    print(dash_line)
    print(f'INPUT PROMPT:\n{prompt}')
    print(dash_line)
    print(f'BASELINE HUMAN SUMMARY:\n{summary}')
    print(dash_line)
    print(f'MODEL GENERATION - ZERO SHOT:\n{output}\n')
```

You should see the following output

Step 4b: Try zero-shot learning by making your prompt say:

```
Dialogue:
{dialogue}
```

```
What was going on?
```

Let’s see what the results are:

```
from datasets import load_dataset
from transformers import AutoModelForSeq2SeqLM
from transformers import AutoTokenizer
from transformers import GenerationConfig

huggingface_dataset_name = "knkarthick/dialogsum"
dataset = load_dataset(huggingface_dataset_name)

example_indices = [50, 500]
dash_line = '-'.join('' for x in range(100))

model_name = 'google/flan-t5-base'
model = AutoModelForSeq2SeqLM.from_pretrained(model_name)
tokenizer = AutoTokenizer.from_pretrained(model_name, use_fast=Tr

for i, index in enumerate(example_indices):
    dialogue = dataset['test'][index]['dialogue']
    summary = dataset['test'][index]['summary']

    prompt = f"""
Dialogue:

{dialogue}

What was going on?
"""

    inputs = tokenizer(prompt, return_tensors='pt')
    output = tokenizer.decode(
        model.generate(
            inputs["input_ids"],
            max_new_tokens=50,
        )[0],
        skip_special_tokens=True
    )

    print(dash_line)
    print('Example ', i + 1)
    print(dash_line)
    print(f'INPUT PROMPT:\n{prompt}')
    print(dash_line)
    print(f'BASELINE HUMAN SUMMARY:\n{summary}\n')
    print(dash_line)
    print(f'MODEL GENERATION - ZERO SHOT:\n{output}\n')
```

The model got a little better.

Step 4c: Let’s try one shot with a similar dialog question.

```
from datasets import load_dataset
from transformers import AutoModelForSeq2SeqLM
from transformers import AutoTokenizer
from transformers import GenerationConfig

huggingface_dataset_name = "knkarthick/dialogsum"
dataset = load_dataset(huggingface_dataset_name)

example_indices = [50, 500]
dash_line = '-'.join('' for x in range(100))

model_name = 'google/flan-t5-base'
model = AutoModelForSeq2SeqLM.from_pretrained(model_name)
tokenizer = AutoTokenizer.from_pretrained(model_name, use_fast=Tr


def make_prompt(example_indices_full, example_index_to_summarize)
    prompt = ''
    for index in example_indices_full:
        dialogue = dataset['test'][index]['dialogue']
        summary = dataset['test'][index]['summary']

        # The stop sequence '{summary}\n\n\n' is important for FL
        prompt += f"""
Dialogue:

{dialogue}

What was going on?
{summary}


"""

    dialogue = dataset['test'][example_index_to_summarize]['dialo
    prompt += f"""
Dialogue:

{dialogue}

What was going on?
"""

    return prompt


example_indices_full = [50]
example_index_to_summarize = 500
one_shot_prompt = make_prompt(example_indices_full, example_index
print(one_shot_prompt)

summary = dataset['test'][example_index_to_summarize]['summary']

inputs = tokenizer(one_shot_prompt, return_tensors='pt')
output = tokenizer.decode(
    model.generate(
        inputs["input_ids"],
        max_new_tokens=50,
    )[0],
    skip_special_tokens=True
)

print(dash_line)
print(f'BASELINE HUMAN SUMMARY:\n{summary}\n')
print(dash_line)
print(f'MODEL GENERATION - ONE SHOT:\n{output}')
```

Litle better than zero-shot

Step 4d: We can try to pass few shots and see the output

```
from datasets import load_dataset
from transformers import AutoModelForSeq2SeqLM
from transformers import AutoTokenizer
from transformers import GenerationConfig

huggingface_dataset_name = "knkarthick/dialogsum"
dataset = load_dataset(huggingface_dataset_name)

example_indices = [50, 500]
dash_line = '-'.join('' for x in range(100))

model_name = 'google/flan-t5-base'
model = AutoModelForSeq2SeqLM.from_pretrained(model_name)
tokenizer = AutoTokenizer.from_pretrained(model_name, use_fast=Tr


def make_prompt(example_indices_full, example_index_to_summarize)
    prompt = ''
    for index in example_indices_full:
        dialogue = dataset['test'][index]['dialogue']
        summary = dataset['test'][index]['summary']

        # The stop sequence '{summary}\n\n\n' is important for FL
        prompt += f"""
Dialogue:

{dialogue}

What was going on?
{summary}


"""

    dialogue = dataset['test'][example_index_to_summarize]['dialo

    prompt += f"""
Dialogue:

{dialogue}

What was going on?
"""

    return prompt


# Let's start few start config
example_indices_full = [50, 100]
example_index_to_summarize = 500
few_shot_prompt = make_prompt(example_indices_full, example_index
print(few_shot_prompt)


summary = dataset['test'][example_index_to_summarize]['summary']
inputs = tokenizer(few_shot_prompt, return_tensors='pt')
output = tokenizer.decode(
    model.generate(
        inputs["input_ids"],
        max_new_tokens=50,
    )[0],
    skip_special_tokens=True
)
print(dash_line)
print(f'BASELINE HUMAN SUMMARY:\n{summary}\n')
print(dash_line)
print(f'MODEL GENERATION - FEW SHOT:\n{output}')
```

Pretty much we got the same as a one-shot.

Step 4e: We can try to adjust a few parameters and see what happens. We have adjusted the temperature to 1.0 and also told the model to give up to ONLY 6 tokens.

```
from datasets import load_dataset
from transformers import AutoModelForSeq2SeqLM
from transformers import AutoTokenizer
from transformers import GenerationConfig

huggingface_dataset_name = "knkarthick/dialogsum"
dataset = load_dataset(huggingface_dataset_name)

example_indices = [50, 500]
dash_line = '-'.join('' for x in range(100))

model_name = 'google/flan-t5-base'
model = AutoModelForSeq2SeqLM.from_pretrained(model_name)
tokenizer = AutoTokenizer.from_pretrained(model_name, use_fast=Tr


def make_prompt(example_indices_full, example_index_to_summarize)
    prompt = ''
    for index in example_indices_full:
        dialogue = dataset['test'][index]['dialogue']
        summary = dataset['test'][index]['summary']

        # The stop sequence '{summary}\n\n\n' is important for FL
        prompt += f"""
Dialogue:

{dialogue}

What was going on?
{summary}


"""

    dialogue = dataset['test'][example_index_to_summarize]['dialo

    prompt += f"""
Dialogue:

{dialogue}
```

```
What was going on?
"""

    return prompt


# Let's start few start config
example_indices_full = [50, 100]
example_index_to_summarize = 500
few_shot_prompt = make_prompt(example_indices_full, example_index
print(few_shot_prompt)

summary = dataset['test'][example_index_to_summarize]['summary']
# generation_config = GenerationConfig(max_new_tokens=50)
# generation_config = GenerationConfig(max_new_tokens=10)
# generation_config = GenerationConfig(max_new_tokens=50, do_samp
# generation_config = GenerationConfig(max_new_tokens=50, do_samp
# generation_config = GenerationConfig(max_new_tokens=100, do_sam
generation_config = GenerationConfig(
    max_new_tokens=6, do_sample=False, temperature=1.0)


inputs = tokenizer(few_shot_prompt, return_tensors='pt')
output = tokenizer.decode(
    model.generate(
        inputs["input_ids"],
        generation_config=generation_config,
    )[0],
    skip_special_tokens=True
)

print(dash_line)
print(f'MODEL GENERATION - FEW SHOT:\n{output}')
print(dash_line)
print(f'BASELINE HUMAN SUMMARY:\n{summary}\n')
```

The model couldn’t come up with 6-word summary for the same sample. The sentence prematurely ended. See below:

Summary: We see examples of finding a problem, model, dataset, and app for summarization. The model output didn’t get better with a few shots but it got better when we wanted the answer to be creative instead of realistic. There are many ways you can determine what suits your needs.

Hope this example is simple and useful.

Disclaimer: This is not generated by an AI bot. Also, a lot of these were learned through the DeepLearning.ai course at Coursera.