run

June 21, 2024

1 Case Study: Benchmark Analysis of Takeoff for Consumer Complaints

1.1 Dataset Preparation

[1]:
```python
import pandas as pd
from transformers import AutoTokenizer

# read the data
df = pd.read_csv("data/consumer_complaints.csv")
```

1.1.1 Input/Output Token analysis

[2]:
```python
tokenizer = AutoTokenizer.from_pretrained("mistralai/Mistral-7B-Instruct-v0.1")
df['num_tokens'] = df['consumer_complaint'].apply(lambda x:␣
 ↪len(tokenizer(x)['input_ids']))
```

1.1.2 Prepare prompt input

[6]:
```python
prompt = "You are a customer care specialist. Please read the customer␣
 ↪complaint and categorize it. The categories you can chose from are - debt␣
 ↪collection, mortgage, credit reporting, credit card, bank account or␣
 ↪service, customer loan, and Other. Please only return the category␣
 ↪classification.\nCustomer Complaint: "

# mistral chat template
def chat_template(sys_prompt, user_input):
    return "<s>[INST] " + sys_prompt + user_input +  " [/INST]"

df['prompt'] = df['consumer_complaint'].apply(lambda x: chat_template(prompt,␣
 ↪x))

print("===== Example prompt =====\n", df['prompt'][0])
```

```
===== Example prompt =====
 <s>[INST] You are a customer care specialist. Please read the customer
complaint and categorize it. The categories you can chose from are - debt
```

```
collection, mortgage, credit reporting, credit card, bank account or service,
customer loan, and Other. Please only return the category classification.
Customer Complaint: XXXX has claimed I owe them {$27.00} for XXXX years despite
the PROOF of PAYMENT I sent them : canceled check and their ownPAID INVOICE for
{$27.00}!
They continue to insist I owe them and collection agencies are after me.
How can I stop this harassment for a bill I already paid four years ago? [/INST]
```

[7]:
```
prompt_batch_10 = df['prompt'].head(10).to_list()
prompt_batch_100 = df['prompt'].head(100).to_list()
prompt_batch_1000 = df['prompt'].head(1000).to_list()
```

1.2 Benchmark

Helper function

[25]:
```
## helper functions
import time
from takeoff_client import TakeoffClient

def log_time_taken(start, end, num_prompts):
    """
    Logs the time taken between start and end times in both seconds and a␣
 ↪readable format.

    Args:
    start (float): The start time in seconds.
    end (float): The end time in seconds.
    """
    # Calculate the time difference
    time_taken_seconds = end - start

    # Convert seconds to a more readable format
    hours = int(time_taken_seconds // 3600)
    minutes = int((time_taken_seconds % 3600) // 60)
    seconds = int(time_taken_seconds % 60)

    # Log the time taken
    print(f"Time taken for {num_prompts} prompts: {time_taken_seconds:.2f}␣
 ↪seconds => ({hours}h{minutes}m{seconds}s)")


def run_bench(num_rounds=2):
    client = TakeoffClient(base_url="http://localhost")
    readers = client.get_readers()
    model_name = readers['primary'][0]['model_name']

    print("====== Model Info ======")
    print("Model Name: ", model_name)

    for i in range(num_rounds):
        print("======= Benchmark Round ", i+1, " ========")

        start = time.time()
        response_10 = client.generate(prompt_batch_10)
        end = time.time()
        log_time_taken(start, end, 10)

        start = time.time()
        response_100 = client.generate(prompt_batch_100)
        end = time.time()
        log_time_taken(start, end, 100)

        start = time.time()
        response_1000 = client.generate(prompt_batch_1000)
        end = time.time()
        log_time_taken(start, end, 1000)

    return response_10, response_100, response_1000
```

1.2.1 Setting 1 Mistral Full

- Model: mistralai/Mistral-7B-Instruct-v0.1
- Server: takeoff v0.14.4
- GPU: Nvidia A10G

[26]:
```
response_10, response_100, response_1000 = run_bench()
```

```
====== Model Info ======
Model Name:  mistralai/Mistral-7B-Instruct-v0.1
======= Benchmark Round  1  ========
Time taken for 10 prompts: 5.05 seconds => (0h0m5s)
Time taken for 100 prompts: 28.18 seconds => (0h0m28s)
Time taken for 1000 prompts: 257.29 seconds => (0h4m17s)
======= Benchmark Round  2  ========
Time taken for 10 prompts: 6.69 seconds => (0h0m6s)
Time taken for 100 prompts: 31.00 seconds => (0h0m31s)
Time taken for 1000 prompts: 253.57 seconds => (0h4m13s)
```

1.2.2 Setting 2 Mistral 4bit AWQ

- Model: TheBloke/Mistral-7B-Instruct-v0.1-AWQ
- Server: takeoff v0.14.4
- GPU: Nvidia A10G

[27]:
```
response_10, response_100, response_1000 = run_bench()
```

```
====== Model Info ======
Model Name:  TheBloke/Mistral-7B-Instruct-v0.1-AWQ
======= Benchmark Round  1  ========
Time taken for 10 prompts: 4.81 seconds => (0h0m4s)
Time taken for 100 prompts: 42.12 seconds => (0h0m42s)
Time taken for 1000 prompts: 440.36 seconds => (0h7m20s)
======= Benchmark Round  2  ========
Time taken for 10 prompts: 10.83 seconds => (0h0m10s)
Time taken for 100 prompts: 40.28 seconds => (0h0m40s)
Time taken for 1000 prompts: 423.90 seconds => (0h7m3s)
```

1.2.3 Setting 3 Mixtral Full

- Model: mistralai/Mixtral-8x7B-Instruct-v0.1
- Server: takeoff v0.14.3
- GPU: Nvidia A10G * 8

[35]:
```
response_10, response_100, response_1000 = run_bench()
```

```
====== Model Info ======
Model Name:  mistralai/Mixtral-8x7B-Instruct-v0.1
======= Benchmark Round  1  ========
Time taken for 10 prompts: 19.50 seconds => (0h0m19s)
Time taken for 100 prompts: 126.26 seconds => (0h2m6s)
Time taken for 1000 prompts: 1366.86 seconds => (0h22m46s)
======= Benchmark Round  2  ========
Time taken for 10 prompts: 24.26 seconds => (0h0m24s)
Time taken for 100 prompts: 133.32 seconds => (0h2m13s)
Time taken for 1000 prompts: 1399.05 seconds => (0h23m19s)
```

1.2.4 Setting 4 Mixtral 4bit AWQ

- Model: TheBloke/dolphin-2.7-mixtral-8x7b-AWQ
- Server: takeoff v0.14.4
- GPU: Nvidia A10G * 2

[37]:
```
response_10, response_100, response_1000 = run_bench()
```

```
====== Model Info ======
Model Name:  TheBloke/dolphin-2.7-mixtral-8x7b-AWQ
======= Benchmark Round  1  ========
Time taken for 10 prompts: 16.10 seconds => (0h0m16s)
Time taken for 100 prompts: 180.82 seconds => (0h3m0s)
Time taken for 1000 prompts: 1846.86 seconds => (0h30m46s)
======= Benchmark Round  2  ========
Time taken for 10 prompts: 22.87 seconds => (0h0m22s)
Time taken for 100 prompts: 185.18 seconds => (0h3m5s)
Time taken for 1000 prompts: 1854.00 seconds => (0h30m54s)
```