In [1]:
```
from transformers import AutoModelForCausalLM, AutoTokenizer
import torch

import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
from transformers import AutoModelForCausalLM, AutoTokenizer
import torch

import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
def load_tokenizer_and_model(model="microsoft/phi-2"):
    """
    Load tokenizer and model instance for some specific Phi-2 model.
    """
    # Initialize tokenizer and model
    print("Loading model...")
    tokenizer = AutoTokenizer.from_pretrained(model, padding_side='left')
    model = AutoModelForCausalLM.from_pretrained(model)

    # Return tokenizer and model
    return tokenizer, model
```

In [4]:
```
from intel_extension_for_pytorch.quantization import prepare, convert
import intel_extension_for_pytorch as ipex

def quantize_model(tokenizer, model):
    """
    Adding IPEX dynamic quantization to the model
    """
    # Evaluate model
    model.eval()

    print("Quantization in progress...")

    # Prepare example outputs for the model
    question, text = "What is SYCL?", "SYCL is an industry-driven standard,
    inputs = tokenizer(question, text, return_tensors="pt")
    jit_inputs  = tuple((inputs['input_ids']))

    # Create configuration for dynamic quantization
    qconfig = ipex.quantization.default_dynamic_qconfig

    # Optimize model
    model = ipex.optimize(model)

    # Prepare model for quantization using previously prepared parameters
    prepared_model = prepare(model, qconfig, example_inputs=jit_inputs, inpl

    # Convert types in model
    converted_model = convert(prepared_model)
```

```
    return tokenizer, converted_model
```

In [5]:
```
from intel_extension_for_pytorch.quantization import prepare, convert
import intel_extension_for_pytorch as ipex

def quantize_model(tokenizer, model):
    """
    Adding IPEX dynamic quantization to the model
    """
    # Evaluate model
    model.eval()

    print("Quantization in progress...")

    # Prepare example outputs for the model
    question, text = "What is SYCL?", "SYCL is an industry-driven standard,
    inputs = tokenizer(question, text, return_tensors="pt")
    jit_inputs  = tuple((inputs['input_ids']))

    # Create configuration for dynamic quantization
    qconfig = ipex.quantization.default_dynamic_qconfig

    # Optimize model
    model = ipex.optimize(model)

    # Prepare model for quantization using previously prepared parameters
    prepared_model = prepare(model, qconfig, example_inputs=jit_inputs, inpl

    # Convert types in model
    converted_model = convert(prepared_model)

    return tokenizer, converted_model
```

In [6]:
```
def generate_response(tokenizer, model, chat_round, chat_history_ids):
    """
    Generate a response to some user input.
    """
    # Encode user input and End-of-String (EOS) token
    new_input_ids = tokenizer.encode(input(">> You:") + tokenizer.eos_token,

    # Append tokens to chat history
    bot_input_ids = torch.cat([chat_history_ids, new_input_ids], dim=-1) if

    # Generate response given maximum chat length history of 2000 tokens
    chat_history_ids = model.generate(
        bot_input_ids,
        do_sample=True,
        max_length=2000,
        top_k=50,
        top_p=0.95,
        pad_token_id=tokenizer.eos_token_id
    )

    # Print response
```

```
    print("Phi-2: {}".format(tokenizer.decode(chat_history_ids[:, bot_input_

    # Return the chat history ids
    return chat_history_ids
```

In [7]:
```
def chat_for_n_rounds(tokenizer, model, n=5):
    """
    Chat with chatbot for n rounds (n = 5 by default)
    """

    # Initialize history variable
    chat_history_ids = None

    # Chat for n rounds
    for chat_round in range(n):
        chat_history_ids = generate_response(tokenizer, model, chat_round, c
```

In [8]:
```
# Initialize tokenizer and model
tokenizer, model = load_tokenizer_and_model()

# Adding ipex quantization to the model
tokenizer, model = quantize_model(tokenizer, model)
torch.save(model.state_dict(), './quantized_model')
```

```
Special tokens have been added in the vocabulary, make sure the associated w
ord embeddings are fine-tuned or trained.
Loading model...
Loading checkpoint shards:   0%|          | 0/2 [00:00<?, ?it/s]
Quantization in progress...
```

In [9]:
```
chat_for_n_rounds(tokenizer, model, 2)
```

```
A decoder-only architecture is being used, but right-padding was detected! F
or correct generation results, please set `padding_side='left'` when initial
izing the tokenizer.
```

Phi-2: The first step toward becoming a better teacher is knowing what you d
on't know. You can't expect students to learn material if you don't know how
to present it. Asking questions and being comfortable admitting that you do
n't know something is important. It will keep you from being afraid to make
mistakes in front of students and it will prevent you from being too confide
nt and presenting material you're not familiar with.
Be able to explain your lesson so others can understand it. This is called e
xplaining what you're doing. "Explaining what you're doing" is a teaching te
chnique that helps you explain what you're thinking and doing when you teac
h, rather than telling. An example of telling is "I'm just going to read the
first chapter to the class and then ask some questions about it." An example
of explaining is "We're going to read the first chapter to see if there are
any big ideas that we could discuss and write down. Then, we'll ask if anyon
e has questions about anything we read." You're telling your students what y
ou're going to do. Explaining what you're doing is showing how your teaching
will work.
Being organized and planning ahead can help ensure that you aren't caught of
f guard in the classroom. When you plan ahead, you know what you're teachin
g. You know how you want students to learn and what you think they will get
out of it. You know if you need to make any changes to your lesson before yo
u teach it. You plan out what you're going to use in your classroom and you
prepare for any unforeseen circumstances. When you're prepared for any probl
ems that may occur, you won't have to worry about what to do if something go
es wrong.
Being flexible and having a sense of humor helps you to overcome challenges
that can arise in the classroom. It also makes you approachable and allows s
tudents to ask questions about what you're teaching without feeling embarras
sed.
Being able to keep students engaged in your lesson by making it interesting
and exciting will keep them interested and excited to learn. You can do this
by using hands-on activities and demonstrations, telling stories, having the
m participate in class discussions, using humor, making it interactive, and
using multimedia such as videos, slideshows, and audio/visual equipment.
When you present material in a clear and understandable manner, you reduce t
he risk of miscommunication. Miscommunication is a problem in classrooms bec
ause students have different learning styles and can interpret information i
n various ways. Being able to present information clearly will help avoid th
is problem.
It can be difficult to manage a large classroom with many different personal
ities, but it's essential if you want to be a successful teacher. Managing a
classroom is important because if you can't control your students, you won't
be able to teach them effectively.
Keeping students safe in the classroom is important. You want to ensure that
students don't get injured while they're in your class. This means following
safety procedures and knowing what to do if there's an emergency.
Being organized and having a good relationship with your students are import
ant aspects of being a successful teacher. You should have an organized clas
sroom because it helps you stay on top of your work and it helps your studen
ts know where things are located. You should also have a good relationship w
ith your students because it makes them feel comfortable and allows you to b
etter connect with them.
It's hard to be a good teacher if you're not passionate about what you're te
aching. Being passionate about what you're teaching makes you want to learn
more and be able to share that knowledge with your students.
One way to determine if you're a good teacher is to find out if your student
s are improving academically. You can measure this by having your students t

```
ake standardized tests at the end of the year. If they do better on the test
s than they did the year before, then you know you're doing a good job.
If you can find ways to make learning fun for your students, then you know y
ou're a good teacher. You should be able to make learning fun because that's
how students learn best.
In conclusion, there are many qualities that make someone a good teacher. Th
ese qualities include: being a good communicator, being organized, being pas
sionate about teaching, being able to control a classroom, being able to exp
lain what you're doing, being flexible, being able to keep students engaged,
managing a classroom, and keeping students safe. If you possess these qualit
ies, then you will be a great teacher.
```

```
A decoder-only architecture is being used, but right-padding was detected! F
or correct generation results, please set `padding_side='left'` when initial
izing the tokenizer.
```

Phi-2: By: Mark E. Ferenciak & Brian L. Ferenciak
It is often difficult to understand an insurance policy. Even after reading over the coverage, the policy may be ambiguous as to whether it applies. Unfortunately, this is the reality of insurance. Insurance is a contract between two parties: the insured and the insurer.
There are many terms that are not defined. In the insurance world, this is ISSUE OF CONTRACT. This means that the policy language or the terms of the policy is the controlling force. The parties can agree to add or exclude certain terms, but the parties will never agree that the contract is the issue of contract. A contract by definition is a written agreement. If there is no written agreement, then the parties must turn to the state's governing law to determine what the insurance contract means. Most state insurance codes have "default rules" and provisions to interpret insurance contracts. These rules will help determine who is the insured and who the insurer is.
"Insured" is defined in a standard automobile insurance policy as an owner or operator of an automobile who has paid the premium. The automobile must also be maintained in an acceptable condition. The definition also states that the insured person was or should have been in physical possession of the automobile at the time of an accident. This means that the policyholder who owns and operates an automobile must be physically at or close to his car to file an insurance claim. In cases where there are multiple drivers on the vehicle at the time of the accident, the person who was at the wheel will have a duty to protect the others' insured status. If the policyholder was not the one driving the vehicle at the time of the accident, there is no coverage.
"Insurer" is defined as the insurance company, or a broker or agency of the insurance company, for the policy. In some policies, it does not matter if the company is listed. If an insurance company has a policy number, then it may be the "insurer".
A policy will state the named insured. This is the only person on the policy that the insurance company is responsible for. The policy will also have a list of people who are included as additional insureds. This is an exclusion of insured status. Additional insureds can be added to a policy for specific events. If a named insured or additional insured are killed, the insurance company has a duty to defend them. This also means that there is an obligation to defend claims against the deceased. If 450 employees are listed as additional insureds on an auto policy and two employees are killed, it is only necessary to pay for the defense of the two employees, and the insurance company is responsible for the cost of the defense. This applies to employers who are listed as additional insureds. In order to receive reimbursement for these costs, the employer must show proof of the defense costs. If an employer is an additional insured and the employer is not named on the policy, the employer will not be protected.
The most important element of an insurance policy is the amount of coverage. The standard definition of coverage will state "insured bodily injury and property damage sustained by the person(s) for whom the insurer is obligated to pay and/or legal liability". What does this mean? First, bodily injury means an injury to a person. There are some policies that list the definitions in the policy itself. Other policies use the definitions from the governing law in the state. The definitions for property damage and legal liability will also be dependent on state law. Some definitions are similar, while other definitions are different. One thing is for certain, there is a defined difference between property damage and personal injury. Property damage includes the damage to the insured person's car and the damage to the property around the car. An example would be if a truck hits a fence that was in the way of the truck. The fence God destroyed would be considered property damage and it is the insured person's liability. On the other hand, injuries to people o

```
ther than the insured person, or the people who own the property that was da
maged, would be considered bodily injury. A good example of this would be a
car accident. The person whose car is damaged is the insured person. The oth
er person is injured and that is considered a bodily injury.
Coverage is the subject of every insurance claim. Before filing a claim, it
is necessary to understand the policy terms. It is also necessary to determi
ne if the policy covers the claim at all. If the insurance policy does not c
over the claim, or if the policy limit does not exceed the value of the clai
m, then there will be no recovery.
Attorney Mark Ferenciak received his undergraduate degree from Michigan Stat
e University in 1993 and is currently enrolled in the graduate insurance law
program at Michigan State University College of Law, from which he is expect
ed to receive his J.D. in May of 2016. Mark is currently a first-year associ
ate in the law firm of Gebhardt & Ferenciak, PLC in East Lansing, Michigan,
and serves as an extern for the law school's Insurance and Risk Management L
aw Clinic. He has been involved with the law school's Insurance Law Associat
ion since the 2012-2013 school year and has served as the Association's Exec
utive
```

In [10]:
```python
from time import time
def test_inference(model, data, warmup=5 , iters=25):
    print("Warmup...")
    for i in range(warmup):
        out = model(data)

    print("Inference...")
    inference_time = 0
    for i in range(iters):
        start_time = time()
        out = model(data)
        end_time = time()
        inference_time = inference_time + (end_time - start_time)

    inference_time = inference_time / iters
    return inference_time
```

In [11]:
```python
print("Inference with FP32")
tokenizer_fp32, model_fp32 = load_tokenizer_and_model()
data = torch.randint(model_fp32.config.vocab_size, size=[1, 512])
fp32_inference_time = test_inference(model_fp32, data = data)
```

```
Inference with FP32
Loading model...
Special tokens have been added in the vocabulary, make sure the associated w
ord embeddings are fine-tuned or trained.
Loading checkpoint shards:   0%|          | 0/2 [00:00<?, ?it/s]
Warmup...
Inference...
```

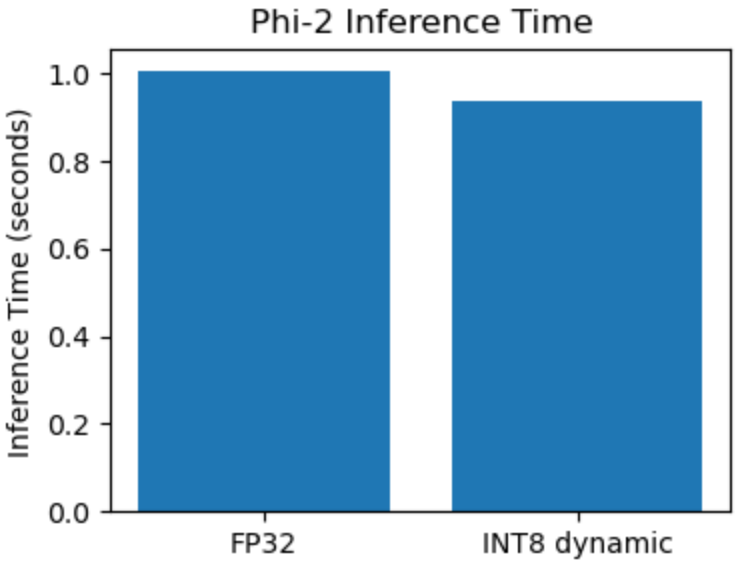
In [12]:
```python
print("Inference with Dynamic INT8")
tokenizer_int8, model_int8 = load_tokenizer_and_model()
tokenizer_int8, model_int8 = quantize_model(tokenizer_int8, model_int8)
data = torch.randint(model_int8.config.vocab_size, size=[1, 512])
int8_inference_time = test_inference(model_int8, data = data)
```

```
Special tokens have been added in the vocabulary, make sure the associated w
ord embeddings are fine-tuned or trained.
Inference with Dynamic INT8
Loading model...
Loading checkpoint shards:   0%|          | 0/2 [00:00<?, ?it/s]
Quantization in progress...
Warmup...
Inference...
```

In [13]:
```
import matplotlib.pyplot as plt

# Create bar chart with training time results
plt.figure(figsize=(4,3))
plt.title("Phi-2 Inference Time")
plt.ylabel("Inference Time (seconds)")
plt.bar(["FP32", "INT8 dynamic"], [fp32_inference_time, int8_inference_time]
```

Out[13]: <BarContainer object of 2 artists>



In [14]:
```
print("[CODE_SAMPLE_COMPLETED_SUCCESFULLY]")
```

```
[CODE_SAMPLE_COMPLETED_SUCCESFULLY]
```