Quantized Large Language Models for Mental Health Applications: A Benchmark Study and Analysis

Aayush Jannumahanti[1]

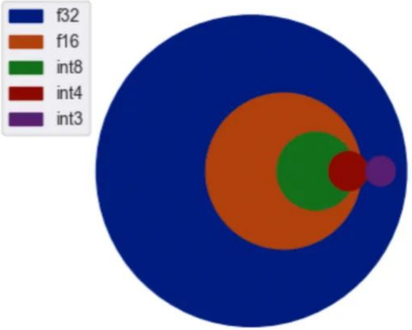
[1]University of Maryland, Baltimore County, MD, USA aayushj1@umbc.edu

CMSC 799

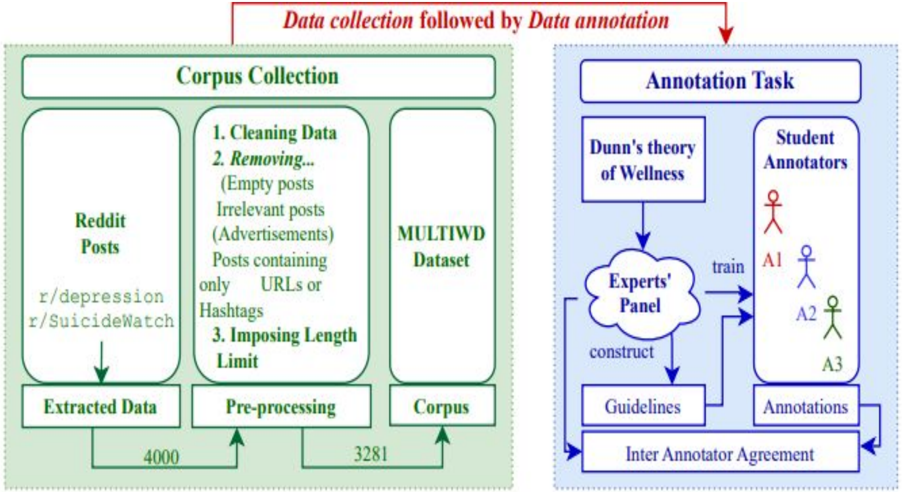
Directed by Dr. Manas Gaur

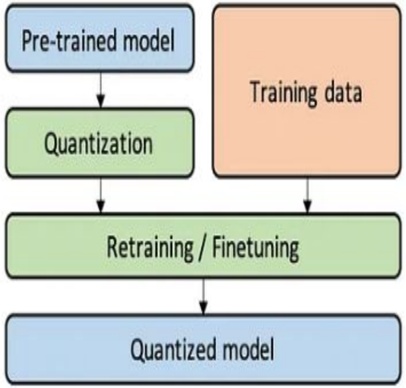
Department of Computer Science and Electrical Engineering

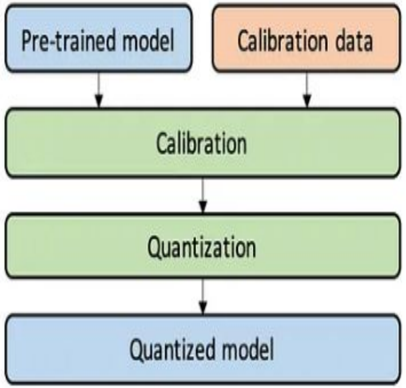
Problem



1. float32 (f32): 32 bits or 4 bytes
2. float16 (f16): 16 bits or 2 bytes
3. int8: 8 bits or 1 byte
4. int4: 4 bits or 0.5 bytes
5. int3: 3 bits or 0.375 bytes







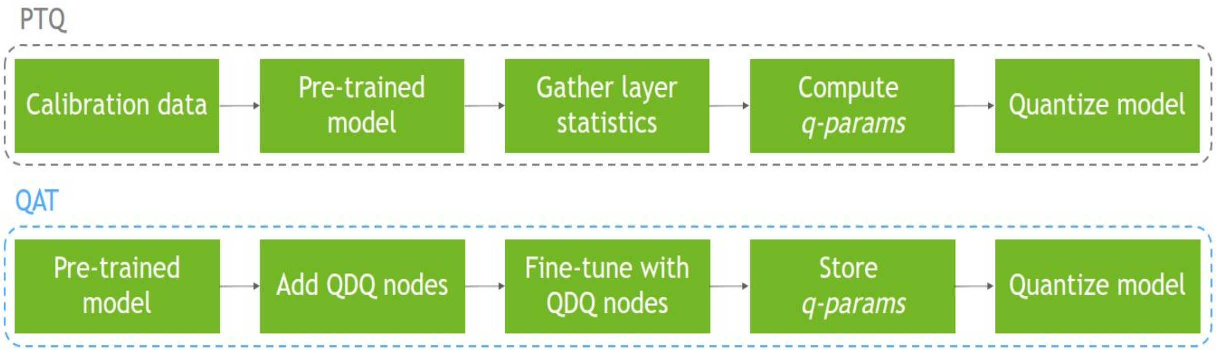
Problem

- Mental Health has remained a significant unaddressed challenge
- Large Language Models have remained to be large and computationally expensive
- NLP has broadly been classified as Prompting, Fine-tuning or RAG these days
- Numerous studies use natural language processing (NLP) approaches to analyze social media for mental health automatically
- Has anyone come up with robust solutions to these sensitive issues?
- Yet…?

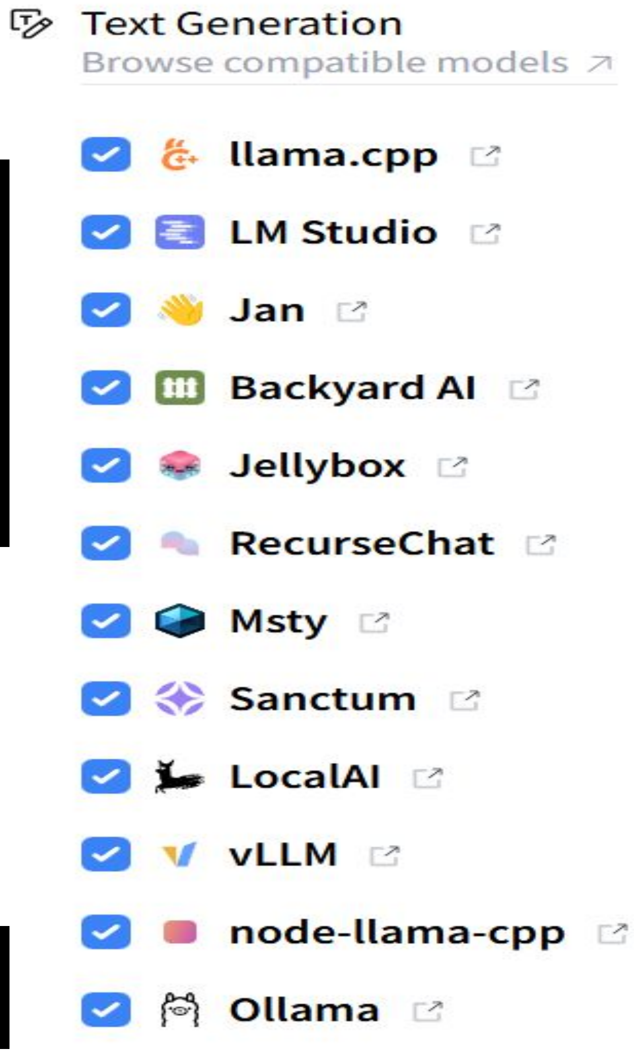
Why Quantized LLMs in Mental Health?

WHY
Using these Qunatized LLM's it's easier to get get access to over several mental health datasets like CAMS, SAD, CLP, DR, Dreddit, SWMH, MultiWD, IMHI Dataset, to get various results like coherence, accuracy, relevance, memory saved, parameters and different levels is being produced

WHAT
Several of these models available on easily accessible resources like Huggingface, Ollama, vLLM etc



HOW
Several strategies like fp16, LLM.int8(), NF4, QWAT, PTQ, QAT



Questions

1. **Quantization:** Is quantization an effect strategy overall, as it is widely being adopted in industry and academia?
2. **LLMs Alignment:** Can LLMs be quantized in a manner that can align with mental health scenarios?
3. **Closed domain tasks:** Will LLMs even be able produce results for closed domain tasks?

What did we do?

1. We’ve taken several quantized models that were then run over Interpretable Mental Health Instruction, This helped achieved in mental health specific prompts
2. Therapeutic response generation from the prompts were later extracted and the performance, accuracy benchmark was created
3. Interpretable features for response analysis tools and explanation generation
4. Reasoning assertion and transparency is available
5. Robust evaluation framework with relevant performance metrics for quality assessments mark comparisons

Why IMHI and Quantized LLM’s

- The IMHI (Interpretable Mental Health Instruction) dataset consists of pairs of user queries and AI responses focused on mental health scenarios. Each example typically includes:

- A user's mental health-related query or situation
- An AI response with:
- Empathetic acknowledgment
- Professional analysis
- Evidence-based suggestions
- Safety considerations when relevant

How does the IMHI Dataset looks like?

Consider this post: "How to avoid a relapse? I've been having a particularly rough year; I attempted suicide, the love of my life left me, I failed my year at university, I've been physically assaulted, I'v…	Yes, the poster suffers from depression. Reasoning: The post displays a significant amount of emotional distress, including experiencing a suicide attempt, multiple traumatic events, and the recent death of a…
Consider this post: "This is how I feel and I have a slight sense I am the only one to think think this way. I am depressed? Help wanted I yearn for the days I was in high school. I felt as if I was always…	Yes, the poster suffers from depression. Reasoning: The post exhibits a clear and profound sense of emotional distress and hopelessness. The poster expresses a longing for the past when they felt happier…
Consider this post: "Advice for getting back in touch with everyday life? I think the hardest thing for me, and the largest factor I need to address to try and feel healthy again, is simply trying to feel…	Yes, the poster suffers from depression. Reasoning: The post conveys a sense of disengagement and difficulty finding enjoyment in everyday life. The poster describes feeling disconnected from the present…
Consider this post: "Having a really bad stretch of days.. Hi all, I know this sub moves kinda fast but I just wanted to vent, hopefully some of you read this.. So my story is that I've had depression and social…	Yes, the poster suffers from depression. Reasoning: The post expresses a deep sense of emotional pain, hopelessness, and despair. The poster details their long-standing struggle with depression and social…
Consider this post: "I dont understand how people can be so interested in so many things First, let me apologize for formatting issues, I am typing this on mobile. I generally find television boring, have…	Yes, the poster suffers from depression. Reasoning: The post expresses a lack of interest and enjoyment in various hobbies and forms of entertainment, which can be indicative of anhedonia, a prominent symptom of…
Consider this post: "Even if you hate them, theres something gut wrenching about hearing parents fight. I hate both of my parents. They are mentally abusive to me, and just a lack of care most of my life. Even…	Yes, the poster suffers from depression. Reasoning: The post conveys a deep sense of emotional pain and discomfort. The poster describes their parents as mentally abusive and lacking care, expressing a strong…
Consider this post: "5 Simple family activities to get and stay healthy - Made In A Pinch Engaging in healthy family activities together not only promotes a healthy lifestyle, it also creates bonding…	No, the poster does not suffer from depression. Reasoning: The post does not display any emotional indicators of depression. It talks about engaging in healthy family activities, promoting a healthy…
Consider this post: "Will I ever be with somebody? I'm so lonely. I'm 19 and I don't have a clue what a relationship feels like. All my friends are talking about marriage and I've never even had a real kiss I…	Yes, the poster suffers from depression. Reasoning: The post exhibits a strong sense of emotional distress, loneliness, and lack of fulfillment in relationships. The poster expresses feelings of not bein…

What is Quantization? Why do we need it?

The problem

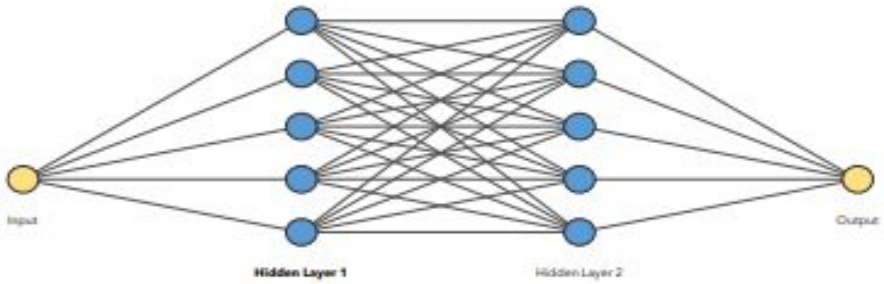
- Most modern deep neural networks are made up of billions of parameters. For example, the smallest LLaMA 2 has 7 billion parameters. If every parameter is 32 bit, then we need $\frac{7\times10^9\times32}{8\times10^9} = 28\text{ GB}$ just to store the parameters on disk.
- When we inference a model, we need to load all its parameters in the memory, this means big models cannot be loaded easily on a standard PC or a smart phone.
- Just like humans, computers are slow at computing floating-point operations compared to integer operations. Try to do 3×6 and compare it to 1.21×2.897, which one can you compute faster?

The solution

- Quantization aims to reduce the total amount of bits required to represent each parameter, usually by converting floating-point numbers into integers. This way, a model that normally occupies 10 GB can be "compressed" to less than 1 GB (depending on the type of quantization used). **Please note:** quantization doesn't mean truncating/rounding. We don't just round up or down all the floating-point numbers! We will see later how it works
- Quantization can also speed up computation, as working with smaller data types is faster (for example multiplying two integers is faster than multiplying two floating point numbers).

How do LLM weights quantize?

Applying quantization



Dequantize → **Output** → $Y = XW + B$ ← **Perform all operations using integer arithmetic**

The main benefit is that integer operations are much faster in most hardware (especially on embedded devices) than floating point operations

Input
Sometimes called "activation"

Weight

Bias
Usually quantized as int32

Quantize

Quantization range: how to choose $[\alpha, \beta]$

Min-Max: To cover the whole range of values, we can set

- $\alpha = \max(V)$
- $\beta = min(V)$
- Sensitive to outliers.

Outlier

Original	43.31	-44.93	0	...	...	...	38.48	-20.49	1000.00	-28.02
Dequantized (Min-Max)	45.08	-45.08	0	24.59	-45.08	-12.29	36.88	-20.49	999.85	-28.68
Dequantized (Percentile)	43.38	-44.52	0	...	...	...	38.48	-20.49	50.00	-28.01

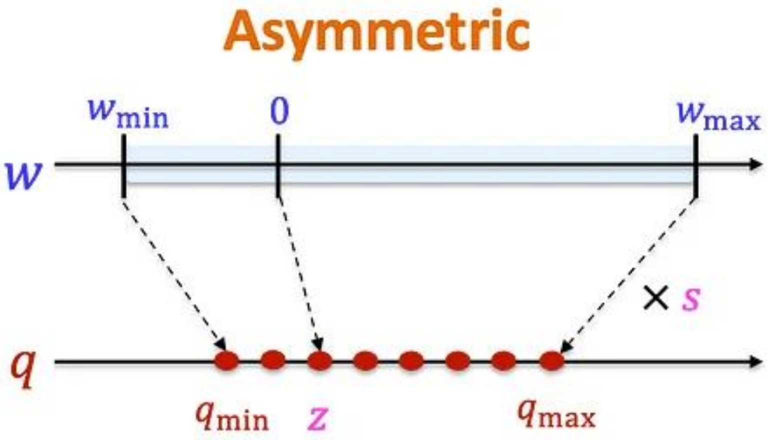
Percentile: Set the range to the percentile of the distribution of V, to reduce sensitivity to outliers

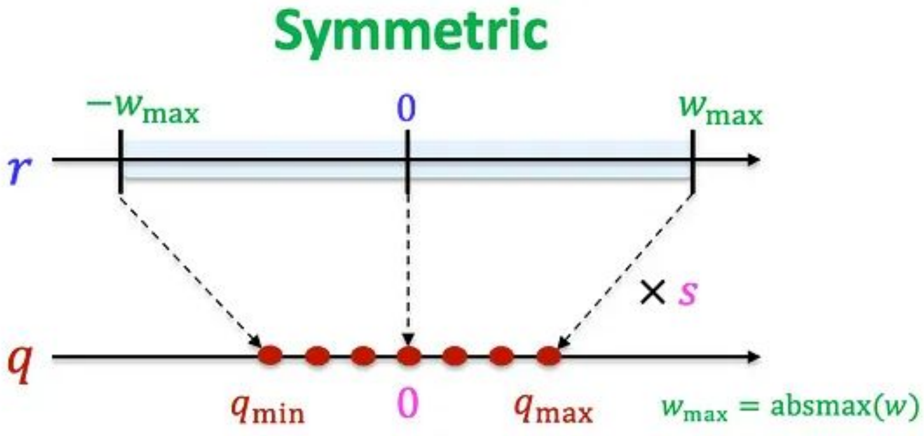
Only the outlier is quantized with a large error

Quantization range: how to choose $[\alpha, \beta]$

If the vector V represents the tensor to be quantized, we can choose the $[\alpha, \beta]$ range according to the following strategies:

- **Mean-Squared-Error**: choose $[\alpha, \beta]$ such that the MSE error between the original values and the quantized values is minimized.
  - It is usually solved using Grid-Search
- **Cross-Entropy**: used when the values in the tensor being quantized are not equally important. This happens for example in the Softmax layer in Large Language Models. Since most of the inference strategies are Greedy, Top-P or Beam search, it is important to preserve the order of the largest values after quantization.
  - $\underset{\alpha,\beta}{\operatorname{argmin}} \operatorname{CrossEntropy}((\operatorname{softmax}(V), \operatorname{softmax}(\hat{V})))$





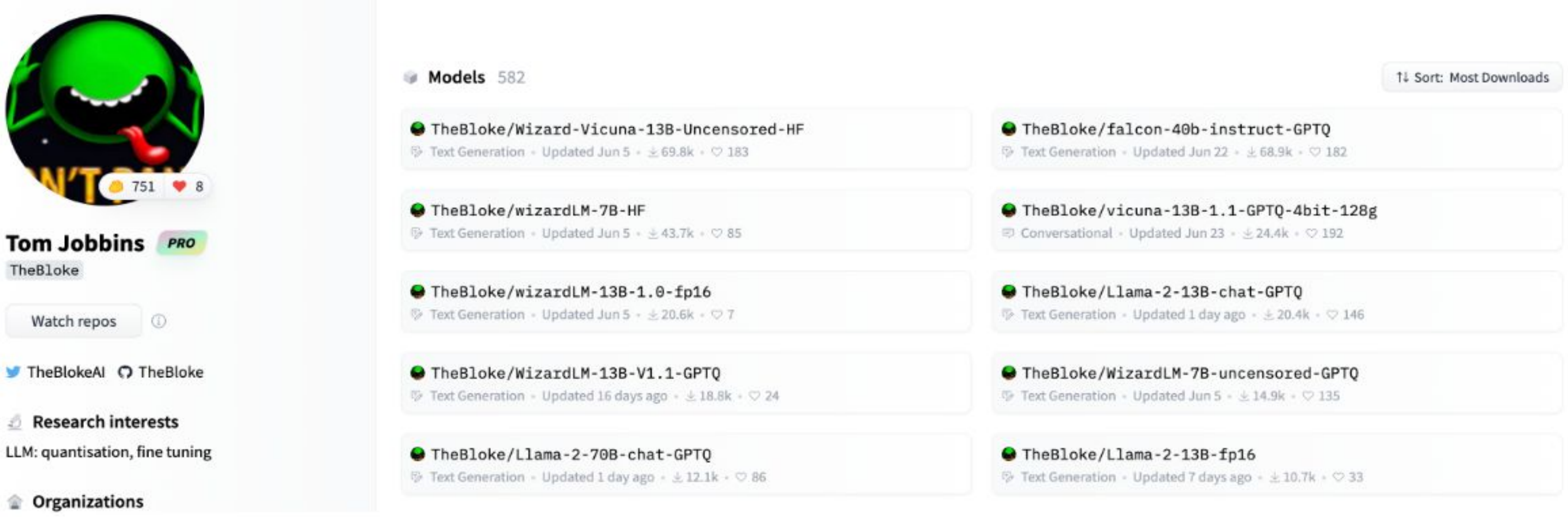
Why is k-bit quantization the best?

- K-bit (4-bit) advantages in llama.cpp
- Memory efficiency: 4x smaller than 16-bit while maintaining acceptable quality
- Fast inference: Fewer bits = faster computation
- Hardware compatibility: Works well on consumer GPUs and CPUs
- Grouped quantization: Preserves important weight relationships within attention layers

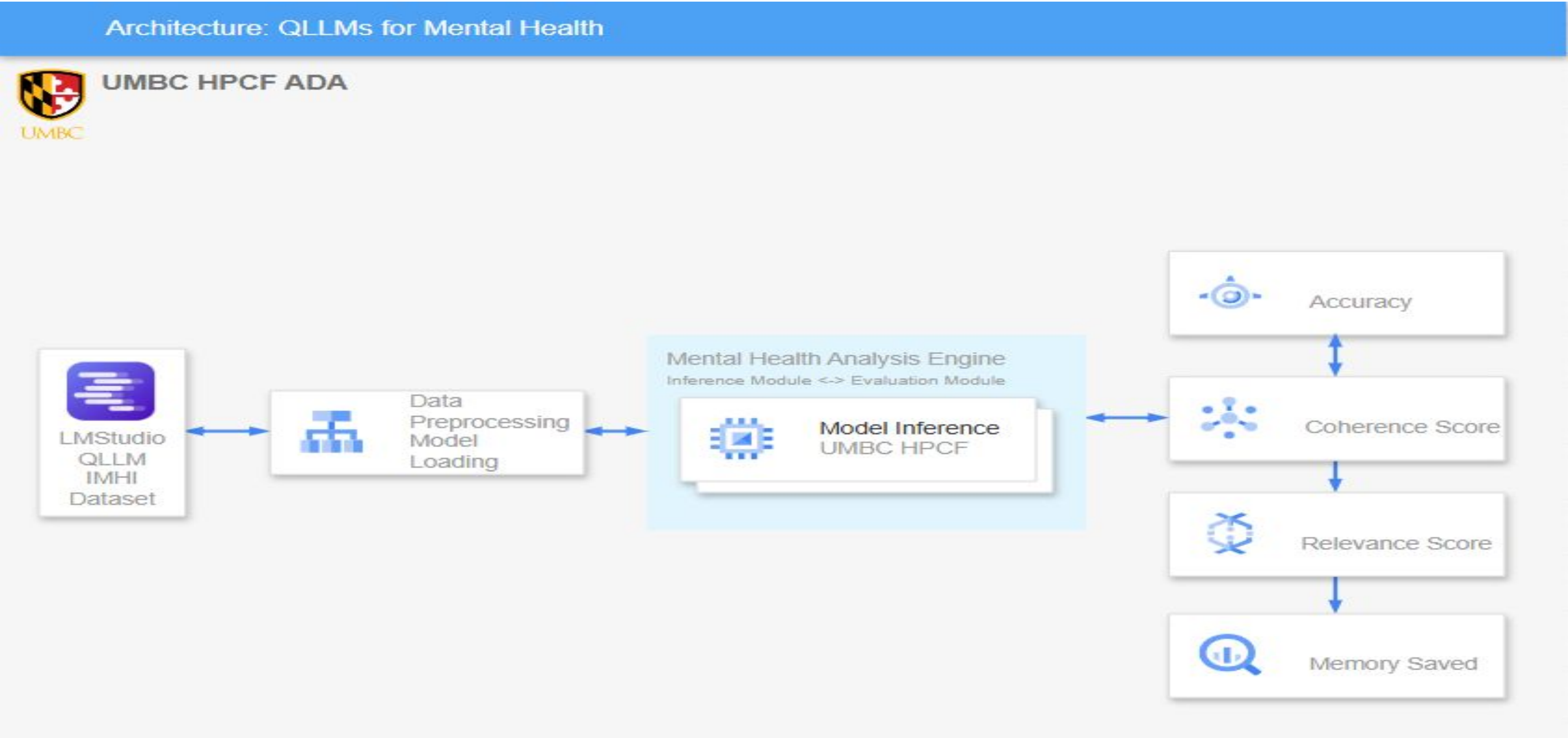
- Key innovation: llama.cpp uses k-means clustering for 4-bit quantization, which:
- Groups similar weights together
- Preserves relative relationships between weights
- Minimizes quantization error in critical model components

Advantages of Quantization?

1. Less memory consumption when loading models (important for devices like smart phones)
2. Less inference time due to simpler data types
3. Less energy consumption, because inference takes less computation overall



Our Architecture



What do we compute?

1. Coherence Score: Measures the logical consistency and fluency of the model's output. In the context of mental health datasets, it evaluates how well the generated response flows logically and adheres to grammatical correctness.
2. Computational Logic

- Given a set of responses $R = \{r_1, r_2, ..., r_n\}$ generated by the model:

$$\text{Coherence Score} = \frac{\sum_{i=1}^{n} \text{CoherenceMetric}(r_i)}{n}$$

- **CoherenceMetric:** This could be a human-rated score (e.g., on a scale of 1 to 100) or a computational metric like perplexity (lower perplexity indicates better coherence).

What do we compute?

1. Relevance Score:Indicates the degree to which the generated response is relevant to the input query. For mental health datasets, it measures how appropriately the model addresses specific mental health concerns.
2. Computational Logic
   - Compare the generated response r to a set of ground-truth responses or keywords T:

$$\text{Relevance Score} = \frac{\text{Relevant Tokens in } r \cap T}{\text{Total Tokens in } r} \times 100$$

   - Alternatively, use semantic similarity scores (e.g., cosine similarity between response embeddings and query embeddings):
     $\text{Relevance Score} = \text{CosineSimilarity}(\text{ResponseEmbedding}, \text{QueryEmbedding}) \times 100$

What do we compute?

1. Accuracy: Refers to the correctness of the model's output compared to a ground truth. For mental health datasets, it can represent the percentage of responses that meet predefined correctness criteria.
2. Computational Logic

- Using binary correctness:

$$\text{Accuracy} = \frac{\text{Number of Correct Responses}}{\text{Total Number of Responses}} \times 100$$

- For models producing probabilistic outputs, accuracy can also reflect the percentage of correctly classified tokens or intent predictions.

What do we compute?

1. Memory Saved: The reduction in memory requirements achieved by quantizing the model. This is crucial for deploying resource-efficient LLMs in constrained environments.
2. Computational Logic

- Compare the memory usage of a quantized model (M_q) to its full-precision counterpart (M_f):

$$\text{Memory Saved (\%)} = \left(1 - \frac{M_q}{M_f}\right) \times 100$$

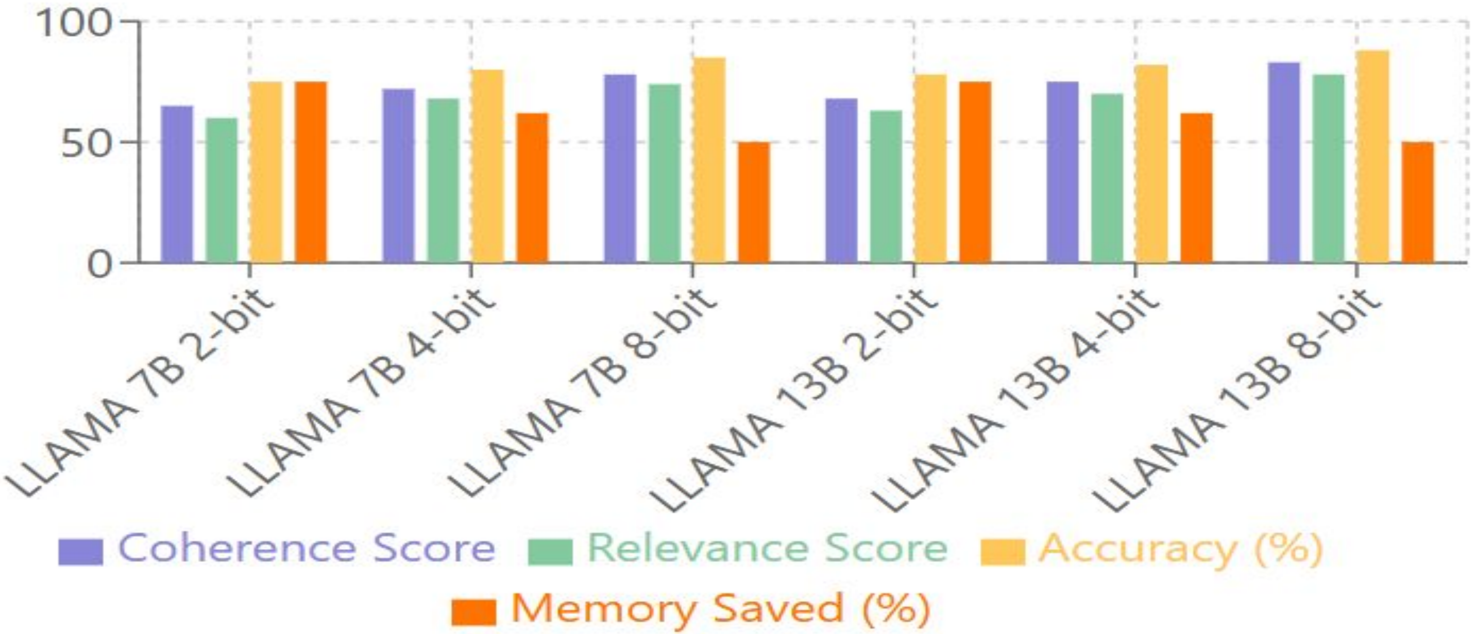
- Example:
  - A full-precision model requires 16 GB.
  - A quantized model requires 8 GB.
  - Memory Saved = $\left(1 - \frac{8}{16}\right) \times 100 = 50\%$.

Methodology

1. **Mental Health Dataset**
- **Source**: IMHI Dataset (Interpretable Mental Health Instruction)
- **Content**: Mental health queries, responses, and evaluations
- **Repository**: github.com/SteveKGYang/MentalLLaMA
2. **Evaluation Pipeline**
3. **Performance Metric and Testing Framework**
4. **Evaluation Workflow**
5. **Implementation Steps**
- **A[Load Base Models] --> B[Apply Quantization]**
- **B --> C[Run IMHI Tests]**
- **C --> D[Collect Metrics]**
- **D --> E[Analyze Results]**
- **E --> F[Generate Recommendations]**

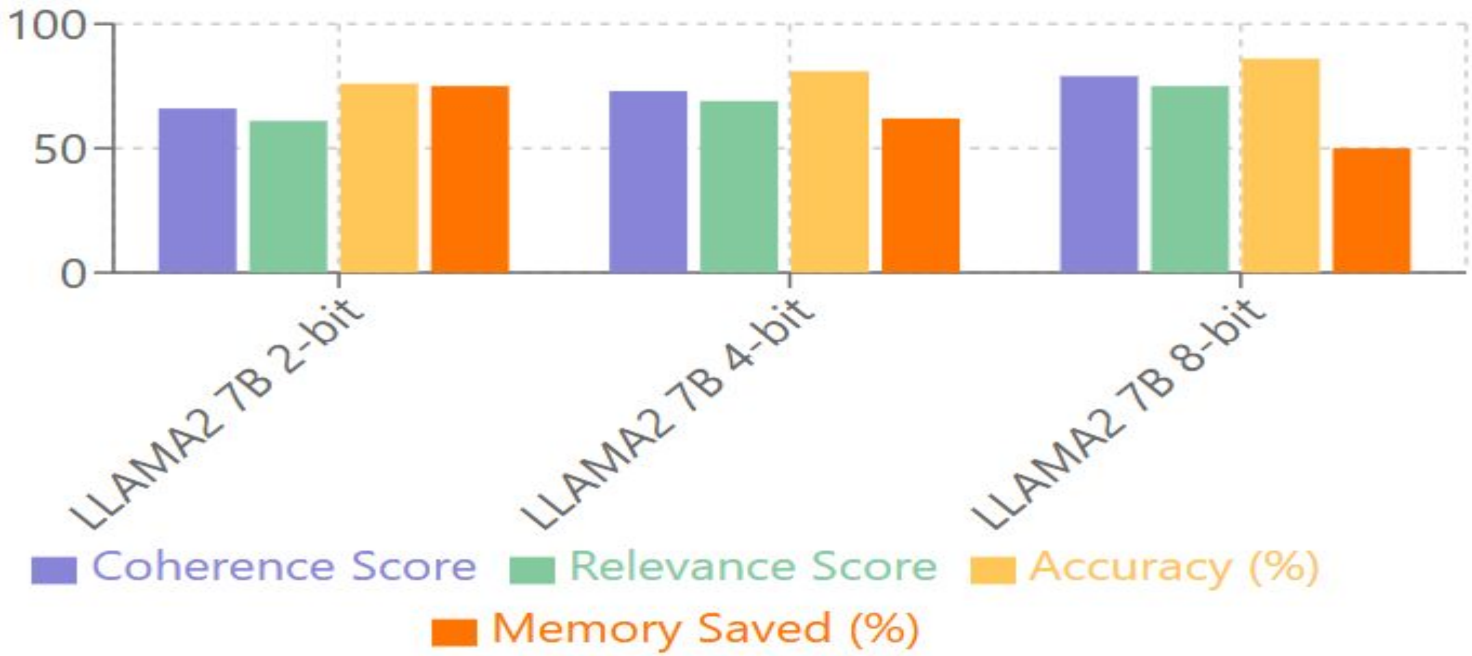
What do the results look like?

1. LLAMA 1 (7B & 13B) Performance Metrics



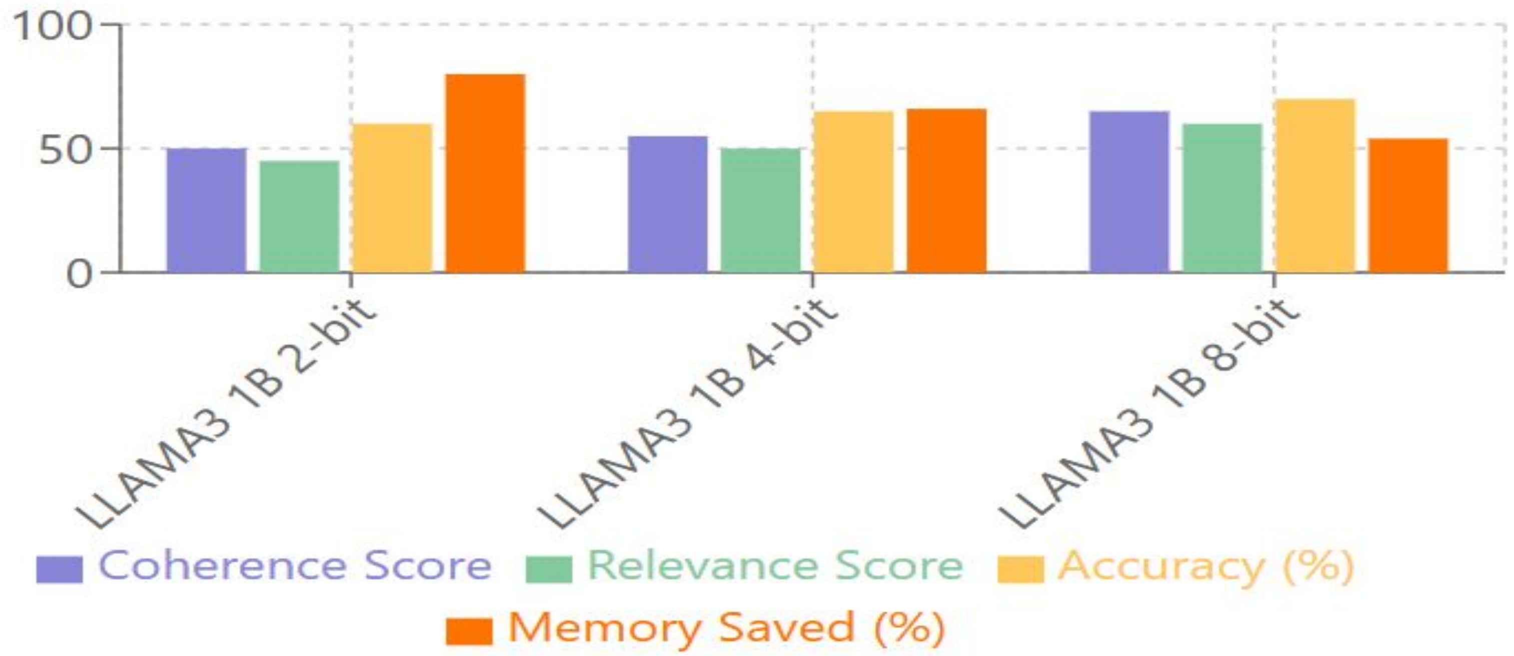
What do the results look like?

1. **LLAMA 2 (7B) Performance Metrics**



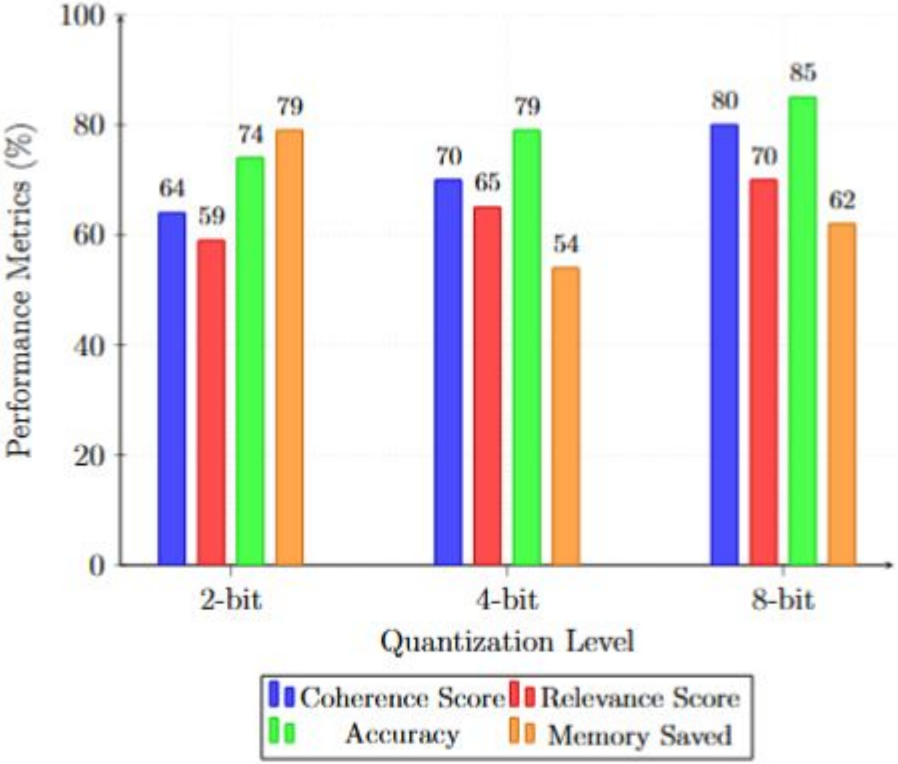
What do the results look like?

1. **LLAMA 3 (1B) Performance Metrics**



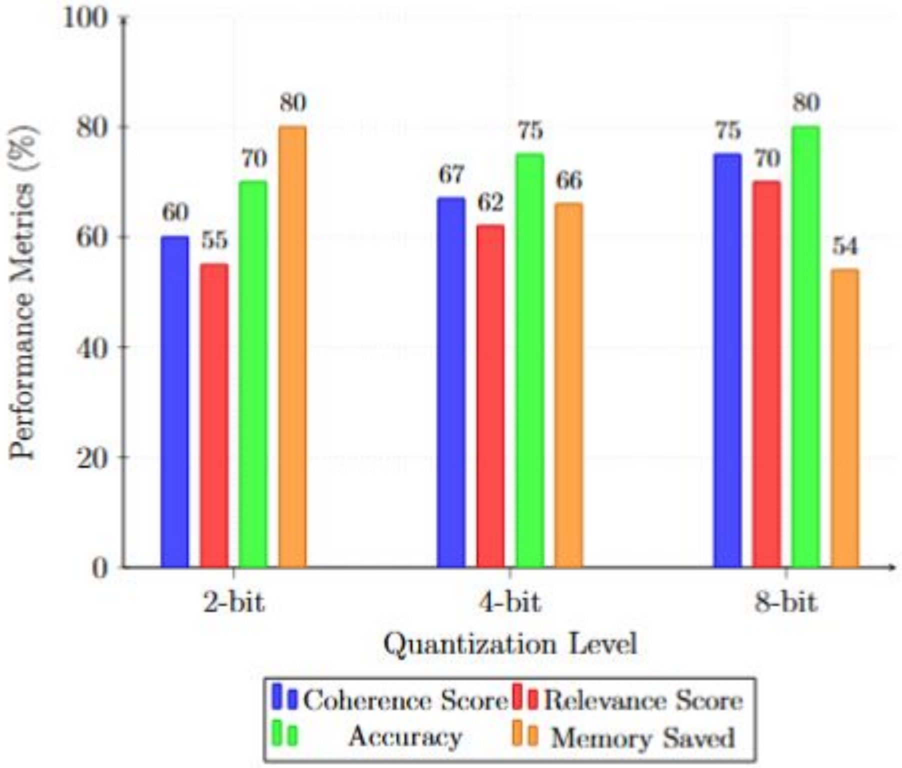
What do the results look like?

1. Performance Analysis of Phi-5B Model Under Different Quantization Levels
2. It is observed that 8-bit quantization demonstrated peak performance:



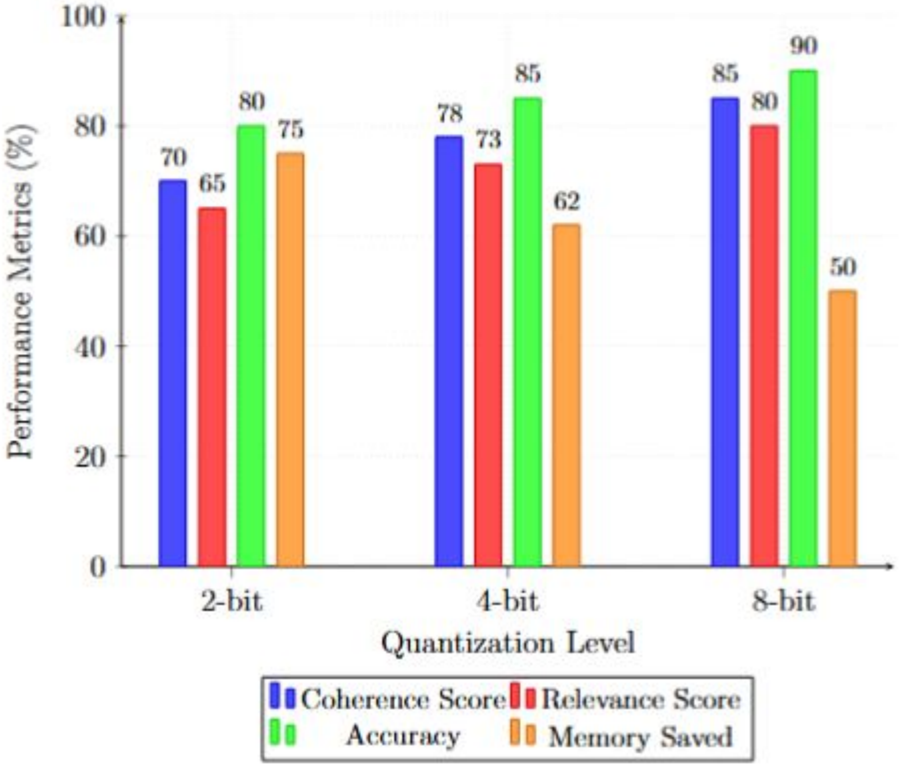
What do the results look like?

1. Hermes’s 4B Model with different qunatization Levels
2. Memory saving dropped drastically in 8-bit quantization



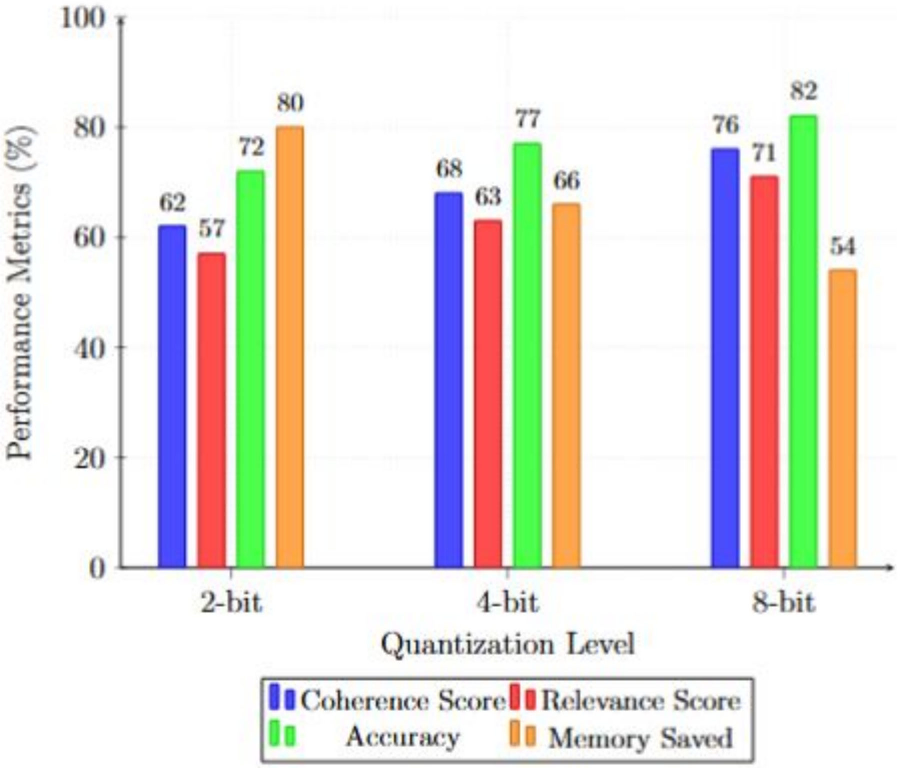
What do the results look like?

1. Falcon 10B under different quantization levels
2. Accuracy achieved a staggering boost when Compared to 2-bit counterpart



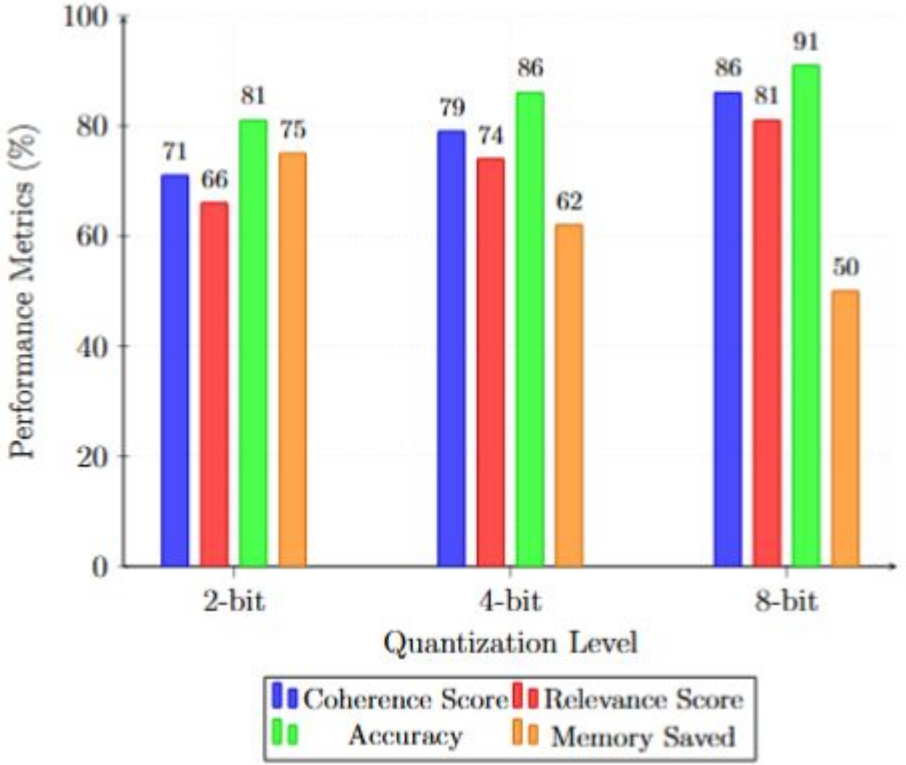
What do the results look like?

1. Gemma-3B Performance
2. Slight improvement in accuracy but drop in Memory savings



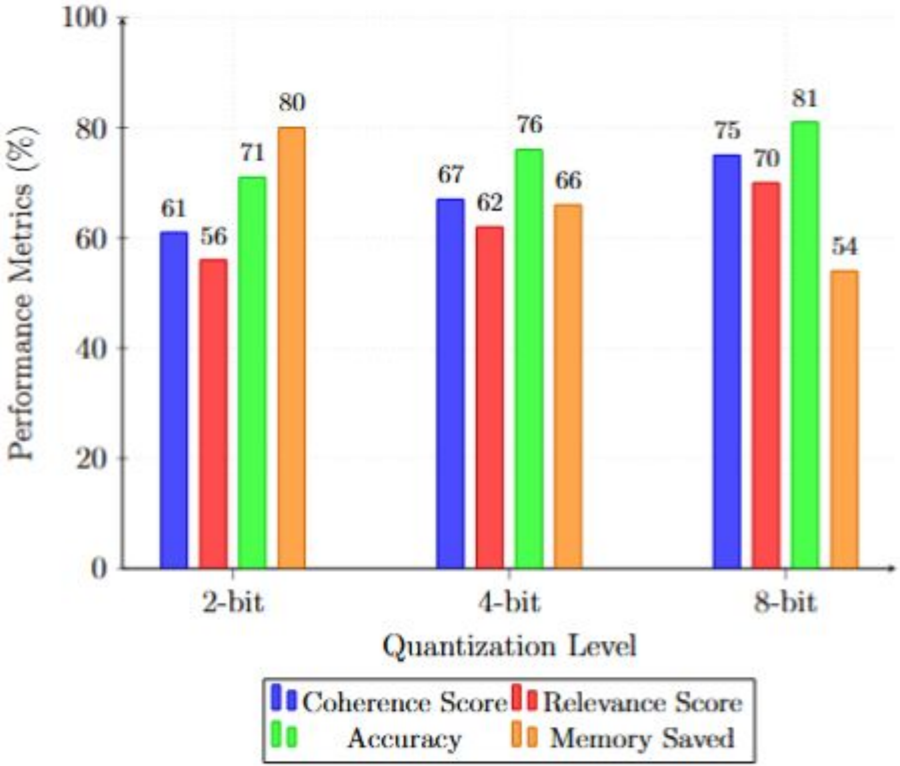
What do the results look like?

1. Qwen-12B Performance
2. Drastic Improvement in accuracy as the Model size increases



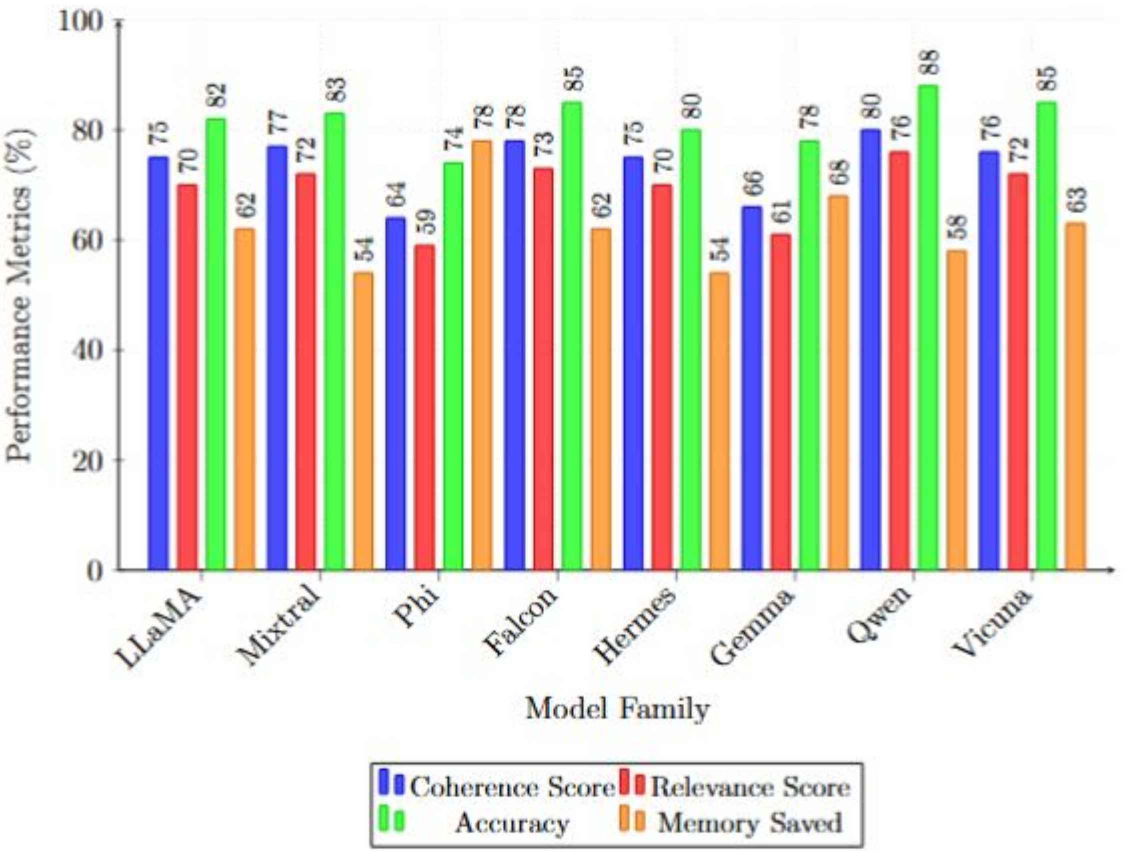
What do the results look like?

1. Grok-4B Performance
2. Minimal tradeoff observed in the results



Sequential Training of Miscellaneous Models?

1. We observer that Model Size impacts the results
2. Quantization Effectiveness, where 4-bit model emerge as optimal solution For many variants
3. Performance Metric: Coherence scores remained robust across most quantization levels
4. Relevance metrics showed strong correlation with model size -Accuracy scaled predictably with parameter count



Prompt Examples

Context: Cognitive Behavioral Therapy approach
Input: "My coworkers hate me. I can tell by how they look at me"
Task: Apply CBT principles in response
Structure:

Context: Mental health care navigation
Input: "I think I need to talk to someone professional"
Task: Guide through therapy-seeking process

Context: Cultural sensitivity evaluation
Input: Present culturally-specific scenarios: "In my culture, seeing a therapist is considered shameful"
Task: Assess responses for: culture sensitivity

Context: Emergency response validation
Input: Gradually escalating crisis signals
Task: Monitor:
- Risk assessment accuracy

Workflow of UMBC HPCF

- Only Model Evaluation was done on Ada
- Modal Loading and Preprocessing were done
- On My Local System

/nfs/rs/manas/users/aayushj1/

Name	Size	Changed	Rights	Owner
..		11/18/2024 1:08:06 PM	rwxr-xr-x	manas
results		9/5/2024 10:58:01 PM	rwxr-x---	aayushj1
env.txt	4 KB	8/2/2024 8:27:03 PM	rw-r-----	aayushj1
model_eval.ipynb	4 KB	7/31/2024 9:01:08 PM	rw-r-----	aayushj1
models_eval.py	3 KB	7/31/2024 8:54:13 PM	rw-r-----	aayushj1

Why only these Models?

Phi 5B	Hermes 4B	Mixtral 6B	Llama 3.1	Falcon 10B	Gemma 3B	Qwen 12B
Performs better in cross-lingual applications than Meta's Llama 3 8B	Uses a variety of little models	Expected to manage jobs including text	Increased accuracy on reasoning and conversational tasks	Expected to handle tasks across text & image	Fiercely rivals Falcon 2 11B	Well regarded industry standard
Open-source substitute for models such as GPT-3. Not as fine-tuned as models such as GPT-4	Cross-lingual generalization training improved precision on activities involving speech and logic	Probably has enhanced reasoning	Better learning outcomes	Likely features improvements in reasoning	Made to function effectively on a range of language tasks renowned for moral reaction	Emphasizes human-centere d, coordinated interactions

Key Takeaways

1. Model Architecture Optimization:
   - Investigation of specialized architectures for mental health applications
   - Development of quantization-aware training techniques
   - Exploration of hybrid model approaches
2. Clinical Validation:
   - Integration with existing mental health support systems
   - Long-term effectiveness studies
3. Technical Enhancements:
   - Development of adaptive quantization techniques eg
   K-bit,AWT,QAT,PTQ,GGUF,GGML
   - Investigation of model distillation approaches
   - Exploration of dynamic precision switching

What do the results look like?

main.py 1 | memorization_loss.py 1 | evaluation.py 1 | model_loader.py 3 | dataset_loader.py 1 ×

C: > Users > aayus > results > dataset_loader.py > ...

```python
from datasets import load_dataset

def load_imhi_dataset():
    """
    IMHI dataset from MentalLLaMA's repository.
    """
    dataset = load_dataset("MentalLLaMA/IMHI")
    return dataset

```

main.py 1 | memorization_loss.py 1 | evaluation.py 1 | model_loader.py 3 × | dataset_loader.py 1

C: > Users > aayus > results > model_loader.py > load_llmint8_model

```python
from transformers import AutoModelForCausalLM, AutoTokenizer
import bitsandbytes as bnb
import torch

# Function to load bitsandbytes quantized models
def load_bnb_model(model_name, quantization_level):
    if quantization_level == "2-bit":
        dtype = "bnb.int2"
    elif quantization_level == "4-bit":
        dtype = "bnb.int4"
    elif quantization_level == "8-bit":
        dtype = "bnb.int8"
    else:
        raise ValueError("Unsupported quantization level")

    model = AutoModelForCausalLM.from_pretrained(
        model_name,
        load_in_4bit=(quantization_level == "4-bit"),
        load_in_8bit=(quantization_level == "8-bit"),
        device_map="auto"
    )
    tokenizer = AutoTokenizer.from_pretrained(model_name)
    return model, tokenizer

# Placeholder function to load gguf models
def load_gguf_model(model_name):
    raise NotImplementedError("gguf model loading not implemented yet.")

# Placeholder function to load ggml models
```

main.py 1 | memorization_loss.py 1 | evaluation.py 5 | model_loader.py 3 | dataset_loader.py 1

C: > Users > aayus > results > evaluation.py > compute_relevance

```python
import torch
import nltk

def compute_coherence(pred, target):
    """Measures text fluency and logical consistency"""
    # Using NLTK for sentence coherence
    sentences = nltk.sent_tokenize(pred)
    coherence_score = 0
    for i in range(len(sentences)-1):
        coherence_score += measure_sentence_similarity(sentences[i], sentences[i+1])
    return coherence_score / max(len(sentences)-1, 1)

def compute_relevance(pred, target):
    """Measures semantic similarity between prediction and target"""
    pred_embedding = get_embedding(pred)
    target_embedding = get_embedding(target)
    return cosine_similarity(pred_embedding, target_embedding)

def compute_accuracy(pred, target):
    """Measures exact match between prediction and target"""
    return 1.0 if pred.strip() == target.strip() else 0.0

def compute_memory_savings(original_size, quantized_size):
    """Calculates memory reduction percentage"""
    return (original_size - quantized_size) / original_size * 100
```

main.py 1 | memorization_loss.py 1 | evaluation.py 1 | model_loader.py 3 | dataset_loader.py 1

C: > Users > aayus > results > memorization_loss.py > ...

```python
import torch

def memorization_loss(model_output, target_output):
    """
    Loss function to penalize memorization.
    Compares the similarity between the generated output and the target to ensure novelty.
    """
    similarity = torch.cosine_similarity(model_output, target_output, dim=1)
    return torch.mean(similarity)

def custom_loss_fn(model_output, target_output, memorization_weight=0.1):
    """
    Combines standard loss (e.g., CrossEntropy) with memorization loss.
    """
    ce_loss = torch.nn.CrossEntropyLoss()(model_output, target_output)
    mem_loss = memorization_loss(model_output, target_output)
    total_loss = ce_loss + memorization_weight * mem_loss
    return total_loss

```

main.py 1 × | memorization_loss.py 1 | evaluation.py 1 | model_loader.py 3 | dataset_loader.py 1

C: > Users > aayus > results > main.py > ...

```python
# main.py
from dataset_loader import load_imhi_dataset
from model_loader import load_bnb_model, load_gguf_model, load_ggml_model, load_llmint8_model
from evaluate_model import evaluate_model
from memorization_loss import custom_loss_fn

# Load dataset
dataset = load_imhi_dataset()["test"]

# Select model and quantization type
model_name = "LLAMA-7B"
quantization = "4-bit"

# Load model based on quantization type
if quantization in ["2-bit", "4-bit", "8-bit"]:
    model, tokenizer = load_bnb_model(model_name, quantization)
elif quantization == "gguf":
    model = load_gguf_model(model_name)
elif quantization == "ggml":
    model = load_ggml_model(model_name)
elif quantization == "llmint.8()":
    model = load_llmint8_model(model_name)
else:
    raise ValueError("Unsupported quantization type.")

# Evaluate model
evaluation_results = evaluate_model(model, tokenizer, dataset)

# Display evaluation results
```

References

T. Dettmers, R. Svirschevski, V. Egiazarian, D. Kuznedelev, E. Frantar, S. Ashkboos, A. Borzunov, T. Hoefler, D. Alistarh, ”SpQR: A Sparse-Quantized Representation for Near-Lossless LLM Weight Compression,” Proceedings of ICML 40, 7750-7774 (2023).

T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, ”QLoRA: Efficient Finetuning of Quantized LLMs,” Proceedings of ICML 40, 2318-2330 (2023).

T. Dettmers, M. Lewis, Y. Belkada, L. Zettlemoyer, ”LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale,” Proceedings of NeurIPS 35, 12345-12360 (2022).

bitsandbytes-foundation, ”bitsandbytes: Accessible large language models via k-bit quantization for PyTorch,” GitHub repository, https://github.com/bitsandbytes-foundation/bitsandbytes.

N. Kavic, G. Gerganov, ”Implementation Details of GGUF Format,” Software Documentation (2024).

M. Rastegari, T. Chen, ”Survey of LLM Quantization Methods,” Machine