Knowledge Distillation vs Quantization: Which Holds the Key to AI's Diet Plan?

Andy Seoho Yun
447
<andyseo@cs.washington.edu>

Mohamed Awadalla
447
<awadalla@.cs.washington.edu>

Arman Mohammed
447
<ibm5@cs.washington.edu>

Project Information:

Fill out this part for the midway report.

Including one of the following three tables depending your project type.

Open-ended Project: If you choose to conduct an open-ended research project, please specify the survey topic in the following table.

Project Type	Open-ended Project
Project Title	Quantization vs Knowledge Distillation

You don't need to fill out this part for the midway report, but you should fill this out for your final report.

Specify the individual contributions.

- **Andy Yun:** Implemented a lot of the evaluation code and provided high-RAM compute to run these different models, as well as making the diagram on the poster with the robots, shoutout Professor Yejin Choi. Also did significant research into which evaluation metrics to use for this project.
- **Mohamed Awadalla:** Implemented most of the Quantization pipeline and created most of the evaluation framework and codebases, required a lot of debugging. Also did significant research into the Quantization pipeline, and how to evaluate it relative to the other models.
- **Arman Mohammed:** Implemented the Knowledge Distillation pipeline and contributed some to the evaluation codebase, required a lot of debugging. Also did signficant research on how to calculate distillation loss by performing a weighted average KL-divergence and cross-entropy loss hybrid formula.

1 Introduction (mandatory)

We tackle the challenge of optimizing the performance of large language models (LLMs) while enhancing their computational efficiency. The research question focuses on determining the more effective method between Quantization and Knowledge Distillation for reducing the size and computational demands of LLMs without significantly compromising their performance. This question is motivated by the increasing need for more efficient AI models in natural language processing to make advanced NLP technologies more accessible and environmentally friendly.

Our methods involved applying quantization and knowledge distillation techniques to the flan-T5-large model, comparing their effectiveness in reducing model size and computational demands. At a high level our findings show that quantization with qint8 datatype retains a high level of accuracy while reducing the size from 3 GB of the flan-T5-large to 0.9 GB for the quantized model. However, using the Hugging Face quantization pipeline for quantization required us to run the quantized model on cpu which drastically slowed down performance. With regards to the Knowledge Distillation results, we were able to get slightly better results than vanilla flan-T5-small with the distilled student model, but worse than the quantized model. Since we could run the student model on gpu we had a 4x speed up from the flan-T5-large teacher model, which was 12x faster than the quantized model. But having implemented the Knowledge Distillation pipeline we realized that the hyperparameter tuning for the specific training pipeline was expensive both time wise and compute wise.

2 Background (optional)

You're encouraged to describe the background of your project here. For instance, you may discuss related work or provide background knowledge about the problem that you study (especially if it involves a specific domain, e.g., chess).

Quantization is just reducing the precision of the model's parameters (e.g., from float32 to float16 or qint8). This can significantly decrease the model size and speed up inference times by allowing models to perform operations with lower-precision arithmetic. Quantization has been shown to be effective in various models, but its impact on model performance can vary, with more aggressive quantization (e.g., qint8) potentially leading to degradation in output quality due to the reduced expressiveness of the model weights.

Knowledge Distillation is a process where a smaller "student" model is trained to mimic the behavior of a larger "teacher" model. The goal is to transfer the knowledge from the teacher model to the student model, making the student model more efficient while trying to preserve as much performance as possible. This technique relies on the idea that a compact model can achieve similar performance to a larger model if it learns to replicate the larger model's outputs closely

3 Method (mandatory)

Describe your current method and how you're planning to improve on it in your final report. Discuss any advantages of your method (e.g., requiring fewer resources), and provide intuition about why your method makes sense for the problem you are trying to solve. Aim for your explanation to be understandable to any other student in the class.

Our methodology involves a two-pronged optimization approach: quantization and knowledge distillation. The initial phase focuses on quantization. In this phase we will take the flan-T5-large model and quantize it to a smaller precision. In this case, we are going from float32 precision to qint8 precision. By taking this step, we are decreasing the size of the model, making it faster and also capable of running on less powerful machines. Although reducing the precision of the weight parameters by this much risks losing a lot of the capability, our evaluation experiments will determine how significant that loss is.

Our second phase will involve distilling knowledge from our large model to a small model. In this case, we use flan-T5-large as the large teacher model and flan-T5-base as the smaller student model. The teacher model is about 750M parameters whereas the flan-T5-small is about 80M parameters. The smaller model is about 1GB which is roughly the same size (in bytes) as the quantized large model which is 0.9GB. From there, we fine-tune the smaller model on outputs from the large model based on the ”Nicolas-BZRD/Parallel_Global_Voices_English_French” English to French translation dataset. We use the first 10000 pairs of data for the fine-tuning process and keep a low learning rate of 1e-5. Considering the size of the student model and the fine-tuning for the specific task of English to French translation, the student model should be more accurate during the evaluation experiments.

Our experiments will explore the success of both of these approaches.

4 Experiments (mandatory)

Describe your current experiments and what future experiments you're planning to run. Be clear about your experimental set-up and metrics for measuring performance.

Our experiments are centered around the translation accuracy of models after employing Knowledge Distillation and Quantization techniques independently within the English to French translation domain from the ”Nicolas-BZRD/Parallel_Global_Voices_English_French” dataset.

In regards to the Quantization setup, we started with the flan-T5-large model and then using the HuggingFace quantization pipeline we reduced the precision of the model's parameters from float32 to qint8. It is important to note the details of the HuggingFace quantization pipeline, especially how it uses dynamic quantization which focuses on linear layers. This process, which converts the model to use qint8 data types for linear layers, aims to reduce model size and potentially speed up inference on CPU-bound environments, but is limited to being run on CPU.

In regards to the Knowledge Distillation setup, we started with the flan-T5-large model as the teacher model and then set up flan-T5-small as the student model. After preparing the first 10k pairs of the English to French dataset, we fine-tuned the student model. During the fine-tuning loop we do a few things: first we get the logits of the student model on the current prompt, then we get the logits of the teacher model on the current prompt, then we calculate a weighted average loss that contains one the KL-divergence between student logits and teacher logits, and two the cross-entropy loss between the student logits and the labels. This weighted average required us to do some hyper parameter tuning and we will discuss the optimal alpha value for this distillation loss function more in the Results section.

Then we took the quantized model which is about 0.9GB and runs on CPU, and the distilled student model which is about 1GB and can run on GPU, and evaluated it on the next 2k pairs of the same English to French Dataset. The key metrics that we were looking at were the BLEU scores which tell us the translation accuracy of the models and ROUGE-L F1 scores which tell us the precision and recall of the models. These scores will represent the key metrics that will determine how effective these two diet plans are for optimizing this performance to size ratio of these models.

Going further in detail, our main experiment is evaluating the vanilla flan-T5-small model, the distilled student model, the quantized model, and the teacher flan-T5-large model on the 2k pair English to French evaluation dataset while keeping track of the BLEU scores and the ROUGE-L F1 scores. Our secondary experiment will be testing all of the same models on an OOD, out of domain, dataset which will illustrate how much of the model's generalizability is retained after going through both of these diet plans (Quantization and Knowledge Distillation).

4.1 Model (optional)

The focus is on flan-T5-large for optimization experiments, we use flan-T5-large for quantizing as well as the teacher model for the knowledge distillation pipeline with the flan-T5-small being our student model.

4.2 Datasets (optional)

We focused the Knowledge Distillation pipeline on a specific task to make sure our experiments were representative of the techniques to their maximum capability. In this light we used ”Nicolas-BZRD/Parallel_Global_Voices_English_French” as our main dataset and we took the first 10k pairs of translations for the trainingfine-tuning dataset and then we used the following 2k pairs for evaluation.

4.3 Baselines (optional)

One baseline will be the outputs of the original flan-T5-large model on our evaluation dataset to determine the upper boundary for performance on the evaluation dataset for both optimization techniques. Our other baseline will be the outputs of the vanilla flan-T5-small model on the evaluation dataset to determine the lower boundary of performance on the evaluation dataset for both optimization techniques. From this we will be able to determine which of these two techniques is more effective at improving the performance to size ratio. We we will be able to compare to the outputs of the quantized model and the distilled model using BLEU score calculations and KL divergence.

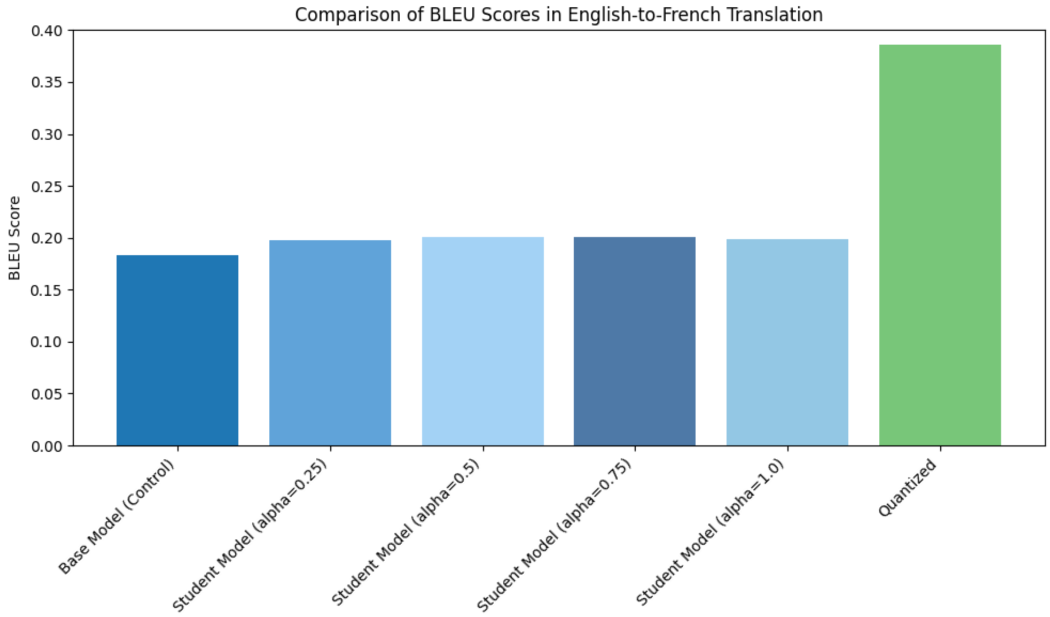
4.4 Code (mandatory)

Quantization vs Knowledge Distillation main evaluation codebase
Quantization pipeline
Knowledge Distillation pipeline
Rouge-Plot Evaluation
BLEU-Plot Evaluation

5 Results (mandatory)

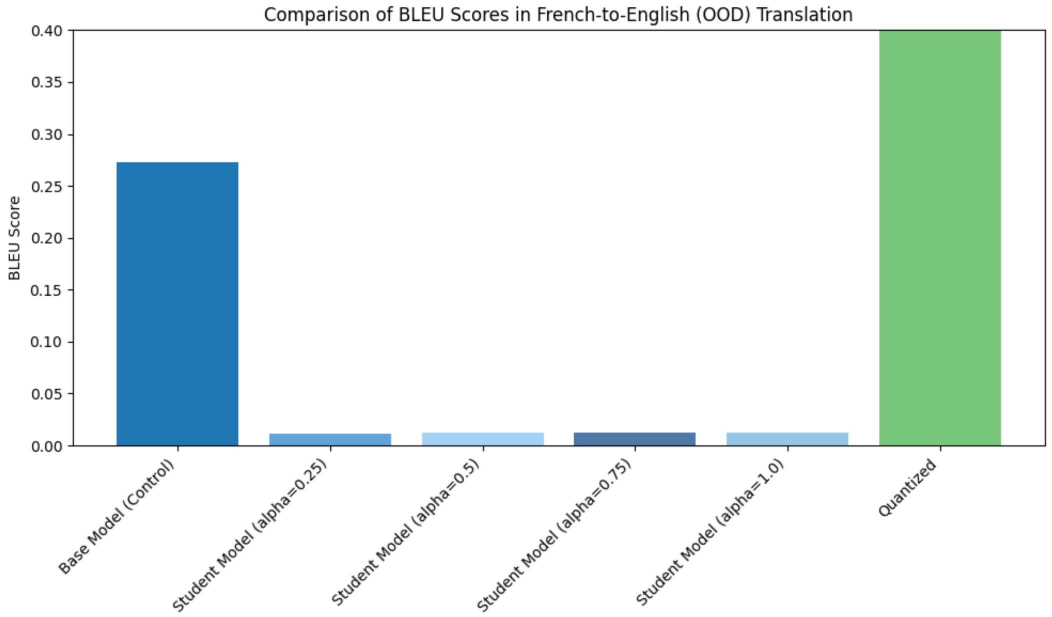
Present and discuss your results. How does your method compare to baselines? Any surprising findings? For the Default 517 project, discuss whether your results match those from the original paper.

At a high level Quantization seems to be a more performant compression technique due to stronger BLEU scores and ROUGE-L F1 scores in comparison to the lower bound baseline and the distilled student model, but the decision of whether to use Knowledge Distillation or Quantization as a compression technique use is a lot more nuanced.

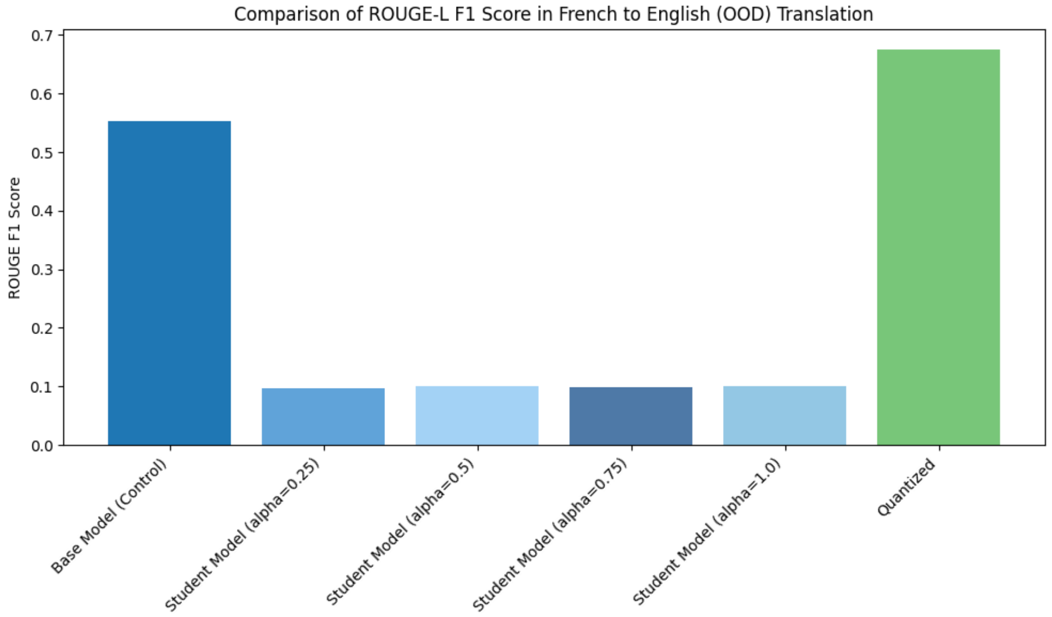


As the chart indicates we can see how much better the Quantized model is relative to the other models ran on the evaluation dataset of 2k pairs of English to French translation. Additionally we evaluated different student models from the Knowledge Distillation pipeline, fine-tuned with different alpha values for the weighted average distillation loss. The chart indicates that the most accurate model was trained with an even weighting of the KL-divergence, between student and teacher outputs, and the ground-truth cross entropy loss. This illustrates that the Knowledge Distillation pipeline requires a precise balance of hyperparameters to properly extract the most representative performance of the technique, which is costly both time-wise and compute-wise.

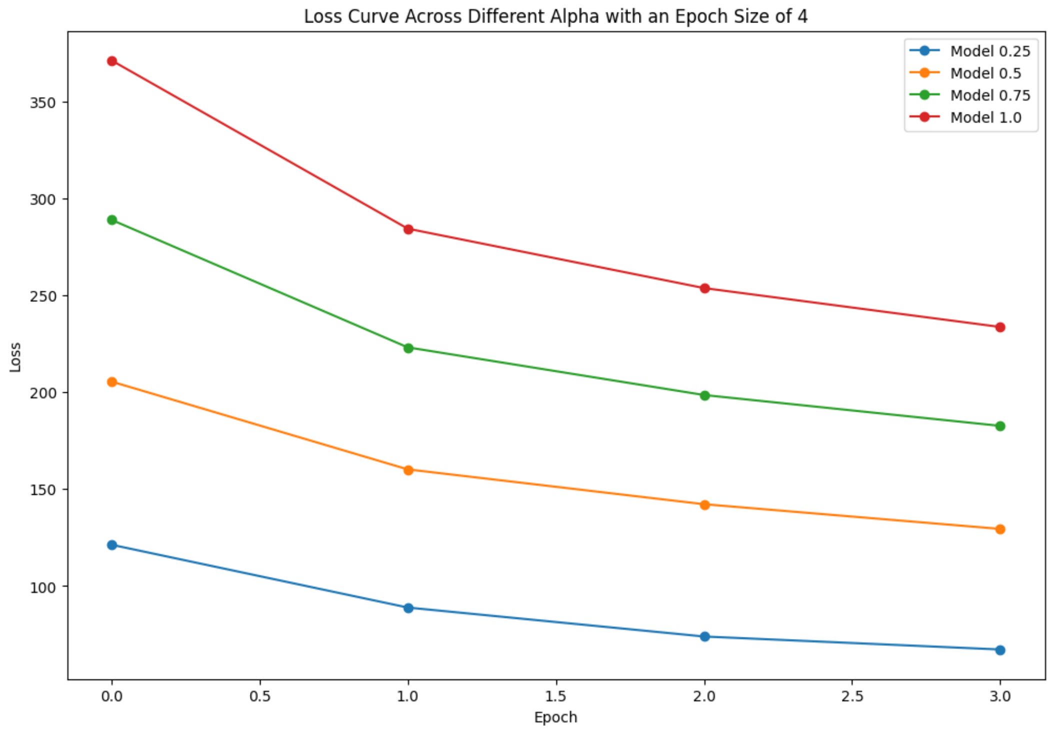
Considering the massive improvement in translation accuracy of the Quantized model, it seems like a no brainer to conclude that Quantization is better than Knowledge Distillation, the crux of this research project; however, there are some big tradeoffs especially with respect to having to run the Quantized model on CPU. This led to significantly slower evaluation times, meaning the Quantized model would run 12 times slower than the base vanilla model and the distilled student model. For 2k generations the quantized model took 12 minutes, while the distilled student model took 1 minute. So efficiency was a big tradeoff in our case because of the way HuggingFace does dynamic quantization. If we implemented static quantization we would be able to run on GPU, and we can re-assess the performance of Quantization as a technique.



Now applying these models to out of domain applications, we can see how poorly the distilled-student model generalizes. This illustrates that we are losing data to the fine-tuning process, and that is a very significant tradeoff. First of all to properly perform Knowledge Distillation there needs to be a specific dataset/task to fine-tune on, which limits the generalizability of the Knowledge Distillation technique, and now seeing the results, you can see how the Knowledge Distillation pipeline loses significant capabilities that the vanilla model had. So generalizability is a big problem with Knowledge Distillation



Repeating the same trend with regards to precision and recall of the different models as represented by the ROUGE-L F1 scores respectively, you can see how the student model regardless of the alpha value consistently performs worse than the vanilla and quantized models. Further illustrating the issue of forgetting during the Knowledge Distillation pipeline.



Finally, going deeper into Knowledge Distillation, we can see how the different alpha values correlate to loss and even the loss optimization trend. Keep in mind we used AdamW which is the state of the art loss optimization function. From this it seems that the highest performing model should be the distilled student model with alpha value of 0.25, which means 25% of the loss will be determined by the teacher model's outputs, and the other 75% of the loss will be determined by the ground truth. Interestingly, if we correlate this with the BLEU Scores for in domain evaluation on English to French dataset we can see that the model with an alpha value of 0.5 actually achieves the highest accuracy relative to all other distilled student models.

6 Discussion (optional)

This is a flexible space for you to use. For instance, you might discuss implications of your results for the broader NLP community, hypotheses about why you see the results you do, or any insights, confusions, and new curiosities stemming from your project.

Our exploration into the optimization techniques of Quantization and Knowledge Distillation for Large Language Models (LLMs) has yielded critical insights. Our experiments with the flan-T5-large model revealed that quantization, particularly with qint8 precision, effectively reduces the model size while largely preserving its accuracy. However, the necessity to run the quantized model on the CPU, a limitation of using the Hugging Face dynamic quantization pipeline, leads to a significant decrease in computational speed.

Knowledge Distillation provided some benefits over the baseline flan-T5-small model in terms of improved accuracy, but it was not as effective as the quantized model. Despite this, the distilled model's ability to run on the GPU and achieve faster inference times compared to the quantized model presents a notable advantage. Additionally, its important to note the catastrophic forgetting that the distilled student model displayed after evaluating on the French to English as well as after testing on another OOD dataset of Korean to English. Nonetheless, the process of hyperparameter tuning in Knowledge Distillation is both time-consuming and computationally expensive, representing a significant investment of resources.

The findings from this study inform the trade-offs between model size reduction and computational efficiency. They suggest that the choice between Quantization and Knowledge Distillation should be context-driven, guided by the specific requirements and constraints of the deployment environment.

7 Conclusion (mandatory)

In this section, you should briefly summarize your contributions, state the key takeaways, and potentially mention directions for future work.

In conclusion, this study has underscored the potential of qint8 quantization to shrink LLMs effectively while retaining accuracy, albeit with the trade-off of slower computational performance due to CPU requirements. Knowledge Distillation, while beneficial for speed enhancement on GPUs, demands extensive hyperparameter optimization and may result in lower accuracy compared to quantization.

These outcomes contribute to the understanding of LLM optimization and are particularly relevant for applications where model size and computational resources are limiting factors. Future work could investigate the balance between model performance and optimization in static quantization, extend Knowledge Distillation across a variety of NLP tasks, and explore these optimization techniques in different operational contexts.

This work advances the field of NLP by detailing the comparative impacts of Quantization and Knowledge Distillation on LLM efficiency, providing a foundation for further research into accessible and sustainable AI technologies.

References

A Appendix

You may include other additional sections here if needed.

Acknowledgments

We want to acknowledge Gary for his great feedback during the poster session and thanks to Ximing for the feedback that was necessary for making this project. This template is modified from the COLM 2024 paper template. Instructions are written by Liwei Jiang, Alisa Liu, and Yegor Kuznetsov.