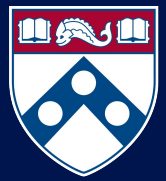
From BART to Edge

CIS 5300 - Final Presentation

Contributors: Xuanyou Liu, Prekshi Vyas, Raksha Ramesh, Manurag Khullar



Penn Engineering

University of Pennsylvania

Illustrative Example

- **Scenario:** A researcher finds themselves in a foreign country, unable to communicate effectively in the local language. To navigate this challenge, they decide to train a large language model (LLM) capable of translating the foreign language into English with high accuracy.
- **Challenge:** The researcher's personal computer lacks sufficient GPU resources, making it impossible to load and process all the model's parameters simultaneously.
- **Solution:** By employing fine-tuning techniques optimized for parameter efficiency, the researcher adapts the model to perform well within the constraints of their limited hardware.
- **Result:** The researcher successfully develops a custom LLM that delivers competitive translation performance in the target domain, all while leveraging their own computational resources.

Problem Statement

Focus: Improving Chinese-to-English (Zh→En) Machine Translation (MT) for Specific Domains

Approach: Leveraging resource-efficient, compact model architectures alongside fine-tuning techniques or quantized large models.

Goal: Identify the most effective method for developing task-specific, smaller-scale translation models.

Outcome: Minimize reliance on massive, server-based networks during fine-tuning. Empower users to fine-tune machine translation models on personal devices, enabling localized, efficient solutions.

Why We're Excited About This Problem

- Makes robust, high-quality translation more accessible by reducing reliance on large servers.
- Enables performance improvements in smaller models to run and update directly on user devices.
- Supports high-quality, domain-specific performance within constrained resource environments.
- Extends the reach of machine translation (MT) technology to scenarios with limited power, connectivity, and computing resources.
- Broadens the real-world impact of MT by bringing its benefits to a wider range of practical applications.

Formal Definition of the Problem Statement

- Let X represent the set of source language sentences in Chinese.
- Let Y represent the set of target language sentences in English.
- The objective is to learn a function $f:X \rightarrow Y$ such that:
  - For each $x \in X$, $f(x)=y$ where $y \in Y$.
  - y accurately reflects the meaning of x.
  - y respects the grammar rules of the target language (English).
  - y achieves high quality as measured by the chosen metric (e.g., BLEU Score).
  - The function f(x) operates within the bounds of available hardware resources, such as memory and processing power, suitable for deployment on edge devices or cloud servers.

Relevance to Topics we Learnt in Class

- The project demonstrates practical applications of theoretical concepts learned in the class:
  a. Employing methods like Layer Freezing and LoRA that align with the principles of Lottery Ticket Hypothesis and QLoRA.
  b. Addressing the challenge of deploying large transformer models in resource-constrained environments, similar to the goals of Distillation and Sparse Attention techniques.

New Topics Explored

- SOTA approaches to Machine Translation Task.
- Explored empirical comparisons among different approaches.
- Different evaluation metrics used for validation of MT.
- Leant about new and upcoming small models.

Problem Dataset

UM-Corpus: A Large-Scale English-Chinese Parallel Dataset

Overview: Designed for **Statistical Machine Translation (SMT)**, UM-Corpus contains approximately **15 million English-Chinese parallel sentences**.

- **Public Access:** Includes **2+ million training sentences** and **5,000 testing sentences** available for research.
- **Versatility:** Spanning **8 diverse domains**, the dataset is ideal for:
  - **Cross-language Information Retrieval.**
  - **Data-driven NLP tasks** like bilingual embeddings and domain adaptation.
- **Key Strengths:** Large-scale, high-quality, and adaptable for various NLP research applications.

	News	Spoken	Laws	Thesis	Education	Science	Subtitle	Microblog
Articles	173,994	-	64,630	26,853	-	3,158,755	-	61,080
Sentences	4,989,478	275,652	328,642	1,302,750	4,725,846	3,158,755	1,011,543	61,080
Percentage	31.59%	1.75%	2.08%	8.25%	29.92%	20.00%	6.41%	2.08%

For the purpose of the this project we have focused on two domains: **Science** and **Education**

Evaluation Metrics - BLEU Score

What is BLEU (Bilingual Evaluation Understudy)?

- Measures translation quality by comparing the model's output to **reference translations**.
- Evaluates **n-gram precision** and penalizes overly short translations with a **brevity penalty**.

Why Use BLEU for This Task?

- **Standardized Evaluation:** Provides a common metric to compare different model versions.
- **Quantifies Quality:** Measures lexical accuracy and alignment with reference translations.
- **Scalability:** Works well for large-scale datasets and multi-reference evaluation.
- **Progress Tracking:** Easily tracks improvements across fine-tuned models and different parameter settings.

Key Advantages for This Task:

- Enables evaluation of translation quality across domains.
- Helps gauge the impact of techniques like **LoRA** and **Layer Freezing** on output fidelity.
- Objective, reproducible metric for reporting results in research.

Strong Baseline: Fine-Tune with All Layers

Step 1: Hyperparameter Tuning - General

We first performed hyperparameter tuning on 10,000 samples to find the best learning rate, epoch number and batch size. The best combinations can be found as followed:

Model	Domain	Best Learning Rate	Best Batch Size	Epochs
mBART	Science	2e-05	16	1
mBART	Education	2e-05	16	1
M2M100	Science	2e-05	16	1
M2M100	Education	2e-05	16	3

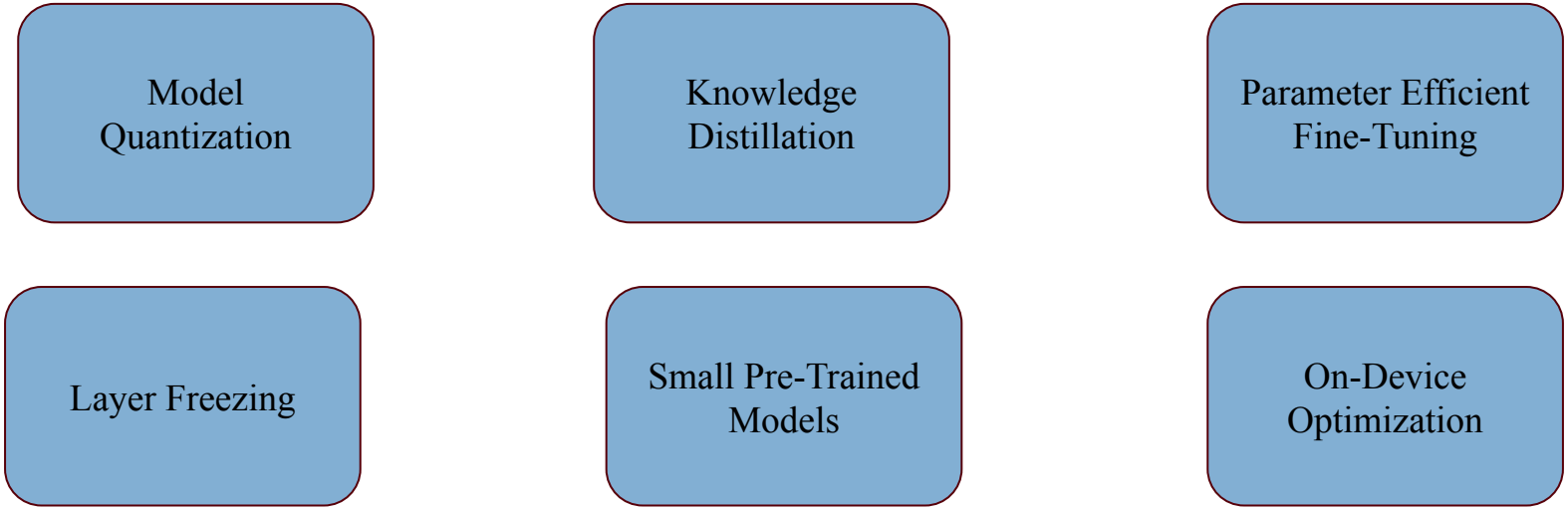
Step 2: Fine-Tuning with All Layers

Using the best best hyperparameter combinations, we performed fine-tuning on Education and Science topic corpus with 50,000 sample size. During fine-tuning, we didn't freeze any layer, meaning that all parameter will be updated.

This method proved to be very effective, resulting in significant improvement in BLEU score compared with baseline model.

Model	Domain	Baseline BLEU Score	Fine-Tuned BLEU Score	Improvement (%)
mBART	Science	0.1223	0.1997	63.28
mBART	Education	0.1175	0.1232	4.85
M2M100	Science	0.0231	0.1200	419.48
M2M100	Education	0.0342	0.0833	143.56

Prior Approaches for MT on Edge Devices



Challenges: Model quantization often sacrifices accuracy, especially for complex tasks. Knowledge distillation is resource-intensive and may yield suboptimal results compared to larger models. Parameter-efficient fine-tuning techniques trade performance for efficiency, while on-device optimizations can be inconsistent across hardware. Lastly, small pre-trained models lack flexibility for out-of-domain content.

Experiment I: Fine-Tune with LoRA

Step 1: Hyperparameter Tuning - LoRA Specific

Retain the general hyperparameters (learning rate, epoch number, and batch size), we performed LoRA specific Hyperparameter Tuning in this section to find the best rank value r, scaling factor α, and dropout rate.

Model	Domain	Rank Values (r)	Scaling Factor (α)	Dropout Values
mBART	Science	8	64	0
mBART	Education	8	64	0
M2M100	Science	8	64	0
M2M100	Education	8	64	0

It's interesting that they are stable across models and topics!

Step 2: Fine-Tuning with LoRA

Using the best best hyperparameter combinations, we performed fine-tuning on Education and Science topic corpus with 50,000 sample size. We freeze all layers except LoRA matrix. Therefore, only weights in LoRA matrix will be updated

LoRA Method can improve BLEU score in most cases but less effective then regular method.

Model	Domain	Baseline BLEU Score	LoRA BLEU Score	Improvement (%)
mBART	Science	0.1223	0.1352	10.55
mBART	Education	0.1175	0.1153	-1.87
M2M100	Science	0.0231	0.0976	322.51
M2M100	Education	0.0342	0.0782	128.65

Experiment II: Fine-Tune with Layer Freezing

Step 1: Hyperparameter Tuning - Layer Freezing Specific

Retain the general hyperparameters (learning rate, epoch number, and batch size), we performed layer freezing specific hyperparameter tuning in this section to find the best frozen encoder layer number and frozen decoder layer number.

Model	Domain	Frozen Encoder	Frozen Decoder
mBART	Science	8	8
mBART	Education	8	8
M2M100	Science	8	8
M2M100	Education	8	8

8 is the smallest frozen number we provided. The result might indicate that the translation is a complicated problem has a pretty high intrinsic rank.

Step 2: Fine-Tuning with Layer Freezing

Using the best best hyperparameter combinations, we performed fine-tuning on Education and Science topic corpus with 50,000 sample size. We freeze the first 8 layers of encoder and first 8 layers of decoder, leaving the rest layers trainable.

Layer freezing is more effective than LoRA but less effective than regular method.

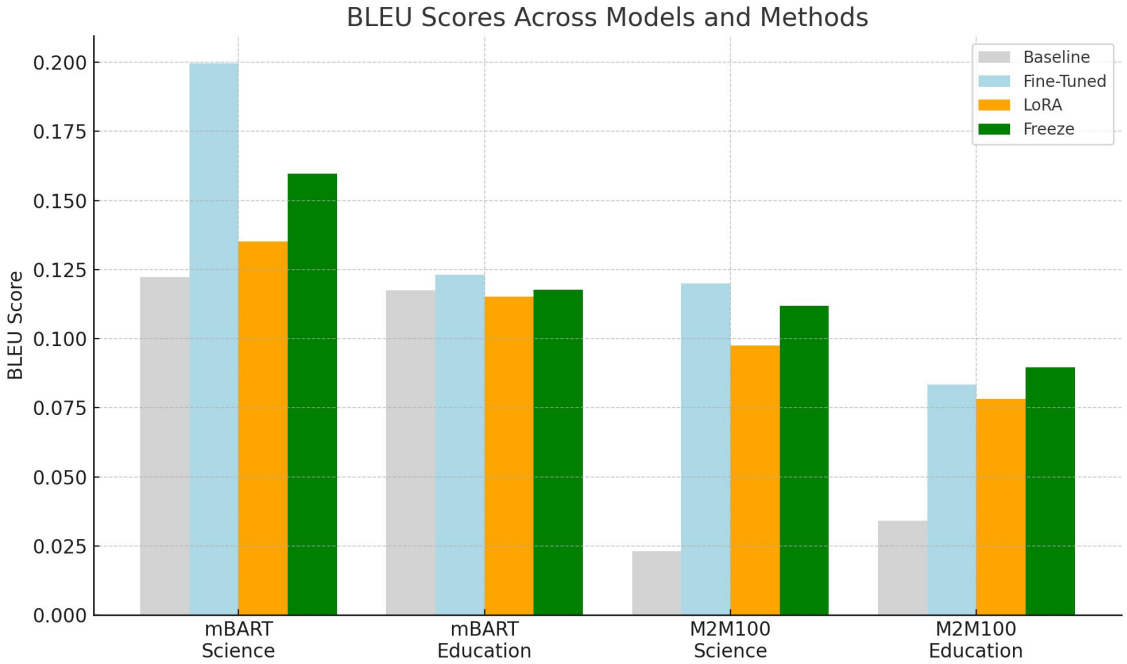
Model	Domain	Baseline BLEU Score	Freeze BLEU Score	Improvement (%)
mBART	Science	0.1223	0.1597	35.58
mBART	Education	0.1175	0.1178	0.026
M2M100	Science	0.0231	0.1118	383.98
M2M100	Education	0.0342	0.0896	161.99

Evaluation: BLEU Scores

1. Domains: Fine-tuning achieved the highest BLEU scores in both domains, with **Science** showing better improvements than **Education**.

2. Models: mBART outperformed **M2M100** across all methods and domains, especially in regular fine-tuning.

3. Methods: **Fine-tuning > Layer Freezing > LoRA** across most cases. Layer freezing outperformed LoRA by 2-18% but trailed fine-tuning by 4-20%.



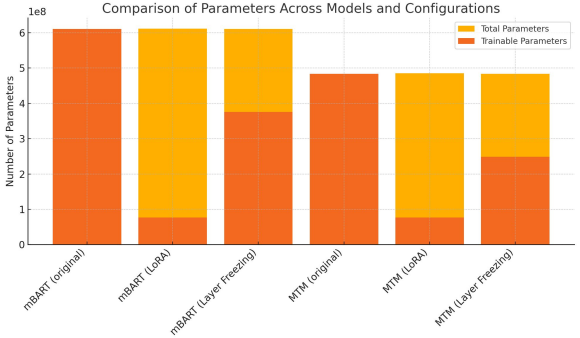
Penn Engineering

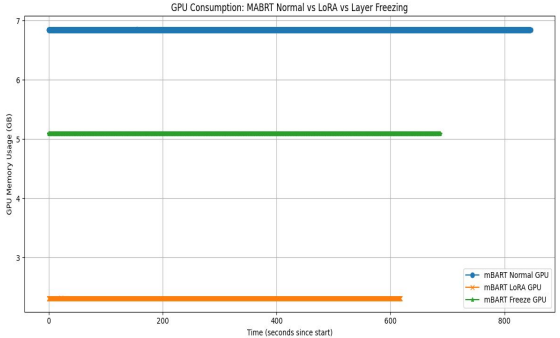
Evaluation: Parameters and Time Efficiency

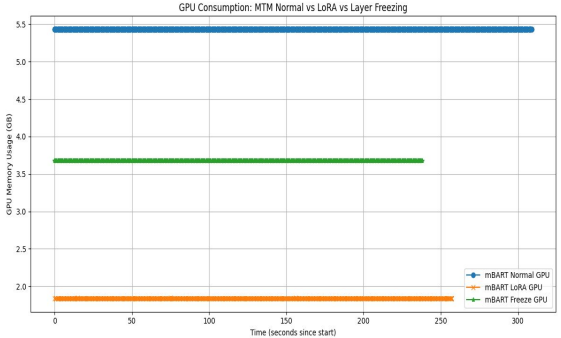
Normal Fine-Tuning: Using all parameters results in the highest GPU memory usage (about 7 GB for mBART and 5.5 GB for MTM) and considerably longer training times compared to the other methods.

Layer Freezing: Freezing roughly half the parameters cuts memory usage to about 4.5 GB for mBART and 3.5 GB for MTM. Training is shorter than normal fine-tuning but similar in duration to LoRA.

LoRA: This approach is the most parameter-efficient, using only around 2 GB of memory for both models, and it maintains stable memory usage throughout training.







Penn Engineering

Fine-Tuning Methods Selection

Normal Fine-Tuning: Achieves the best BLEU scores but has the highest GPU usage (~7 GB for mBART, ~5.5 GB for MTM) and training time. Ideal for maximum accuracy when resources are sufficient.

LoRA: Most memory-efficient (~2 GB) with moderate BLEU score trade-offs. Best for resource-constrained environments or when efficiency is critical.

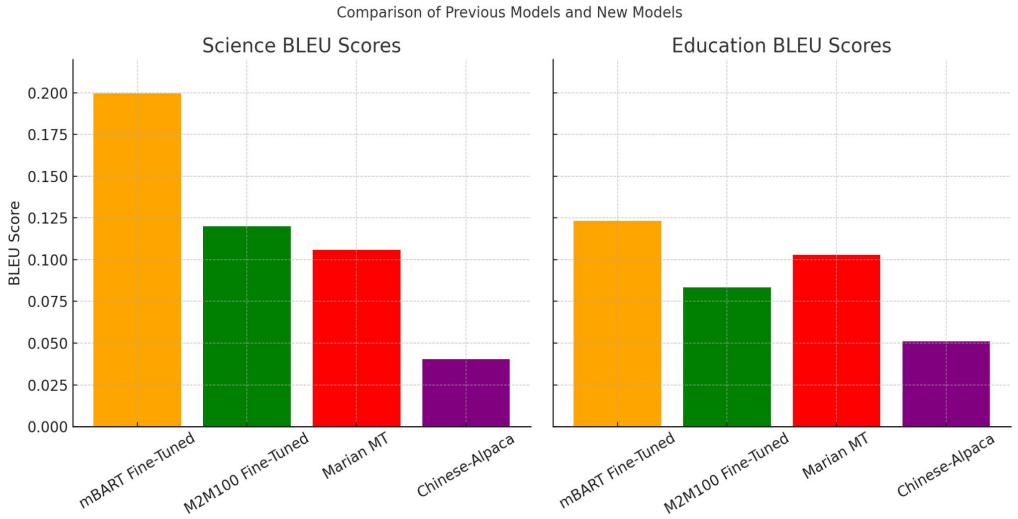
Layer Freezing: Balances performance and efficiency, with reduced GPU usage (~4.5 GB for mBART, ~3.5 GB for MTM) and reasonable BLEU scores. Suitable for balanced needs.

Quantization and Language-Specific Models

In the previous section, we explored **multi-language machine translation models** and various **fine-tuning techniques**. However, we now aim to evaluate whether **large quantized general-purpose models** and **compact, language-specific models** can remain competitive in translation tasks while being small enough to run on personal computers.

- To represent pre-quantized models, we selected "Chinese-Alpaca," a fine-tuned version of LLaMA with **7 billion parameters** optimized for Chinese content. Its quantized version reduces storage requirements by using a **4-bit format**, requiring only **4 GB of storage**.

- For language-specific models, we evaluated "Marian MT," a Chinese-to-English translation model with **60 million parameters**, designed exclusively for this language pair.



Penn Engineering

Translation Error Analysis

Analyzed errors across mBART and M2M100 models under the following methods: Base, Fine-Tuned, Freeze, and LoRA on Education Topic:

Word-Level Errors: Issues with word choice, missing words, or extra words.

- **Ground Truth**: *Though he is fifteen, he has a mental age of less than five.*
- **Prediction**: *Although he is 15 years old, his intellectual age is less than 5 years old.*

Structural Errors: Sentence-level inconsistencies, such as grammar issues or word order mismatches.

- **Ground Truth**: *And few companies offer more products for the management, conversion, distribution and minimization of power than Fairchild.*
- **Prediction**: *From the beginning to the end we adhere to one strategy: to become the world's leading provider of high performance products for many markets.*

Other Errors: Rare issues like nonsensical or untranslated outputs (Likely translate to other languages).

- **Ground Truth**: *Simplify the axiom system of lattice implication algebras, which was given by Y.*
- **Prediction**: *与えられた代数軸系を含む代数系があり、別の軸系を提示します。*

Translation Error Analysis

M2M100 Models:

- **LoRA**: Highest **Word-Level Errors** (3,960), with frequent missing/substituted words and structural misalignments (2,057).
- **Fine-Tuned**: High **Word-Level Errors** (3,884) and **Structural Errors** (2,085), reflecting unstable sentence structures.
- **Base/Freeze**: Moderate **Structural Errors** (~1,500–1,900); dominated by **Word-Level Errors**.

mBART Models:

- **LoRA/Fine-Tuned**: Lower **Word-Level Errors** (~3,250), occasional synonyms or word omissions; minimal **Structural Errors** (~1,300).
- **Base/Freeze**: Stable grammar with fewer structural inconsistencies (~1,200); errors mainly in word substitutions.

Model	Word-Level Errors	Structural Errors	Other Errors
M2M Base	3,652	1,463	6
M2M Fine-Tune	3,884	2,085	2
M2M Freeze	3,813	1,900	2
M2M LoRA	**3,960**	**2,057**	3
mBART Base	3,214	1,212	5
mBART Fine-Tune	3,250	1,278	12
mBART Freeze	3,204	1,268	4
mBART LoRA	3,262	1,313	11

Penn Engineering

Conclusion

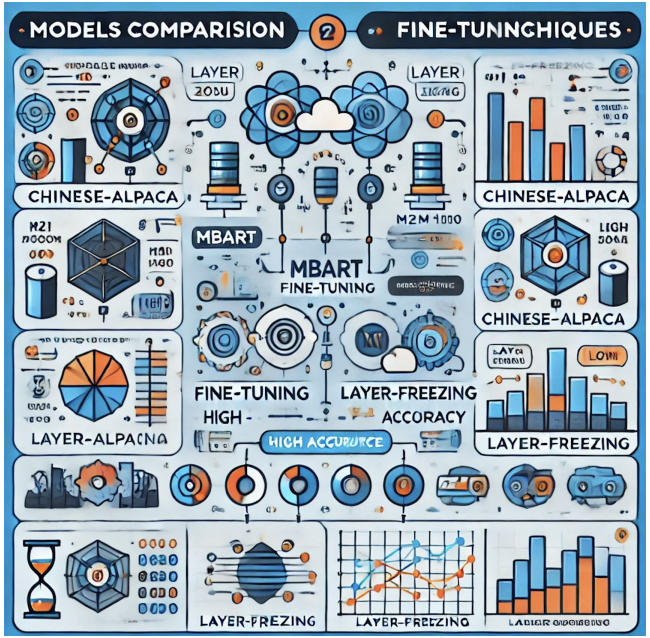
In this project, we evaluated various fine-tuning methods on different machine translation models to identify a **parameter-efficient approach** for enabling users to train personalized models tailored to their specific needs.

We selected **mBART** and **M2M100** as base models and tested three fine-tuning techniques: **traditional fine-tuning**, **LoRA**, and **Layer Freezing**. Our findings revealed:

- **mBART** outperforms **M2M100** in general.
- **Traditional fine-tuning** delivers the best performance but demands the most resources.
- **LoRA** has the lowest performance but is the most resource-efficient.

As an extension, we explored:

- **Large pre-quantized models (e.g., LLaMA)**: Found to be ineffective for machine translation tasks.
- **Small language-specific models (e.g., Marian MT)**: An excellent choice for resource-constrained scenarios.



Penn Engineering

Conclusion

- **High-Performance Tasks**: Use **mBART Fine-Tuning** for the best translation accuracy if resources are available.
- **Balanced Tasks**: Use **mBART Layer Freezing** for good performance with moderate resource usage.
- **Low-Resource Scenarios**: Use **mBART LoRA** for efficiency or **Marian MT** for resource-constrained Chinese-to-English translations.
- **Avoid**: General-purpose large pre-quantized models (e.g., LLaMA) for machine translation due to poor performance.

Thank You!

From Bart to Edge

Contributors: Xuanyou Liu, Prekshi Vyas, Raksha Ramesh, Manurag Khullar

Penn Engineering
University of Pennsylvania