# Project Proposal: Implementing GPT-2 with 4-bit Quantisation for Efficient Multilingual Natural Language Inference

**Le Thi Thuy Hang (1010883); Do Viet Anh (1007143); Pham Hong Quan (1007131)**

*Group 23*

## Abstract

This project implements GPT-2 from scratch, trains it on a small dataset, and then fine-tunes it to understand text relationships in English and 15 other languages. We also compress the model using 4-bit quantisation to make it smaller and faster. Our goal is to reduce model size by 75% while keeping performance high (losing less than 1% accuracy) and achieving 1.5–2× faster inference. This work shows how to deploy large language models on devices with limited resources.

## 1 Introduction

Large language models like GPT-2 (Radford et al., 2019) are powerful but difficult to use because they require substantial computer memory and processing power. This project addresses these challenges by teaching us how to build GPT-2 from first principles and then make it work efficiently on devices with limited computational resources through compression techniques. The work consists of four main components: building GPT-2, teaching it to understand text relationships in English, extending it to work in 15 languages, and compressing it using quantisation. We plan to complete this work from November 3 to December 7, spanning five weeks of development and experimentation.

## 2 Project Tasks and Timeline

### 2.1 Task 1: Building GPT-2 and Initial Training (Weeks 9–10)

We will build all the necessary components of GPT-2 from scratch, following the architecture described by Radford et al. (2019). This includes constructing the attention mechanism with causal masking, which ensures that the model only looks at previous words when making predictions. We will also implement feed-forward layers, embedding layers that convert words into numerical representations, and the output layer. Each component will be tested against the official GPT-2 implementation from Hugging Face to ensure correctness.

Once the architecture is complete, we will train our GPT-2 on a small sample of text from OpenWebText to verify that the model learns effectively. By the end of Week 10, we expect to see the model's performance improving steadily and demonstrate that it can generate simple text in a coherent manner.

### 2.2 Task 2: Fine-tuning on English Text Relationships (Weeks 10–11)

Building on the pretrained GPT-2 model, we will fine-tune it to recognize relationships between pairs of sentences. The task involves understanding whether a premise and hypothesis have one of three relationships: Entailment (the first sentence implies the second), Contradiction (the sentences disagree with each other), or Neutral (the sentences are unrelated). For this task, we will use the XNLI dataset (Conneau et al., 2018) in English as our training and testing resource.

We will systematically test different training settings and approaches to identify which configuration produces the best results. By the end of Week 11, we expect the model to achieve accuracy above 33.3%, which represents the random baseline for a three-class classification task.

### 2.3 Task 3: Extending to 15 Languages (Weeks 11–12)

We will expand our work to include languages beyond English by testing how well the English model performs when applied directly to other languages without any additional training, a technique called zero-shot transfer. In parallel, we will analyze how efficiently each language is encoded by the model's tokenizer, measuring the average number of word pieces needed per word for each

language. This fertility analysis, inspired by cross-lingual transfer research, will help us understand which languages are naturally easier for the model to handle.

Following this analysis, we will take two complementary approaches: we will train separate models for each promising language, and we will also train a single model using data combined from all languages. We will then compare all these different approaches to understand which strategy works best. By the end of Week 12, we expect the newly trained models to perform noticeably better than the original English model when tested on text in other languages.

### 2.4 Extension: Compressing the Model with 4-bit Quantisation (Weeks 12–13)

To make our trained models practical for real-world deployment, we will apply a compression technique called 4-bit GPTQ quantisation (Frantar et al., 2023). This technique reduces the model size by 75% and cuts memory requirements by 4× while maintaining very high accuracy. The quantisation process works by reducing the precision of the model's numerical weights, which allows us to store the same model using much less space.

We will measure several important metrics for the compressed models, including the reduction in file size, the decrease in memory usage, the improvement in inference speed, and whether the compressed models still maintain their accuracy. By the end of Week 13, we expect to achieve more than 70% size reduction while experiencing less than 2% loss in accuracy.

## 3 Experimental Setup

Our experimental setup includes three main components: the data we will use, the computational hardware we need, and the specific settings we will apply during training.

### 3.1 Data Sources and Languages

For the initial training phase, we will use 5,000 lines of text from OpenWebText, a large publicly available corpus. For the main fine-tuning tasks, we will use the XNLI dataset (Conneau et al., 2018), which provides parallel datasets across 15 languages. These languages include English, French, Spanish, German, Chinese, Japanese, Korean, Russian, Arabic, Vietnamese, Thai, Turkish, Polish, Portuguese, and Greek. This multilingual coverage allows us to test how well our approach generalizes across different writing systems and linguistic families.

### 3.2 Computational Resources

We will use a personal computer with a CPU for code development and testing. For the actual model training, which is computationally intensive, we will use a GPU such as a T4 or A100. We estimate that the entire project will require between 10 and 20 hours of GPU computation time.

### 3.3 Training Configuration

For the initial GPT-2 building task, we will use a learning rate of $1\times10^{-3}$ and train for 3 complete cycles through the training data. The toy model for this phase will be small, with 128 hidden units, 2 layers, and 4 attention heads. For the fine-tuning tasks on text relationships and multilingual data, we will use a lower learning rate of $5\times10^{-5}$ and train for 1 cycle with early stopping to prevent overfitting. We will use a batch size of 4 and limit input text to a maximum of 128 tokens. For these tasks, we will use the full pretrained GPT-2 model with 768 hidden units, 12 layers, and 12 attention heads. For the quantisation phase, we will use 4-bit precision, a group size of 128, and enable the exllama optimization.

## 4 Evaluation Methodology and Deliverables

We will evaluate the project's success through multiple metrics tailored to each task and phase.

For Task 1, we will examine the trajectory of training loss over time to ensure it decreases as expected. We will also qualitatively assess whether the model can generate reasonable and coherent text samples.

For Task 2, we will measure the primary evaluation metric of accuracy on the held-out English test set. Additionally, we will conduct a sensitivity analysis to determine which training settings have the strongest impact on performance.

For Task 3, we will measure accuracy separately for each of the 15 languages. We will calculate fertility scores for each language and use statistical correlation methods such as Pearson or Spearman correlation to determine whether there is a relationship between how efficiently a language is encoded and how well the model performs on that language. We will also compare the performance

of per-language models against the single multilingual model trained on combined data.

For the quantisation extension, we will directly compare the compressed and original models across four dimensions: the reduction in model file size, the decrease in peak memory usage during inference, the improvement in inference speed measured in tokens per second, and the difference in accuracy between the compressed and uncompressed versions.

Our deliverables will follow this timeline: By November 9, we will submit the project proposal. By November 17, we will complete Task 1 with all components tested and verified. By November 24, we will finish Task 2 and report results on the English test set. By December 1, we will complete Task 3 with a full multilingual analysis. By December 7, we will submit the final deliverables, which will include all source code, trained model files, complete results tables, and a comprehensive written report.

## 5 Conclusion

This project provides a complete learning experience in implementing and deploying modern language models. By building GPT-2 from scratch, extending it to work effectively across multiple languages, and compressing it using state-of-the-art techniques, we will develop both theoretical understanding and practical skills directly applicable to professional natural language processing work. The quantisation extension ensures this project directly addresses real-world challenges that practitioners face when deploying models in production environments, making the work immediately relevant to modern AI engineering.

## References


Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *Proceedings of the 11th International Conference on Learning Representations*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.