Reconstructing GPT-2: Pretraining, Fine-Tuning, and Cross-Lingual Generalization in Transformer Models

Vancence Ho, Lee Ruiyu, Justin Cho
1007239, 1006993, 1007496
Singapore University of Technology and Design
{vancence_ho, ruiyu_lee, justin_cho}@mymail.sutd.edu.sg

Abstract

This project aims to reconstruct and extend the GPT-2 language model to gain a deeper understanding of large-scale transformer architectures, optimization dynamics, and multilingual transfer in natural language processing. After implementing the GPT-2 architecture and fine-tuning it for English Natural Language Inference (NLI), we extend the evaluation with **zero-shot machine translation** to non-English languages to analyze how translation quality correlates with linguistic distance and tokenizer efficiency. Furthermore, we replace GPT-2 with Qwen2.5, a modern multilingual model with Rotary Position Embeddings and SwiGLU activations, to investigate how architectural differences in models affect cross-lingual reasoning.

1 Introduction

This project involves three key tasks using GPT-2. First, we implemented the model from scratch—including its causal self-attention and the AdamW optimizer—and validated it through pretraining on English text. Next, we fine-tuned a pretrained GPT-2 from HuggingFace on English Natural Language Inference (NLI) and adapted it to classify relationships (entailment, neutral, contradiction) between premise-hypothesis pairs. Finally, we extended the English NLI task to 15 other languages, comparing the performance of zero-shot transfer, per-language fine-tuning, and a single model fine-tuned on all languages.

As for our **Extension** (more on it in **Section 5: Extending GPT-2**), building upon our implementation and multilingual experiments from Task 2 and 3, we aim to further explore GPT-2's capabilities and limitations across tasks and architectures.

For extension 1, we intend to explore the multilingual analysis from Task 3 to machine translation, examining whether the language-specific patterns observed in the NLI task generalizes to translation quality. Using prompt-based translation with the pretrained GPT-2 model, we evaluate zero-shot translation performance across linguistically diverse languages based on grammaticality, meaning preservation, and correspondence with previously observed NLI patterns.

The second extension replaces GPT-2 with Qwen 2.5, a multilingual transformer employing Rotary Position Embeddings (RoPE) and SwiGLU activations, optimized for cross-lingual reasoning through extensive multilingual pretraining. This comparison reveals how architectural choices (RoPE vs learned positional embeddings) and training data diversity impact zero-shot transfer efficiency, particularly for linguistically distant languages like Chinese. The analysis demonstrates how modern multilingual architectures leverage both structural innovations and data scale to achieve more consistent performance across language boundaries.

2 Implementing GPT-2

2.1 Phase 1: Tokenizer

We used GPT-2 BPE tokenization to break texts into subword units that balance vocabulary size with the ability to handle out-of-vocabulary words, providing a foundation for the subsequent model implementation and training pipeline.

2.2 Phase 2: Model

The `GPT2Model` class defines the overall model architecture. It integrates all submodules, including embeddings, multiple transformer layers, and output normalization. Please refer to the architecture diagram (Figure 1) in the Appendix.

2.3 Phase 3: Adam Optimizer

The `AdamW` class implements the Adam optimizer with decoupled weight decay to stabilize training and helps prevent overfitting.

2.4 Phase 4: Pretraining

We performed a small-scale pretraining to verify if our implementation can be successfully trained by using the training dataset (labeled as `pretrain.txt`) provided and the `TextDataset` class.

The training utilizes the following settings:

- **Text Size**: 4 (across 4 samples in parallel)
- **Epochs**: 3 (iterate over the dataset 3 times)
- **Learning Rate**: 1×10^{-3} (Adam parameter)
- **Weight Decay**: 1×10^{-4} (Adam parameter)

- **Bias Correction**: True (Adam parameter)

We start training and track running loss stats by monitoring. After training over two epochs, we achieved a result of: `Global Avg Train Loss: 2.8317`. A decreasing trend in the loss would indicate that our model, optimizer, and training loop are correctly implemented (refer to Figure 2 in Appendix).

3 English NLI with GPT-2

3.1 Phase 1: Model Loading & Text Generation

In Task 2, we will load a pretrained GPT-2 model from HuggingFace with official weights. After loading the model, we generated texts directly, then finetune it for downstream tasks in Task 3. This text generation uses greedy decoding in `generate_gpt2()`. When prompted with `"Singapore University of Technology and Design is"`, the pretrained model generates coherent continuation text, while our toy GPT-2 model produces nonsensical output. This demonstrates how large-scale pretraining enables fundamental language capabilities.

3.2 Phase 2: NLI Dataset

We use the XNLI dataset for English NLI, with three labels: entailment, contradiction, neutral. The evaluation function `evaluate_gpt2_xnli` generates predictions via `generate_gpt2()` and computes accuracy via `compute_accuracy()`.

3.3 Phase 3: Fine-Tuning GPT-2

Fine-tuning employs next-token prediction: only the final token (`label`) contributes to loss, with other tokens masked (-100). The training loop utilizes the following hyperparameters:

- **Batch Size**: 4 (process 4 samples in parallel)
- **Epochs**: 1 (iterate over the dataset 1 time)
- **Learning Rate**: 5×10^{-5} (Adam parameter)
- **Weight Decay**: 1×10^{-2} (Adam parameter)
- **Bias Correction**: True (Adam parameter)

And fine-tuned using next-token prediction with label masking. We achieved a `Global Average Loss: 0.6160` and a `Evaluation Accuracy: 79.52%`. (`best_model/model.pt`). We then load this saved model (`best_model/model.pt`) and evaluate its performance on the test set, giving us a result of `Evaluation Accuracy: 78.82%`.

3.4 Phase 4: Hyperparameters Experiment

We also conducted an experiment by adjusting two of the five hyperparameters, mainly the **learning rate** and **batch size**. In the first adjustment, we adjusted the **learning rate** higher (`lr=1e-4`), whereas in the second adjustment we adjusted the **learning rate** lower (`lr=1e-5`). In the final adjustment, we modified the batch size by increasing it (`BATCH_SIZE=8`). The results can be found in Table 2 in the Appendix. We concluded that smaller learning rates and larger batch sizes improve model performance. Higher learning rates caused accuracy to drop while gradual parameter updates with lower rates enhanced generalization and stability.

4 Multilingual NLI with GPT-2

4.1 Phase 1: Zero-shot Cross-Lingual Transfer analysis

We conducted zero-shot evaluation of our English-finetuned GPT-2 model from Task 2 across 14 non-english languages available in the XNLI to assesses GPT-2's cross-lingual capabilities and establish a baseline for multilingual performance. Latin-based languages (e.g. French, Spanish and German) demonstrated reasonable zero-shot transfer, whereas non-latin based languages (e.g. Arabic, Chinese, Thai) performed closer to baseline. (Refer to Table 3 in the Appendix)

4.2 Phase 2: Tokenizer Fertility Evaluation

Fertility is defined as the average number of subword tokens produced per word ($\frac{\text{Total Tokens}}{\text{Total Words}}$). We conducted fertility analysis as a proxy for tokenizer efficiency to asses GPT-2 BPE tokenizer support. We compared fertility scores across languages, using English as a reference baseline and concluded that languages with lower fertility (better tokenizer support) achieved higher accuracy. (Refer to Table 4 in the Appendix)

4.3 Phase 3: Language Selection Strategy

We selected 5-8 languages for multilingual fine-tuning based on fertility and zero-shot performance: **Tier 1 (High Priority):** Low fertility languages (< 2.5) and reasonable zero-shot performance ($> 40\%$): e.g. French, Spanish, German. **Tier 2 (Medium Priority):** Medium fertility languages (2.5–3.5) or moderate zero-shot performance (30–40%): e.g. Russian, Bulgarian, Turkish. **Tier 3 Languages (Low Priority/Excluded):** High fertility languages (> 3.5) and poor zero-shot performance ($< 30\%$) due to limited GPT-2 tokenization support.
Our selection balances computational constraints with linguistic diversity across different languages. After sorting through our list of languages and filtering them to fit the above selection criteria, we used these 7 languages for the remaining tasks: Spanish, French, German, Bulgarian, Arabic, Greek, and Hindi.

4.4 Phase 4: Per-language Fine-Tuning

We trained specialized models for each non-English language. We started from the base GPT-2 model and trained each model on the respective XNLI language datasets with the same hyperparameters. We tracked each model's performance during training and saved the best version, before testing it on that language's test set.

Referring to Table 5 in the Appendix, the average improvement in all 7 languages was 2.31%. The Latin-based languages had the best improvement scores, with French being the best scorer (7.24%). On the other hand, Non-Latin based languages did worse, with Bulgarian being the worst scorer (-3.81%).

4.5 Phase 5: All-Language Fine-Tuning

We then trained a single multilingual model on combined data from all selected languages, plus english. The procedure is roughly outlined as follows:

- Combine training data from English (XNLI-1.0) and selected non-english languages (XNLI-MT-1.0)
- Shuffle the combined dataset to ensure language mixing within batches
- Load pretrained GPT-2 and fine-tune based on the combined dataset using the same hyperparameters as per-language training
- Evaluate the single multilingual model separately on each language's test set

This approach tests whether multilingual training enables positive transfer across languages, potentially helping lower-resource or higher-fertility languages benefit from English and other language data. We will also assess whether multilingual training causes negative transfer, degrading performance on high-performing languages.

Referring to Table 6 in the Appendix, the average improvement in all 7 langugues was 0.00%. TO BE CONTINUED

4.6 Phase 6: Comparative Analysis

Across the below three experimental settings, we have come to the following conclusions:

1. **Zero-Shot Transfer:** English-finetuned model tested on other languages
2. **Per-Language Model:** Language-specific models trained independently
3. **All-Language Model:** Single multilingual model trained on combined data

Our analysis will investigate whether tokenizer fertility predicts model performance across languages, identify which languages benefit most from fine-tuning, and examine potential positive or negative transfer effects in the multilingual setting. We will also assess the practical trade-off between per-language models (higher accuracy) versus a single multilingual model (deployment efficiency). The results will be presented through comprehensive tables, grouped bar charts, and scatter plots examining correlations between fertility and accuracy metrics.

We anticipate that zero-shot transfer will succeed primarily for Latin-script languages with low fertility, while per-language fine-tuning will yield the highest individual accuracies at the cost of maintaining multiple models. The multilingual model may facilitate positive transfer to challenging languages while experiencing minor degradation on high-performing ones. As Task 3 is partially open-ended, any results providing insights into GPT-2's multilingual capabilities will be valuable nonetheless.

5 Extending GPT-2

In this section, we will describe the additional exploration and direction based on our implementation, experiments, and observations. Our team has explored two of the following extensions listed below:

- We extended the multilingual analysis from Task 3 to machine translation and investigated if the language-specific patterns observed in NLI (e.g. zero-shot transfer performance, fertility correlations) partially generalizes to other tasks. This revealed that grammar generation and fertility correlation patterns are task-agnostic, while semantic meaning preservation and translation fluency is task-specific.
- We also replaced GPT-2 with another pretrained multilingual LLM (Qwen) and evaluated them on the same NLI setup. We analyzed the differences in their cross-lingual performance and discussed how architectural or pretraining choices contributed to their multilingual abilities.

5.1 Extending GPT-2 to Machine Translation

We utilized the language-specific patterns that we have observed, extending the multilingual analysis done in Task 3, and generalized it to machine translation. Using the pretrained GPT-2 model, we explored the quality of its translation in languages that have vastly different language structures and grammar compared to English.

We created a new function to implement the translation task, with parameters as such: `model`, `src_lang`, and `src_text`. We will use the pretrained GPT-2 for the model parameter for the translation task to test for zero-shot translation capability. In this function, we will use prompt engineering to prompt GPT-2 with translation tasks. The exact prompt is selected from a set of simple prompts, similar to the following prompt: "`translate this src_lang to English: 'src_text'`, via a rudimentary automated selection function that tests the templates on 2 sample sentences per language. It then measures word overlap between GPT-2's output and reference translation, then chooses the prompt with the highest average overlap. Next, using the XNLIDataset from Task 3, we loaded the different languages for the translation task. We split the dataset into smaller subsets for testing with

`max_samples = 100`.

This model's output has a better basis for comparison to the zero-shot output from the multilingual NLI model in Task 3. This is because using the same base model would allow us to isolate whether translation capabilities come from pretraining alone versus fine-tuning.

Finally, we will evaluate the translation based on the following three metrics:

1. Grammaticality Score (rated from 1-5)
2. Meaning Preservation (rated from 1-5)
3. Evaluation against Task 3 patterns for the NLI task to see if the same languages also have difficulty in the translation task.

With this extension, we determined which tasks are task-agnostic and task-specific:

Task-Agnostic Capabilities:

1. Language Performance Ranking: The relative order of the languages remained consistent: Spanish and French performed best, but Arabic and Greek were the worst in NLI and translation. This suggests that GPT-2 has consistent language affinity that transcends specific task requirements.
2. Tokenization Efficiency Correlation: Fertility scores predict performance in both tasks (with similarly inverse correlation coefficients). This indicates that high-fertility languages struggle uniformly across different NLP applications.
3. Grammatical Structure Transfer: Languages that produce grammatical English in NLI contexts also produce grammatical translations ($r = 0.718$), showing that syntactic patterns learned from english pretraining transfer across comprehension and generation tasks.

Task-Specific Capabilities:

1. Semantic Understanding and Generation: NLI requires semantic comprehension, while translation requires semantic generation. We can conclude that high NLI accuracy does not guarantee good translation meaning preservation.
2. Cross-lingual Mapping Precision: GPT-2 recognizes cross-lingual relationships as evidenced by the NLI success, but translation requires precise lexical mapping that does not fully transfer. The model learns that the languages are related but fails to accurately convert between them.

Limitations and Assumptions

To keep things understandable and simple, we only implemented a crude function to measure the correctness of the translation's grammar and meaning. This function scores a translation by measuring how many words it shares with a reference sentence, converting that proportion into a value on a 1-5 scale to approximate how much meaning is preserved. This is useful as a baseline metric because it provides a coarse meaning check that identifies translations which omit essential content or fail to translate key terms. However, this choice of metric fails in capturing deeper semantic relationships, paraphrasing, or contextual meaning, since it relies solely on word overlap. As a result, it can underestimate good quality translations that use different wordings while overestimating poor translations which merely repeat reference vocabulary without preserving the actual meaning.

Additionally, our evaluation design might end up artificially inflating GPT-2's apparent capability by comparing against machine-translated references instead of native text. By using XNLI's English premises, which were themselves produced by machine translation from the original multilingual corpus, it creates a circular comparison where we evaluate GPT-2's translation quality against sentences that can contain the same translation artifacts and unnatural phrasing that characterizes machine output. As a result, a translation that merely mimics the patterns of machine-translation English could receive falsely high scores, while a more natural and fluent translation using different wording might be penalized. This flaw prevents us from differentiating whether GPT-2 is genuinely capturing meaning or simply learning to reproduce the specific patterns of the machine-translated training data it was evaluated against.

5.2 Internal Behaviour Analysis of GPT-2

Attention Patterns and Fertility Scores

Attention pattern visualization revealed systematic variation in attention entropy across languages. Spanish and German exhibited cleaner, more focused attention distributions (lower entropy, r=0.718 with grammar scores), while Greek showed more scattered patterns. This correlation suggests that diffused attention may impair grammatical coherence by failing to capture long-range syntactic dependencies. For example, when attention spreads across many tokens rather than focusing on syntactically relevant ones (e.g., subject-verb agreement), the model may lose track of grammatical constraints. However, with only three languages examined, this pattern warrants cautious interpretation.

Layer Representation Analysis and Performance Gradients

Layer-wise similarity probing showed that early-layer alignment with English correlates with NLI performance. Spanish and German maintained high

similarity (>0.94) from initial layers, corresponding to stronger NLI scores (0.4535, 0.3896), while Greek's lower early similarity (0.6959) aligned with weaker performance (0.3629). The moderate convergence observed (+0.0168 to +0.1014) suggests that languages requiring greater representational transformation may face processing challenges. Whether this reflects training data differences, typological distance, or genuine processing mechanisms remains unclear.

Performance Implications
Greek required the largest convergence gain (+0.1014) yet achieved the lowest external performance, while Spanish and German needed minimal convergence (+0.0168, +0.0080) and performed best. This inverse relationship between convergence requirements and task performance may partially explain the moderate cross-task correlation (r=0.620) observed. However, high convergence requirements could be either a cause of processing difficulty or a symptom of linguistic distance from English, as languages farther from English may both require more transformation and independently struggle with tasks due to training data scarcity or typological differences.

Connecting Internal Mechanisms to Observed Error Patterns
The internal behavior analysis provides potential mechanistic explanations for the specific error patterns observed in previous experiments:

1. **Grammatical Errors**: Languages with scattered attention patterns (high entropy) produced translations with weaker grammatical structure, suggesting that attention focus may play a role in maintaining grammatical coherence through better tracking of syntactic dependencies.

2. **Semantic Loss:** The representation gap in early layers for high-fertility languages correlates with poor meaning preservation scores ($\sim$1.2/5), suggesting that initial representational misalignment may limit semantic transfer capability.

3. **Performance Consistency:** The stable high similarity maintained by Spanish and German across layers corresponds to their consistent performance across both NLI and translation tasks, while Greek's transformation requirements associate with greater performance variability.

4. **Task Correlation Limits:** The potentially different internal processing requirements for NLI (representation alignment) versus translation (attention focus) may partially explain why correlation is moderate rather than strong between tasks.

5.3 Replacing GPT-2 with Qwen2.5-0.5B

This experiment aims to compare the performances of traditional and modern language models on the same NLI task. We chose Qwen2.5-0.5B pretrained model as the model trained on modern techniques to explore its cross-lingual performance as compared to GPT-2. We hypothesize that with Qwen2.5-0.5B's multilingual architectures and modern training methods, it will outperform GPT-2 on the same NLI task.

Qwen2.5-0.5B Architecture
Qwen2.5-0.5B is a decoder-only transformer model with 490 million parameters optimized for multilingual NLP tasks. It features a 32K-token context window. The base model utilizes Rotary Position Embedding (RoPE) for relative positional encoding, SwiGLU activation function to incorporate non-linearity expressivity and RMSNorm for stable normalization. The model utilizes Grouped Attention Query (GQA) with 14 attention heads for `Q` and 2 for `KV` as its attention mechanism.

Multilingual support
Compared to GPT-2's English-first model, Qwen2.5 was trained to be a multilingual model. RoPE allows Qwen2.5 to handle varied syntactic structures by applying a rotation to the word vector. (Azhar, 2024) Qwen2.5's attention mechanism, GQA,, allows efficient cross-lingual knowledge transfer through partitioning query heads which consists of flexible formulation of multi-query attention, into multiple groups. These groups share a set of keys and values. (Bergmann, Stryker, 2025) Lastly, the SwiGLU activation function in the feed-forward network introduces gradient preservation and non-linearity. This allows the model to learn crucial relationships between each token while preventing abruptness of ReLU at zero, allowing better gradient flow during training. (Lu, 2024)

5.4 Fine-Tuning Qwen2.5-0.5B

Experimental Setup
This experiment used PyTorch, HuggingFace Transformers and Parameter-Efficient Fine-Tuning (PEFT) ran on A100 GPU. The same XNLI dataset used to fine-tune and tested on GPT-2 is also used in this experiment for fairness. The Qwen2.5 model is trained on the full English training dataset and tested on the full test dataset for 4 other languages namely Thai, Arabic, German and Chinese.

LoRA Configuration for Qwen2.5
Following modern techniques to train a Large Language Model (LLM) with a large number of parameters, this experiment implemented Low-Rank Adaption (LoRA) to freeze the original 490M pretrained weights in Qwen2.5-0.5B and perform low-rank weight updates for efficient training. This shortened the model training to 1.5 hours instead of the initial 4 hours. Below are the parameters for the LoRA configuration:

```
lora_config = LoraConfig(
task_type=TaskType.CAUSAL_LM,
```

```
r=8,
lora_alpha=32,
lora_dropout=0.1,
target_modules=["q_proj", "v_proj",
"k_proj", "o_proj"],
bias="none")
```

Optimal LoRA parameters depend on the specific task and dataset size. (Yan, et al, 2025) The parameters selected for this experiment are selected to balance capacity and parametrization for a medium-sized dataset and model. Studies have shown that higher ranks `r` have diminishing returns. Ranks of 8 and above have been shown to match the performance of full fine-tuning on datasets with tens of thousands of text (Yan, et al, 2025), which is fitting for the English XNLI dataset size of 39K. Common practice also indicates that the scaling factor (determined by `lora_alpha / r`) is more critical than the individual values (Hu, et al, 2021). Therefore, a value of `lora_alpha=32` was chosen, creating a scaling factor of 4. This provides a significant but stable weight to new task-specific adaptations. To prevent overfitting, a `lora_dropout=0.1` was applied. Following the approach in the original LoRA paper, which suggests targeting self-attention modules for adaptation in transformer models, the `target_modules` were set to the attention projections (`q_proj`, `k_proj`, `v_proj`, `o_proj`) (Hu, et al., 2021). A higher learning rate of `2e-4` was used, as LoRA typically requires faster adaptation from its small, initialized parameter matrices compared to full fine-tuning.

Training Optimization Techniques

This experiment applied the same training method for fine-tuning pretrained GPT-2 in Task 2 to LoRA only parameters. This experiments utilized gradient accumulation with `gradient_accumulation_steps = 4` and `effective_batch_size = 32`, mixed precision training optimized to the A100 GPU, gradient scaling and AdamW optimizer. Below is the training configuration:

```
EPOCHS = 1
BATCH_SIZE = 32
LR = 2e-4 # Learning Rate
WEIGHT_DECAY = 0.01
CORRECT_BIAS = True
```

Other than the modified batch size for training efficiency and modified learning rate to match the model size, the rest of the configurations were kept the same as when GPT-2 was being finetuned.

5.5 Evaluation Strategy

This experiment implemented a unified evaluation framework for both GPT-2 and Qwen2.5 `evaluate_model_nli` for the NLI task. The 2 criteria are **accuracy** in correct predictions and **zero-shot cross-lingual** on other languages. The rationale for the 4 non-English languages chosen is to evaluate how both models perform on languages that does and does not have a similar structure to English.

5.6 Experiment Results

Performance Comparison

Language	GPT-2	Qwen2.5-0.5B	Improvement
English (en)	78.82%	84.21%	+5.39%
Thai (th)	34.71%	63.39%	+28.68%
Arabic (ar)	35.89%	67.66%	+31.77%
German (de)	40.76%	71.14%	+30.38%
Chinese (zhi)	37.6%	71.5%	+33.9%

Table 1: Model Accuracy Comparison on NLI Task

5.7 Conclusion

As hypothesize, Qwen2.5 outperforms GPT-2 by an average of 26%. It showcased consistent performance across linguistic families as there were minimal performance drop from English to other languages. Qwen2.5 can effectively transfer language-agnostic representations due to its model architecture.

Qwen2.5's architectural choices are specifically optimized for multilingual processing compared to GPT-2's English-focused designs. RoPE enable effective handling of diverse syntax across languages, overcoming limitations of absolute positional encoding applied in GPT-2. Next, the GQA mechanism provides a balanced approach that shares semantic knowledge across languages while allowing language-specific processing. Third, SwiGLU activations captured fine-grained morphological patterns for understanding complex languages. Lastly, LoRA enables task specialization while preserving multilingual knowledge, demonstrating that cross-lingual capabilities reside in low-rank parameter subspaces. These technical innovations explain why Qwen2.5-0.5B, despite having fewer parameters than GPT-2, achieves dramatically better performance across diverse languages. It represents a fundamentally different approach to multilingual language modeling that prioritizes architectural efficiency and cross-lingual compatibility over simple parameter scaling.

Acknowledgments

This project proposal was developed with reference to the document *default_project.pdf*, which was provided as part of the course materials. The structure and task descriptions outlined in the document served as the foundations for this work. Additionally, this proposal was prepared and written using the official ACL LaTeXtemplate distributed by the *Association for Computational Linguistics*.

Code Availability

The source code supporting this report is available at: https://github.com/vancenceho/nlp-scratch-gpt2

Appendix

Contributions

1. Vancence Ho: Completed task 1 and 2, and wrote the abstract, Section 1, 2, 3 of the report.

2. Justin Cho: Completed task 3 and the extended task (Machine Translation), and wrote Section 4, 5.1, and 5.2 of the report.

3. Lee Rui Yu: Completed the extended task (Replacing GPT-2 with Qwen2.5-0.5B), and wrote Section 5.3, 5.4, 5.5, 5.6, 5.7 of the report.

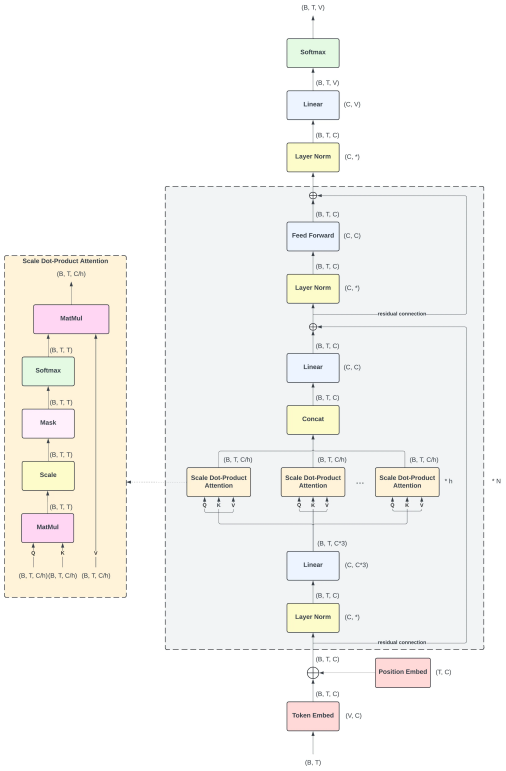

Figure 1: Overview of the GPT-2 Architecture, adapted from Wu (2024)

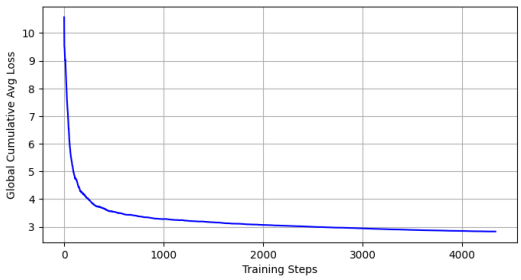

Figure 2: Global Cumulative Average Loss

Experiment	Dev Acc	Test Acc	Train Loss
Baseline	79.52%	78.82%	0.6160
LR ↑ (1e-4)	75.46%	74.01%	0.6930
LR ↓ (1e-5)	81.16%	81.08%	0.5881
Batch Size (8)	80.92%	80.20%	0.5952

Table 2: Hyperparameter Experiments Comparison

References

Azhar. Rotary positional embeddings: A detailed look and comprehensive understanding.

Dave Bergmann and Cole Stryker. What is grouped query attention (gqa).

Lu. Beyond relu: Discovering the power of swiglu.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Henry Wu. 2024. Gpt-2 detailed model architecture. *Medium*. Online; accessed 2025-11-09.

Minghao Yan, Zhuang Wang, Zhen Jia, Shivaram Venkataram, and Yida Wang. 2025. Plora: Efficient lora hyperparameter tuning for large models. *arXiv:2508.02932v1*.

Experiment	Dev	Test
Higher LR (1e-4)	-4.06%	-4.81%
Lower LR (1e-5)	+1.64%	+2.26%
Larger Batch Size (8)	+1.40%	+1.38%

Table 3: Improvements Over Baseline

Language	Zero-shot Accuracy	Zero-shot Performance
en	78.82%	Reasonable
de	40.76%	Reasonable
es	46.17%	Reasonable
fr	42.18%	Reasonable
ar	35.89%	Moderate
bg	37.23%	Moderate
el	35.09%	Moderate
hi	34.77%	Moderate
ru	37.64%	Moderate
sw	36.51%	Moderate
th	34.71%	Moderate
tr	38.00%	Moderate
ur	34.89%	Moderate
vi	36.97%	Moderate
zh	37.62%	Moderate

Table 4: Zero-shot cross-lingual accuracy per language

Performance categories:

- **Reasonable:** Accuracy $\geq 40\%$
- **Moderate:** Accuracy 30% - 40%
- **Poor:** Accuracy $< 30\%$ (not present in this data)

Language	Fertility	Fertility Group
English	1.11	Low
Spanish	1.83	Low
French	1.75	Low
German	2.10	Medium
Swahili	2.08	Medium
Turkish	2.73	Medium
Vietnamese	3.62	Medium
Chinese	3.77	Medium
Arabic	4.70	High
Hindi	5.12	High
Urdu	5.16	High
Bulgarian	5.53	High
Russian	5.90	High
Greek	6.16	High
Thai	9.48	High

Table 5: Fertility (Tokens per Word) by Language

Language	Zero-Shot	Dev Acc	Test Acc	Improvement
de	40.76%	43.69%	45.11%	+4.35%
es	46.17%	51.29%	53.11%	+6.95%
fr	42.18%	49.20%	49.40%	+7.23%
ar	35.89%	38.92%	38.38%	+2.50%
bg	37.23%	33.61%	33.41%	-3.81%
el	35.09%	35.22%	35.43%	+0.34%
hi	34.77%	33.17%	33.37%	-1.40%

Table 6: Per-language fine-tuning results summary

Language	Fertility	Early (0-3)	Late (8-11)	Gain	Interpretation
Spanish	Low	0.9486	0.9654	+0.0168	Moderate convergence → Moderate NLI (0.4535)
German	Medium	0.9452	0.9532	+0.0080	Moderate convergence → Moderate NLI (0.3896)
Greek	High	0.6959	0.7973	+0.1014	Strong convergence → Better NLI (0.3629)

Table 7: Layer similarity convergence and NLI performance by language fertility