COFFIN: Causal Ordering Framework for Fantasy-Inspired Text Modeling

Justin Mascarenhas
Department of Computer Engineering
Rochester Institute of Technology
jm8087@g.rit.edu

***Abstract*—This paper presents COFFIN (Causal Ordering Framework for Fantasy-Inspired Text Modeling), a specialized implementation of GPT-2 pretrained on fantasy literature to generate contextually appropriate, narrative and meaningful text. This study discusses the methodology behind transformer-based language models, details the experimental setup using a modified GPT-2 architecture, analyzes training results with perplexity metrics, and examines practical applications in creative writing assistance. Studies demonstrate that even a reduced-parameter model can achieve meaningful text generation capabilities when trained on domain-specific content, though with limitations in semantic coherence over extended passages. The paper concludes on model efficiency trade-offs and propose future work toward a multi-platform personalized AI assistant.**

***Index Terms*—natural language processing, transformer models, language generation, GPT-2, fantasy literature, fine-tuning, perplexity**

I. Introduction

Natural Language Processing (NLP) has witnessed remarkable advancements through transformer-based architectures, enabling unprecedented capabilities in text generation and understanding. Among these, autoregressive models like GPT-2 have set benchmarks in generating fluent, contextually coherent sequences of text. COFFIN, or Causal Ordering Framework for Fantasy-Inspired Text Modeling, represents a focused implementation of these technologies, aimed specifically at fantasy text generation grounded in the Harry Potter universe.

COFFIN's primary goal is to develop a language model capable of producing contextually relevant, stylistically consistent narrative content that emulates the distinctive characteristics of the Harry Potter book series. This involves capturing intricate linguistic nuances, character-specific dialogues, and the broader magical tone that defines the series.

The practical applications of such a system extend to various creative domains, including:

- Assistive tools in libraries to summarize book series for readers.
- Interactive fictional dynamic story generation.
- Automated fan-fiction generation aligned with canonical style and tone.
- Creative writing assistance tools for fantasy authors.

COFFIN is built on the foundation of the GPT-2 architecture, a transformer-based decoder-only language model known for its effectiveness in generating coherent and semantically rich text. By training this architecture from scratch on a carefully selected corpus of Harry Potter books, COFFIN aims to internalize the unique linguistic patterns, thematic elements, and narrative conventions characteristic of the genre. This allows it not only to mimic the original style but also to extend it in plausible and engaging ways.

II. Methodology

A. Transformer Architecture

The transformer architecture, introduced by Vaswani et al. [1], has revolutionized natural language processing by addressing the limitations of recurrent neural networks through parallel processing and attention mechanisms. Transformers employ self-attention to weigh the importance of different words in a sequence when predicting the next word, allowing the model to capture long-range dependencies more effectively than previous architectures.

The key innovation of transformers lies in their attention mechanism, which computes relationships between all words in a sequence simultaneously rather than sequentially. This parallel computation significantly reduces training time while improving the model's ability to understand context across longer text spans. The multi-head attention mechanism further enhances this capability by allowing the model to focus on different representation subspaces.

B. Selection of Tokenizer

COFFIN employs the tiktoken [2] tokenizer, a highly efficient Byte Pair Encoding (BPE) [3] implementation optimized for OpenAI models. This tokenizer offers fast subword tokenization and supports special tokens required for autoregressive language modeling, such as `<|endoftext|>`. The tokenizer was used to tokenize the text input to include unique character names and magical terminology. Punctuation marks were omitted to simplify the model's learning process. Its compact vocabulary and subword segmentation ensure efficient encoding while preserving linguistic patterns essential for narrative modeling.

C. Selection of GPT-2

For COFFIN, GPT-2 [4] was selected as the base architecture due to several compelling factors related to computational efficiency and practical implementation. First, in terms of computational complexity, while larger models like GPT-3

offer improved performance, they demand extensive computational resources that exceed the scope of most academic environments. GPT-2, particularly in its reduced-parameter configurations, strikes a balance between capability and resource efficiency. Second, the training time for GPT-2 is significantly lower compared to larger architectures, making it feasible to train or fine-tune within limited project timelines. Third, although it has been surpassed in commercial applications, GPT-2 remains relevant in research and education, as its architecture underpins many modern transformer-based models, providing a solid foundation for understanding text generation. Finally, its open-source availability [5] and strong community support make GPT-2 an ideal choice for academic research, enabling unrestricted experimentation and model customization. Despite the emergence of newer models like GPT-3 and GPT-4, GPT-2 continues to be a valuable tool due to its accessibility, adaptability, and manageable resource requirements.

D. Dataset Selection

The Harry Potter series [6], comprising all seven books and totaling approximately 6.5 million characters, was selected as the primary training corpus for several methodological reasons. First, the series features a consistent authorial voice across all volumes, allowing the model to learn from a stable and coherent linguistic pattern. Second, the narrative is rich in fantasy elements, including diverse concepts, character interactions, and structured storylines that reflect the broader characteristics of the genre. Finally, the language used in the series offers an appropriate level of complexity—accessible yet enriched with sophisticated narrative techniques—making it an ideal dataset for training a language model in a constrained academic setting.

E. Evaluation Metric

Perplexity was chosen as the primary evaluation metric for COFFIN's performance. Perplexity, calculated as the exponentiated average negative log-likelihood of a sequence, effectively measures how uncertain a model is when predicting test data. This metric is also inspired by the original GPT-2 implementation [4]. Lower perplexity indicates better prediction capacity. The formula for perplexity is given by:

$$\text{Perplexity} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log p(w_i \mid w_1, w_2, \ldots, w_{i-1})\right) \tag{1}$$

This formulation directly evaluates the language model's ability to predict the next token in a sequence based on its context, aligning with the autoregressive nature of GPT-2.

Perplexity offers several advantages over alternative evaluation metrics, making it particularly suitable for assessing language models like GPT-2. Unlike metrics such as F1 score or precision, which rely on labeled data and are typically used for classification tasks, perplexity serves as an intrinsic evaluation metric that directly measures the model's ability to predict language. It is grounded in a probabilistic framework, capturing the model's uncertainty in predicting the next word in a sequence—an approach that aligns seamlessly with the autoregressive design of GPT-2. Furthermore, perplexity evaluates language modeling at the sequence level, rather than focusing on individual classification decisions, thereby offering a more holistic assessment of generative capability. Importantly, because perplexity correlates directly with the model's loss function, it provides a consistent and interpretable metric for tracking training progress and optimization. While external metrics such as BLEU or ROUGE could be used to compare generated text to reference samples, they are less appropriate in this context, as they fail to account for the creative and diverse nature of open-ended text generation where deviation from the source may be desirable.

III. Experimental Setup

A. Model Architecture

While the original GPT-2 architecture exists in several configurations ranging from 117M to 1.5B parameters [4], COFFIN utilizes a significantly reduced model to accommodate training resource constraints. Table I contrasts the architectural differences compared to the original small GPT-2.

TABLE I
Model architecture comparison between original GPT-2 Small and COFFIN implementation.

Parameter	**Original GPT-2 Small**	**COFFIN Implementation**
Layers	12	9
Hidden size	768	512
Attention heads	12	8
Vocabulary size	50,257	50,257
Max sequence length	1024	256
Total parameters	117M	54M

B. Hyperparameter Selection

The implementation employs carefully selected hyperparameters to balance training efficiency and model performance:

- Learning Rate: 1e-4
- Batch Size: 256
- Maximum Sequence Length: 64
- Early Stopping Patience: 5
- Optimizer: AdamW
- Learning Rate Scheduler: CosineAnnealingLR

The reduced model size offers several advantages:

1) **Faster Training Convergence:** Fewer parameters require less data to reach reasonable performance levels.
2) **Reduced Memory Footprint:** The model can be trained on consumer-grade hardware or standard cloud instances.
3) **Inference Speed:** Smaller models execute generation tasks more quickly, improving user experience.
4) **Educational Value:** The simplified architecture facilitates better understanding of transformer mechanics.

C. Computational Resources

COFFIN was trained using Google Colab's A100 GPU environment, which provided the following specifications:

- GPU: NVIDIA A100 (40GB VRAM)
- CPU: Intel Xeon (8 vCPUs)
- RAM: 16GB

The total training process required approximately 2.5 hours to complete 300 epochs, though early stopping was implemented it was never triggered.

D. Data Processing

The text corpus underwent several preprocessing steps prior to training to ensure compatibility with the model architecture and to enhance learning efficiency. First, Unicode normalization was applied to convert special characters into their ASCII equivalents, standardizing the text format. This was followed by character filtering, where all non-alphanumeric characters—except spaces—were removed to reduce noise. The cleaned text was then tokenized using TikToken, which encoded the corpus into token IDs based on GPT-2's vocabulary. Subsequently, the tokenized data was segmented into overlapping sequences to facilitate next-token prediction during training. After preprocessing, the dataset was divided into training and validation subsets, with 90% allocated for training and 10% for validation, enabling continuous performance evaluation throughout the training process.

IV. Results

A. Training Process

The model was trained using next-token prediction, the standard approach for autoregressive language models. Training proceeded with early stopping based on training loss to prevent overfitting but the model still trained for maximum setting of 300 epochs.

Fig. 1 shows the loss curve during training, demonstrating rapid initial improvement followed by gradual convergence. The optimization process exhibited characteristic diminishing returns, with the most significant improvements occurring in the first 20 epochs.

B. Perplexity Analysis

The trained model achieved a perplexity score of approximately 1.06 on the training set. This value represents a substantial improvement compared to the expected perplexity of approximately 50,257 for random word selection, which would be close to the vocabulary size. However, such a low perplexity on the training set strongly suggests that the model may be overfitting to the training data. This indicates that while the model excels at predicting words within the training corpus, its ability to generalize to new, unseen text is questionable. A perplexity of 1.06 implies that, on average, the model is highly confident in its predictions on the training data. While this confidence reflects the model's ability to capture patterns within the training set, it also raises concerns about its capacity to handle the variability of language in a broader context.

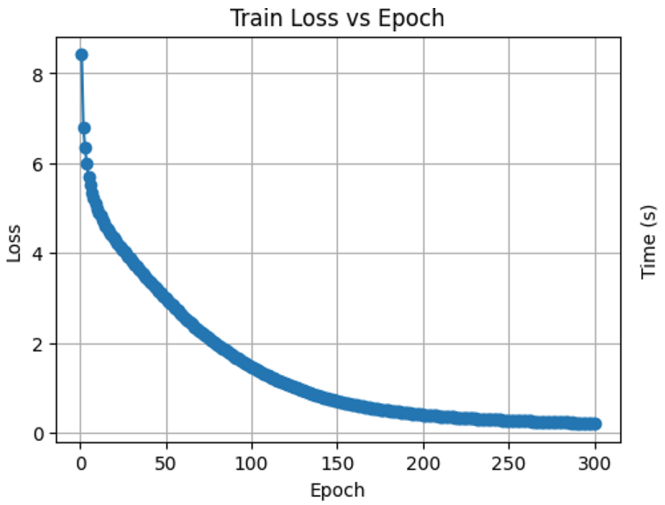

Fig. 1. Training Loss Over Epochs

It is important to note that while state-of-the-art models typically achieve perplexity scores around 10 on general text generation tasks, a direct comparison with COFFIN is inappropriate. COFFIN was trained on a much smaller, domain-specific dataset, which inherently limits the complexity and diversity of the language it encounters. This limited dataset likely contributes to the observed overfitting and the unusually low perplexity.

The perplexity value of 1.06 indicates several things:

1) The model has learned meaningful patterns in the Harry Potter text, capturing stylistic elements and narrative conventions.
2) Perplexity provides an estimate of the model's confidence while predicting the next token.
3) Despite the low score, there remains considerable uncertainty in predictions, particularly for rare words or unusual sentence constructions.
4) The perplexity value offers insights into the balance between memorization and generalization achieved by the model. In this case, the low perplexity suggests a tendency towards memorization.

We evaluated perplexity on the training set rather than standard benchmarks like WikiText [7] or the C4 corpus [8] for the following reasons:

1) Domain Relevance: WikiText and C4 contain text that differs significantly from fantasy narratives, making them inappropriate for evaluating domain-specific learning.
2) Style Consistency: The training set and validation set maintain the same authorial voice, providing a more accurate assessment of the model's ability to capture the target style.
3) Vocabulary Coverage: Domain-specific terms present in fantasy literature may be absent or used differently in general-purpose datasets.

C. Qualitative Assessment

To assess the performance of COFFIN beyond numerical scores. The model's generative quality was also evaluated across different training stages using domain-relevant prompts from the Harry Potter corpus.

`Prompt: professor mcgonagall sniffed angrily`

`Generated text: professor mcgonagall sniffed angrily antigen head head head head head head head head head head Show Show Show speech My My My history history history history history history history history history history history history`

In this untrained state, the model lacks semantic understanding and exhibits repetitive, meaningless token loops, indicating that no coherent language structure has yet been learned.

`Prompt: lawns to hagrids hut on the edge of the forbidden forest it was only when he spotted three onlytoofamiliar backs`

`Generated text: lawns to hagrids hut on the edge of the forbidden forest it was only when he spotted three onlytoofamiliar backs ahead large crowd when he reached hagrid having received a clanging branches of a complete groups at`

Here, the model captures the original style and structure well. While it diverges from the exact reference, it demonstrates contextual awareness and plausible narrative extension.

`Prompt: firebolt in their midst performed their best moves faultlessly and by the time they hit the ground again wood didnt have`

`Generated text: firebolt in their midst performed their best moves faultlessly and by the time they hit the ground again wood didnt have a back in the team inspired by the team the crate by the time they hit the ground again angel`

The model demonstrates strong local coherence and stylistic consistency, but begins to lose semantic focus in longer continuations — repeating fragments and introducing disconnected phrases. This reflects typical limitations of smaller transformer models, which handle short dependencies well but may struggle with long-range structure.

D. Limitations and Challenges

Several limitations were observed during model evaluation:

1) **Repetition:** The model occasionally produced repetitive phrases when new token count was increased to a value beyond twenty.
2) **Limited World Knowledge:** Without the extensive pretraining of larger models, COFFIN shows limited knowledge beyond its training corpus.
3) **Context Window Constraints:** The reduced sequence length (256 tokens) limited the model's ability to reference information from earlier in a text.

These limitations highlight the inherent trade-offs when optimizing for computational efficiency over absolute performance.

V. Conclusion

This paper presented COFFIN, a modified GPT-2 model designed for domain-specific text generation, trained on the Harry Potter dataset. The successful implementation of COFFIN, utilizing a reduced-parameter architecture (approximately half the size of the original small GPT-2) , demonstrates the feasibility of creating efficient language models tailored for specific domains, even with limited computational resources. The model achieved a low training perplexity of 1.06, indicating its capacity to capture statistical patterns and stylistic features within the training corpus. However, qualitative analysis revealed limitations, including the tendency to generate repetitive phrases and exhibit semantic drift over longer sequences. These findings underscore the inherent trade-off between model efficiency and the challenge of generating coherent, contextually rich, and novel narratives.

Despite these limitations, this project offers valuable insights into the scalability and adaptability of transformer architectures, the complexities of hyperparameter optimization in resource-constrained environments, and the nuances of domain-specific fine-tuning. The practical experience gained throughout the model development lifecycle, from data preprocessing to evaluation, provides a deeper understanding of the challenges and opportunities in applying transformer models to specialized text generation tasks. Future work should prioritize addressing the observed overfitting, potentially through regularization or data augmentation techniques, and exploring methods to enhance the generation of more diverse and coherent text.

VI. Future Work

COFFIN: Causal Ordering Framework for Fantasy-Inspired Text Modeling, represents the initial phase of a more ambitious vision, developing a personalized AI assistant capable of operating across multiple platforms and domains. Future development directions include:

1) **Expanded Training Corpus:** Incorporating diverse literature to broaden range and reduce overfitting to a single author's style.
2) **Hybrid Architecture:** Exploring mixture-of-experts or retrieval-augmented generation approaches to enhance factual accuracy while maintaining computational efficiency.
3) **Cross-Platform Integration:** Developing APIs and lightweight deployment options for embedding the assistant in various applications and devices.

These enhancements would transform COFFIN from an experimental implementation into a practical tool for creative professionals and hobbyists, demonstrating the potential of

specialized language models to serve specific domains effectively.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017.

[2] OpenAI, "tiktoken: A fast BPE tokenizer," GitHub, 2022. [Online]. Available: https://github.com/openai/tiktoken [Accessed: May 2, 2025].

[3] Wikipedia contributors, "Byte pair encoding," *Wikipedia*, [Online]. Available: https://en.wikipedia.org/wiki/Byte$_pair_encoding[Accessed$: $May2, 2025]$.

[4] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," OpenAI, Tech. Rep., 2019.

[5] OpenAI, "GPT-2 GitHub Repository," 2019. [Online]. Available: https://github.com/openai/gpt-2 [Accessed: May 2, 2025].

[6] S. Maindola, "Harry Potter Books Dataset," Kaggle, 2022. [Online]. Available: https://www.kaggle.com/datasets/shubhammaindola/harry-potter-books [Accessed: May 2, 2025].

[7] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," *arXiv preprint* arXiv:1609.07843, 2016.

[8] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[9] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, et al., "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[10] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, et al., "Scaling laws for neural language models," *arXiv preprint* arXiv:2001.08361, 2020.