Building a Minimal Language Model on Wikipedia: Approach, Choices, and Results

The goal was to train a small yet functional language model on Wikipedia text that could generate coherent responses. Instead of building a model from scratch, I fine-tuned existing transformer models to keep training efficient while ensuring good performance. I chose FLAN-T5 and BART, two well-optimized models for text generation. After that, I created an ensemble model to combine their strengths. The objective was to create a functional and efficient model while maintaining a balance between performance and resource constraints.

1 Dataset

I picked Wikipedia as the dataset since it's diverse, well-structured, and contains reliable information. I downloaded a subset of Wikipedia dumps, keeping only the relevant text fields. Then, I cleaned the data by removing special characters, citations, and unnecessary formatting. The goal was to make sure the text was readable and free of noise. After that, I structured it in a Q&A format, using article titles as questions and the corresponding text as answers.

To optimize processing, I split the text into chunks and tokenized it in batches. This made training more memory-efficient. The dataset was then split into training and evaluation sets.

2 Model Selection

I fine-tuned two models: FLAN-T5 (Base) and BART (Base) due to following reason :

- **FLAN-T5** is trained to follow instructions, making it a great choice for question-answering tasks. It understands queries well and generates structured responses. The model is based on the T5 architecture but fine-tuned with instruction-based learning, making it particularly useful for tasks that require structured, coherent outputs.
- **BART** is a sequence-to-sequence model known for strong text generation and summarization. It’s robust in handling noisy inputs and paraphrasing responses well. BART's architecture includes an encoder-decoder structure that is particularly effective for denoising tasks, meaning it can generate more fluent and natural text even with imperfect input.

I considered other models like GPT-2, T5-base, and DistilBERT, but they were not ideal for this task:

- **GPT-2**: GPT2 is one of the powerful models, but it is autoregressive and lacks an encoder-decoder structure, making it harder to fine-tune for structured Q&A tasks.
- **T5-Base**: This was a viable option, but FLAN-T5 is a fine-tuned version that performs better in instruction-following.
- **DistilBERT**: This model is lightweight and efficient but lacks the generative capabilities required for response generation.

BART and FLAN-T5 were the best balance between efficiency, fluency, and accuracy, making them ideal for this task.

3 Training and Optimization:

Since training a language model is resource-intensive, I optimized the process by

- Using **batch processing** to speed up tokenization and data loading.
- Adjusting **hyperparameters** like batch size, learning rate, and gradient accumulation to balance speed and stability.
- Enabling **mixed precision (bf16)** when possible to reduce memory usage.
- **Training on a reduced dataset** (100000 samples) to keep things manageable on my setup.

Each model was trained for three epochs while monitoring loss and validation performance. After training, I tested the models by generating responses to different queries.

4 Ensemble Modeling

After evaluating both models, I created an ensemble model. Instead of picking one model over the other, I let them work together.

The ensemble approach was designed to balance precision and fluency. Instead of averaging outputs at the sentence level, the ensemble evaluates responses word by word, selecting the most confident token from each model. The process works as follows:

1. Each model generates a response to the query.
2. The probability of each generated word is extracted from both models.
3. For each word position, the model with the highest confidence score determines the final word selection.
4. The sentence is then reconstructed from these selected tokens.

This method avoids excessive paraphrasing while leveraging the structural precision of FLAN-T5 and the fluency of BART. The tradeoff is that inference time is longer since multiple models must be queried before selecting a final answer. However, the final output retains the strengths of both models, producing responses that are coherent, well-structured, and natural-sounding.

5 Performance Evaluation:

To measure performance, I evaluated the models using:

- **ROUGE** (measures text overlap with references)
- **BLEU** (measures word-level precision for translation-like tasks)
- **METEOR** (accounts for synonym matching and meaning preservation)

Here are the key understandings from the model evaluation

- **FLAN-T5 performed the best**, especially in retaining information and structuring answers properly. It follows instructions well but sometimes lacks fluency in longer responses.
- **BART was good at fluency but weaker in precision**. It generated more readable responses but occasionally paraphrased too aggressively, losing some key details.
- **The ensemble model produced more balanced outputs**. While its scores were slightly lower, it benefited from combining the strengths of both models.

6 Possible Enhancements

One possible improvement is **incorporating a Retrieval-Augmented Generation (RAG) framework**.

- Instead of just relying on a pre-trained model, RAG retrieves relevant information from an external database before generating a response. This could improve factual consistency and reduce hallucinations.
- **Implementation:** Instead of relying solely on FLAN-T5 or BART, a RAG-based model would first query a retrieval system (like a vector database with FAISS or Elasticsearch) and then generate a response based on retrieved information.
- **Benefits:** This would make responses more up-to-date, context-aware, and accurate, addressing one of the key limitations of standard fine-tuned transformer models.

Conclusion

The project successfully trained a minimal-size model that generates Wikipedia-based responses. Instead of training from scratch, I fine-tuned two strong baseline models and combined them into an ensemble. This approach kept training efficient while improving response quality.

The evaluation showed that FLAN-T5 is best for precision, BART is better at fluency, and the ensemble balances both. If given more compute, I would explore larger datasets and fine-tuning on more task-specific instructions to improve the results further.

The goal was to keep things functional, efficient, and optimized for available resources. That mission was accomplished.

References

[1] Huang, Y., Feng, X., Li, B., Xiang, Y., Wang, H., Liu, T., & Qin, B. (2025). Ensemble Learning for Heterogeneous Large Language Models with Deep Parallel Collaboration. *Advances in Neural Information Processing Systems*, *37*, 119838-119860.

[2] Rehman, T., Ghosh, S., Das, K., Bhattacharjee, S., Sanyal, D. K., & Chattopadhyay, S. (2025). Evaluating LLMs and Pre-trained Models for Text Summarization Across Diverse Datasets. *arXiv preprint arXiv:2502.19339*.

[3] Marselino Andreas, V., Indra Winata, G., & Purwarianti, A. (2022). A comparative study on language models for task-oriented dialogue systems. *arXiv e-prints*, arXiv-2201.

[4] OpenAI. (2023). *ChatGPT* (Mar 14 version) [Large language model]. https://chat.openai.com/chat