POST 13 - GEN AI

FROM DATA TO DEPLOYMENT: THE JOURNEY OF BUILDING LARGE LANGUAGE MODELS

Generative AI Deep Dives

GENERATIVE AI FOR ALL

DINESH LAL
(DIRECTOR, DATA SCIENCE)

DINESHLAL

WHAT IS COVERED IN THIS DOCUMENT?

- This document is a step by step guide of how a LLM is designed from the problem defining stage to implementation stage
- There are 8 stages discussed in this document
- Each stage is covered in two pages,
  - Page 1 provides the summary of the stage, while
  - Page 2 elaborates with the help of an example

1. DEFINING THE PROBLEM STATEMENT

1. Problem Definition and Goal Setting:

- Define the objectives of the LLM project.
- Determine the specific tasks the model will be designed to perform, such as
  - language generation,
  - text classification,
  - translation, or
  - question answering.
- Set clear goals for performance metrics and outcomes to measure the success of the LLM.

1. DEFINING THE PROBLEM STATEMENT

- Imagine you're building an LLM to create realistic-sounding news summaries.
- You'd define the goal as generating concise summaries that accurately capture the main points of factual news articles.
- Additionally, you might set a performance target of achieving an F1 score (a combined measure of precision and recall) of over 85% on a benchmark dataset of news summaries.

EXAMPLE

2. COLLECTING THE DATA AND PREPROCESSING

2. Data Collection and Preprocessing:

- Gather a large and diverse dataset of text relevant to the chosen task.
- This dataset may include sources such as books, articles, websites, or domain-specific documents.
- Preprocess the data by cleaning, tokenizing, and formatting it appropriately for training.
- This step involves removing noise, normalizing text, and splitting it into sentences or paragraphs.

2.COLLECTING THE DATA AND PREPROCESSING

- To train your news summarization LLM, you'd gather a massive dataset of news articles from various reputable sources.
- This might include millions of articles from online news websites, newspapers (after obtaining permission), and news aggregators
- During preprocessing, you'd clean the data by removing irrelevant elements like advertisements, HTML tags, and special characters.
- You'd then tokenize the text, splitting articles into sentences and potentially even further into individual words or subwords.

EXAMPLE

3. Model Selection:

1. Choose a suitable pre-trained LLM architecture such as
   a. GPT (Generative Pre-trained Transformer),
   b. BERT (Bidirectional Encoder Representations from Transformers), or
   c. RoBERTa (Robustly Optimized BERT Approach).
2. Consider factors such as model size, computational resources, and task-specific requirements when selecting the architecture.

3. SELECTION THE RIGHT MODEL ARCHITECTURE

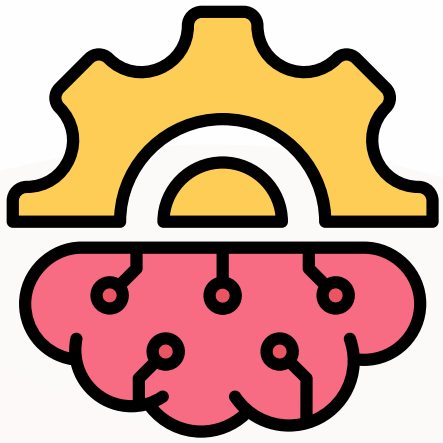
- Considering the task of news summarization, a good choice for the pre-trained LLM architecture could be **BART (Bidirectional and Autoregressive Transformer)** due to its demonstrated effectiveness in summarizing factual text.
- Other options might include T5 (Text-to-Text Transfer Transformer) or a fine-tuned version of BERT specifically trained on a massive dataset of news articles.

EXAMPLE

4. PRETRAINING

4. Pre-training:

- Initialize the chosen LLM architecture with random parameters.
- Pre-train the model on a large corpus of unlabeled text data using unsupervised learning techniques such as masked language modeling or next sentence prediction.
- Train the model to predict the next word in a sequence given the context of previous words, allowing it to learn contextual representations of language.



4. PRETRAINING

- Let's say you choose BART as the base architecture.
- You'd initialize the model with random weights and then pre-train it on a vast corpus of unlabeled text data, not limited to just news articles.
- This could include books, Wikipedia entries, and other general text sources.
- During pre-training, you might employ masked language modeling.
- Here, the model would be presented with sentences where some words are masked out, and it would have to predict the missing words based on the surrounding context.
- This helps the LLM learn general language patterns and relationships between words.e.

5. FINE TUNING

5. Fine-tuning:

1. Fine-tune the pre-trained LLM on specific downstream tasks using labeled data.
2. Customize the model's parameters and update its weights through supervised learning techniques to adapt it to the target task.
3. Fine-tuning may involve adjusting hyperparameters, selecting appropriate optimization algorithms, and experimenting with different learning rates.

5. FINE TUNING

- After pre-training, you'd fine-tune the BART model specifically for news summarization.
- This involves using a dataset of news articles where each article is paired with a corresponding human-written summary.
- The LLM would be presented with full news articles and would be tasked with generating summaries that closely resemble the provided human summaries.
- Through supervised learning techniques, the model's parameters are adjusted to excel at this specific task.

EXAMPLE

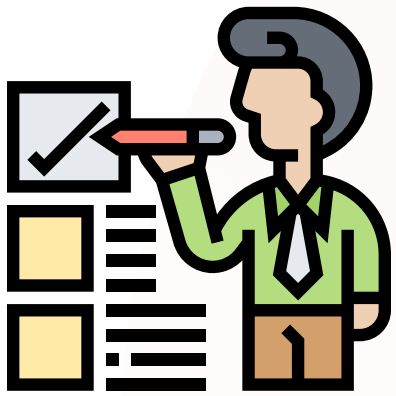
6. EVALUATING THE MODEL

6. Evaluation:

1. Evaluate the performance of the fine-tuned LLM on validation and test datasets. Measure the
    a. model's accuracy,
    b. perplexity,
    c. BLEU score,
    d. F1 score,
    e. or other relevant evaluation metrics depending on the task.
2. Iteratively refine the model by analyzing its performance and identifying areas for improvement.

6. EVALUATING THE MODEL

- To evaluate the fine-tuned model's performance, you wouldn't use the same dataset it was trained on.
- Instead, you'd use a separate validation dataset of unseen news articles.
- Metrics like ROUGE score (measures the similarity between generated summaries and human-written summaries) or BLEU score (focuses on n-gram overlap between generated summaries and references) would be used to assess how well the LLM captures the essential information from the articles.
- Based on these results, you might further fine-tune the model or adjust pre-training parameters for improvement.

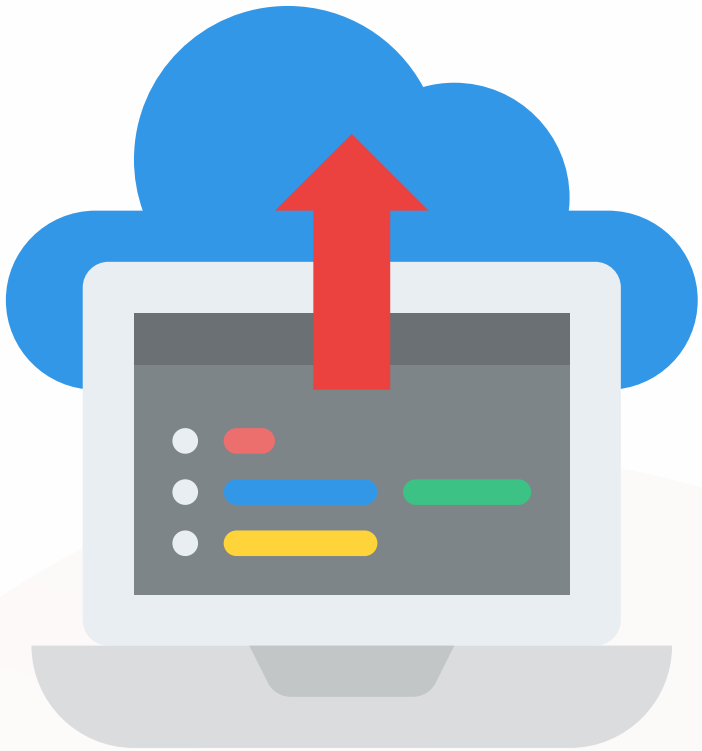


EXAMPLE

7.DEPLOYMENT FOR REAL WORLD APPLICATION

7.Deployment:

- Deploy the trained LLM in a production environment to perform real-world tasks.
- Integrate the model into existing systems or applications, ensuring compatibility and scalability.
- Monitor the model's performance in production and update it as necessary to maintain accuracy and reliability.



7. DEPLOYMENT FOR REAL WORLD APPLICATION

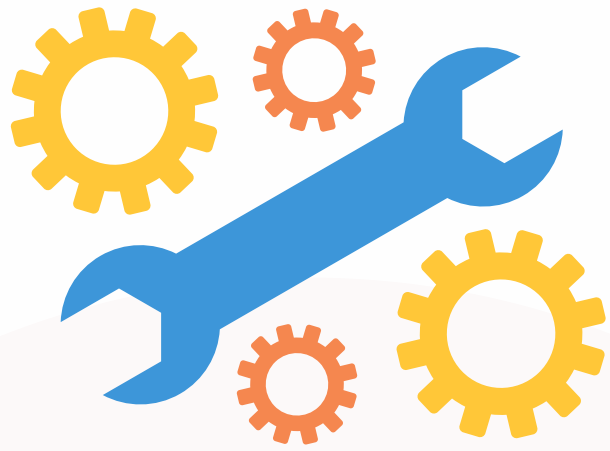
- Once satisfied with the LLM's performance, you'd deploy it in a real-world environment.
- This could involve integrating it into a news website or app where the LLM would automatically generate summaries for incoming news articles.
- The model would be optimized for efficiency to handle a high volume of incoming articles while maintaining accuracy.

EXAMPLE

8. MAINTENANCE OF THE LLM

8. Maintenance and Optimization:

1. Continuously monitor the performance of the deployed LLM and collect feedback from users.
2. Fine-tune the model further based on user interactions, changing data distributions, or emerging patterns in the data.
3. Optimize the model's architecture, parameters, and inference process to improve efficiency and reduce resource consumption.



8. MAINTENANCE OF THE LLM

- As the deployed LLM processes real-world news articles, you'd continuously monitor its performance.
- User feedback and analysis of generated summaries would help identify areas for improvement.
- The model might be further fine-tuned on new data reflecting evolving writing styles or emerging news topics.
- Additionally, you might explore techniques to optimize the LLM's inference process to reduce computational resources required for generating summaries.

EXAMPLE

Thank You

SPECIAL THANKS TO CHATGPT, OPEN AI, COPILOT FOR THE SUPPORT ON CONTENT

DINESHLAL