**Midterm Project:**

**"Building a Deep Learning Classifier for Identifying LLM Usage in Input Pairs"**

To implement this project, the goal is to build a deep learning classifier that can predict which Large Language Model (LLM) was used to complete a given text. Below is a step-by-step approach:

**1. Data Preparation**

We need pairs of input texts (x\_i) and their completions (x\_j) generated by different LLMs.

For example:

X\_i = "Yesterday I went"

X\_j\_1 = "to Costco." (LLM1)

X\_j\_2 = "to Walmart and bought some apple." (LLM2)

Then, for each pair (x\_i, x\_j), we label it with the corresponding LLM (e.g., LLM1, LLM2, etc.). I used multiple LLMs to generate completions for each truncated text (x\_i). The following is the lists of used LLMs:

|  |  |
| --- | --- |
| **Open-Source Platform** | **Model** |
| 1. Google | "gemini-1.5-flash" |
| 1. Google | PaLM |
| 1. Google | "bert-base-uncased" |
| 1. OpenAI API | "gpt-3.5-turbo" |
| 1. Meta | "llama3" |
| 1. Microsoft | "microsoft/phi-3" |
| 1. Amazon | Titan |
| 1. Apple | "apple/OpenELM-3B-Instruct" |
| 1. Bloom | "bigscience/bloom-1b3" |
| 1. Falcon | "tiiuae/falcon-40b" |
| 1. Cohere | "command-r-plus-08-2024" |
| 1. Mistral AI | "mistral-large-latest" |
| 1. AI21 Labs | Jamba |
| 1. CMU&Princeton | “state-spaces/mamba-2.8b” |
| 1. Alibaba | "qwen1.5-110b-chat" |
| 1. Shanghai AI Laboratory | " internlm/internlm2-chat-7b" |
| 1. DataBricks | " mosaicml/mpt-7b-instruct" |
| 1. BigCode | "bigcode/starcoder" |
| 1. Stability AI | "stabilityai/stablelm-2-12b-chat" |
| 1. Zhipu AI | "glm-4" |
| 1. WhyLabs |  |

Various LLMs services type such as:

1. Open-Source
2. APIs/AI Model
3. Developer Communities and Forums:GitHub, Stack Overflow, Kaldi Users Forum

**2. Feature Engineering**

For each pair (x\_i, x\_j), we consider the following:

* Semantic Similarity Compute the similarity between the truncated text (x\_i) and the completion (x\_j).
* Style Features Extract syntactic features like sentence structure, tone, word choice, and complex\_ity to differentiate between LLMs.
* N-gram analysis Analyze common phrases or words each LLM might use more frequently.

**3. Model Selection**

As mentioned, each input will be a tuple (x\_i, x\_j) representing the truncated text and its completion.

We used Transformer Models BERT to encode the text pairs.

**4. Training the Classifier**

We Define a supervised learning task where the input is the pair (x\_i, x\_j) and the label is the LLM that generated the completion.

For loss function, “cross-entropy loss” is used since this is a multi-class classification problem.

Regarding evaluation metrics,accuracy, precision, recall, and F1 score is used to evaluate the performance of the classifier.

**5. Post-processing and Analysis**

After training, we evaluate the model on a test set. Analyzing the features is important to see what kinds of completions each LLM tends to produce.

**6. Hyperparameter Tuning**

Different hyperparameters like learning rate, batch size, model architecture, and embedding dimensions to optimize performance were used.