# MAI 475– Large Language Model
## IV MSAIM
### 09-07-2024

# Regular lab Question

**Lab Exercise 5:**

**Implement and Analyze the Application of BERT and BERT-based Models in NLP Tasks: Sequence Classification, Token Classification, and Question Answering**

**Objective:**

To explore and implement BERT and its variants in three core NLP tasks—**sequence classification, token classification, and question answering**. The objective is to understand how BERT's architecture is adapted for each task and to evaluate its performance through practical implementation and experimentation.

**Task Descriptions:**

**i. Model Selection and Description**

Choose and describe one of the following BERT models for each task:

- **Base Model:** BERT-Base (uncased)

- **Variants (if required):**

    o **DistilBERT** – a lightweight version of BERT

    o **BioBERT** – domain-specific for biomedical text

    o **ALBERT** – A Lite BERT

    o **IndicBERT** – for Indian languages

    o **Multilingual BERT** – for cross-lingual tasks

For each task, provide:

- Architecture overview

- Pre-training objectives (MLM, NSP)

- Dataset used for fine-tuning

- Practical applications (e.g., sentiment analysis, NER, QA systems)

## ii. Implementation Tasks

You will implement BERT for the following three tasks using either HuggingFace Transformers or a PyTorch-based approach:

### a) Sequence Classification

- **Example Use-Case**: Sentiment analysis on English or multilingual reviews.
- **Steps**:
    - Load a pre-trained BERT model for classification (e.g., bert-base-uncased)
    - Fine-tune it on a labeled dataset (e.g., IMDb, SST-2, Indic movie reviews)
    - Evaluate classification metrics like accuracy, precision, recall

### b) Token Classification

- **Example Use-Case**: Named Entity Recognition (NER)
- **Steps**:
    - Use a token classification head on BERT
    - Use datasets like CoNLL-2003 or IndicNER
    - Output labeled tokens (e.g., Person, Location, Organization)

### c) Question Answering

- **Example Use-Case**: Extractive QA from a document paragraph
- **Steps**:
    - Use BERT for extractive QA (e.g., bert-large-uncased-whole-word-masking-finetuned-squad)
    - Pass a context paragraph and a question

- o Evaluate predicted start and end spans for the answer

### iii. Dataset and Preprocessing (Optional Based on Tasks)

- Use standard datasets or create your own:
  - o IMDb/SST2 for sentiment classification
  - o CoNLL-2003/IndicNER for token classification
  - o SQuAD or synthetic QA datasets for QA
- Preprocessing should include:
  - o Tokenization using BERT tokenizer
  - o Padding/truncation
  - o Label encoding (where applicable)
  - o Batch preparation

### iv. Evaluation and Results

Provide for each task:

- Input samples and predictions
- Model performance metrics:
  - o Sequence Classification: Accuracy, F1-Score
  - o Token Classification: Entity-level precision, recall
  - o QA: Exact Match (EM), F1
- Discuss strengths and limitations observed during experiments

### v. Evaluation Metrics

Use both automatic and manual evaluation methods:

- **Automatic**:
  - o Accuracy, F1, Precision, Recall
  - o Span-level EM/F1 for QA
- **Human Evaluation**:

- o Output interpretability

- o Clarity and correctness of predictions

- o Relevance to context

NB: No marks will be credited for the pipeline implementation.

## Program Evaluation Rubrics

| Model Selection and Implementation | 6 Marks |
|---|---|
| Timely Submission | 2 Marks |
| Viva | 2 Marks |

## General Instructions

- The file you have to save with your name, last 3 digits of register number and program number "Anto_501_Lab1".
- The implemented code you have to download and upload in the Google Class room in the given scheduled time.