**CÔNG TY CỔ PHẦN VCCORP**

FIRST WEEK REPORT

LEADER: MR. NGÔ VĂN VĨ

---

# Research about Pre-training (BERT, ELMO, ...)

---

***Lê Văn Hậu***

# Contents

# First Week Report

FireFly

June 12, 2025

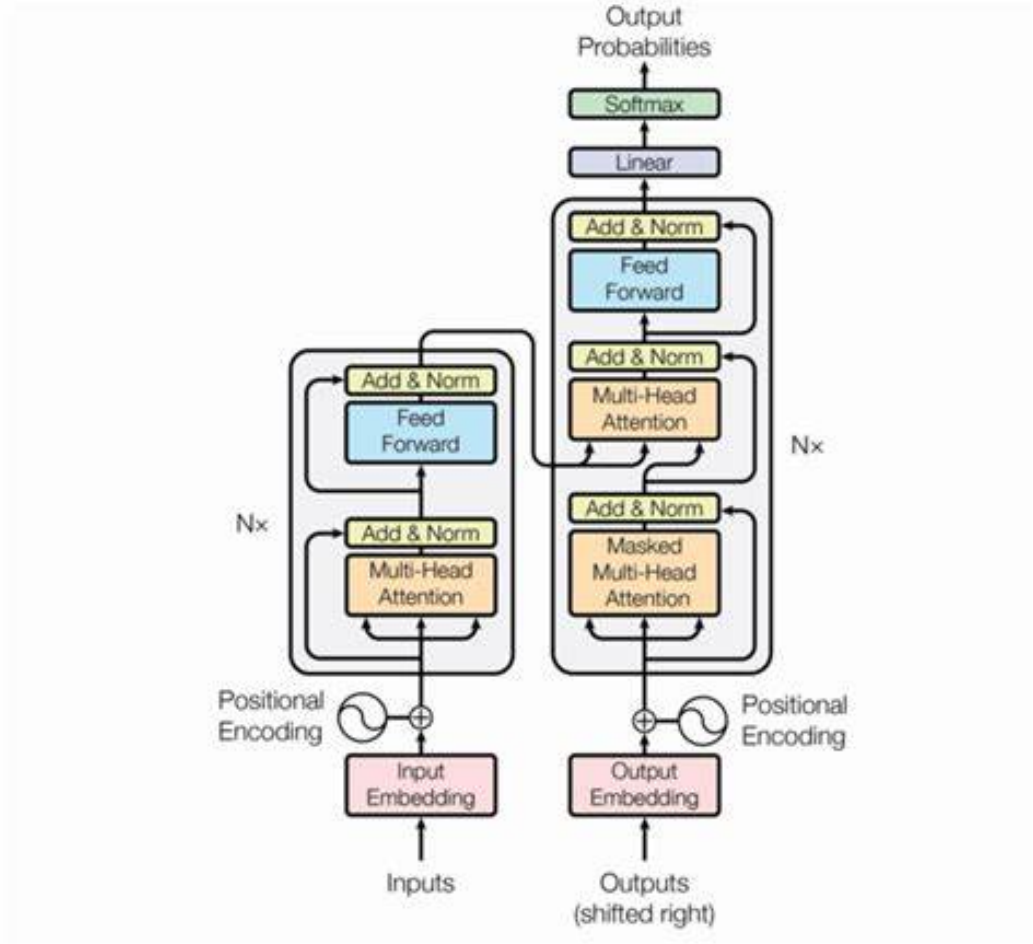
## 1 INTRODUCTION TO PRE-TRAINING

- Pre-training begins by training a neural network on large, unannotated datasets.
- This process uses elf-supervised or unsupervised learning objectives, such as:
  - Predicting missing words (masked language modeling).
  - Forecasting the next token in a sequence (autoregressive modeling).
  - Distinguishing between authentic and corrupted examples (contrastive learning).
- By optimizing these proxy tasks over extensive data, the model learns broad representations of underlying structures, for instance, linguistic patterns like syntax and semantics in text, or visual elements such as edges and textures in images, all without needing manual labels.
- Once this initial foundation is established, the pre-trained parameters provide a valuable starting point for fine-tuning.
- This fine-tuning occurs on smaller, task-specific supervised datasets.
- The benefits of this approach include the following.
  - Downstream models converge more quickly.
  - They require fewer labeled examples.
  - They often achieve superior accuracy compared to models trained from scratch.

# 2 BERT

## 2.1 Architecture

BERT is a transformer-based model.

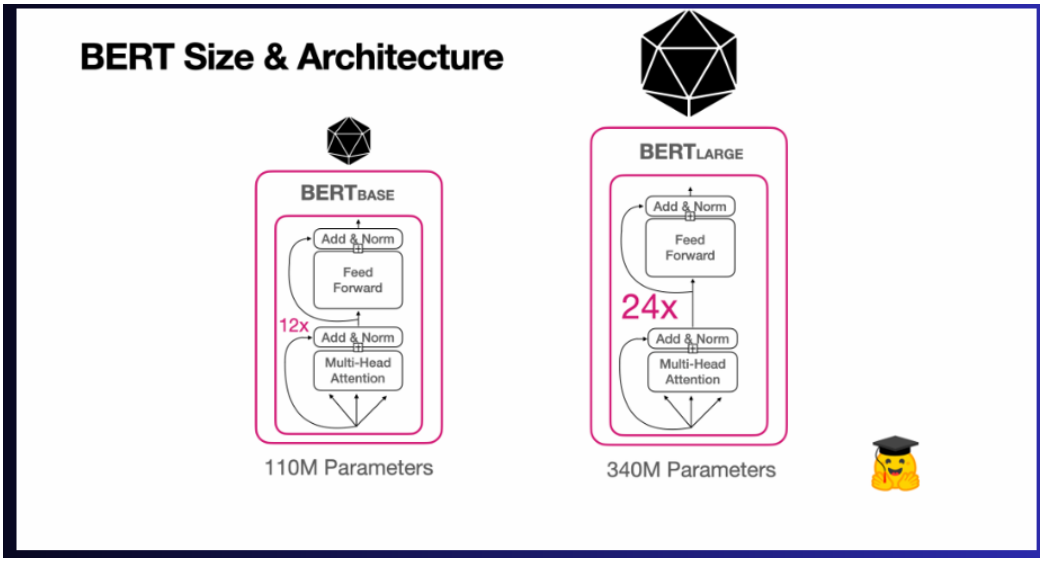



However, BERT's objective is understanding languages, so Google Ai Engineer just uses the encoder of transformer for this architecture.

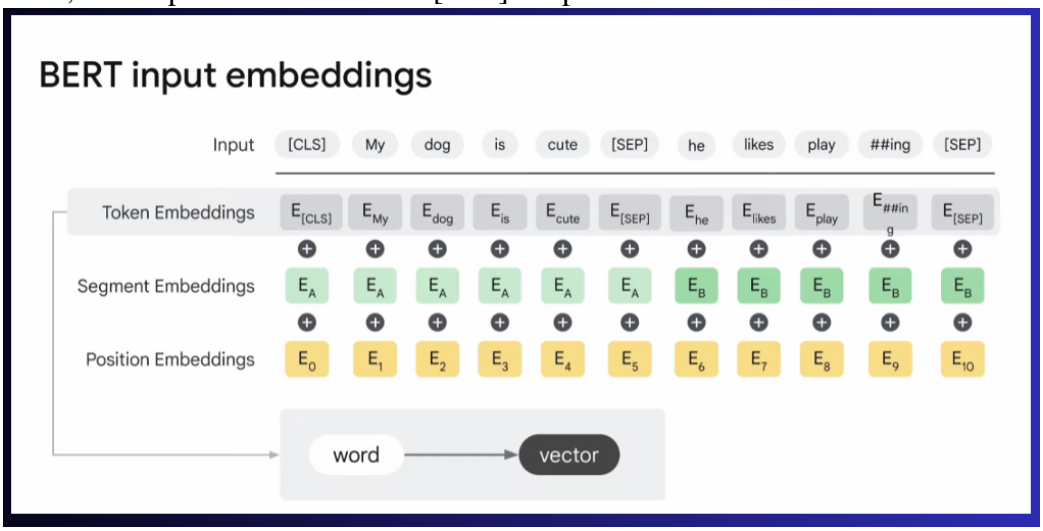
Key Parameters:

- Number of layers (L): 12 (BERTBase) or 24 (BERTLarge)
- Hidden size (H): 768 or 1024
- Attention heads (A): 12 or 16

- Unified architecture for both pre-training and fine-tuning




How does BERT get tokens, [CLS] is token that represents for the start of sentence, more specific tokens about [SEP] - separate two sentences.




## 2.2 Pre-training Methodologies

### 2.2.1 Datasets Used:

- BooksCorpus (800M words)

- English Wikipedia (2,500M words)

#### 2.2.2 Tasks:

- Masked Language Modeling (randomly mask 15% of tokens)
- Next Sentence Prediction: binary classification problem.

### 2.3 Results

The pre-training phase, employing Masked Language Modeling and Next Sentence Prediction on BooksCorpus and English Wikipedia, yielded significant performance enhancements across various NLP tasks.

- **Improved Performance:** Pre-trained models outperform previous results on downstream NLP benchmarks (e.g., question answering, natural language inference, text classification).
- **Faster Convergence:** Fine-tuning these models required fewer epochs and less computational resources.
- **Enhanced Data Efficiency:** Models performed well even with limited labeled data for specific tasks, reducing data scarcity issues.
- **Better Generalization:** The models demonstrated stronger generalization to unseen data, thanks to the broad linguistic features learned during pre-training.
- **Task Contribution:** Both MLM and NSP uniquely contributed, with MLM focusing on bidirectional representations and NSP aiding in inter-sentence understanding.

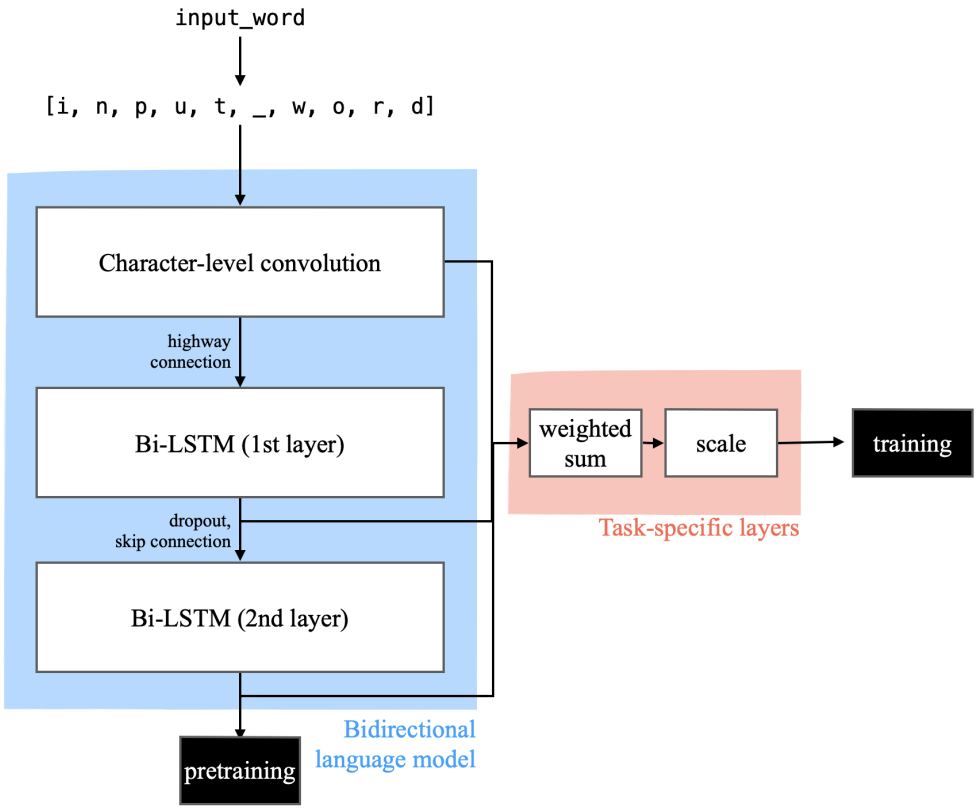
#### 2.3.1 My experiments

- I have already try to build from scratch BERT model to understand architecture, attention mechanism.
- I have already fine-tune pre-trained model on Hugging Face, for NER problem, ... => Results is that this model outperforms my previous model, bi-LSTM and CNN for the same task.

# 3 ELMO

## 3.1 Architecture

ELMO's architecture combines **CNNs for character-level features** and **Bi-LSTMs for contextual understanding** (both left-to-right and right-to-left).




## 3.2 Pre-training Methodologies

ELMO is pre-trained using a **bidirectional language modeling objective**. This involves predicting both the next word from preceding context (forward) and the previous word from following context (backward) on large text corpora, learning deep contextualized word representations.

### 3.3 Results

ELMO significantly advanced NLP by providing:

- **State-of-the-art Performance:** Improved results across many NLP tasks upon it's release.
- **Context-Sensitive Embeddings:** Unique word embeddings based on sentence context.
- **Deep, Layered Representations:** Multiple layers of features adaptable to tasks.
- **Simplified Downstream Models:** Reduced need for complex task-specific architectures.

### 3.4 Disadvantages in current time

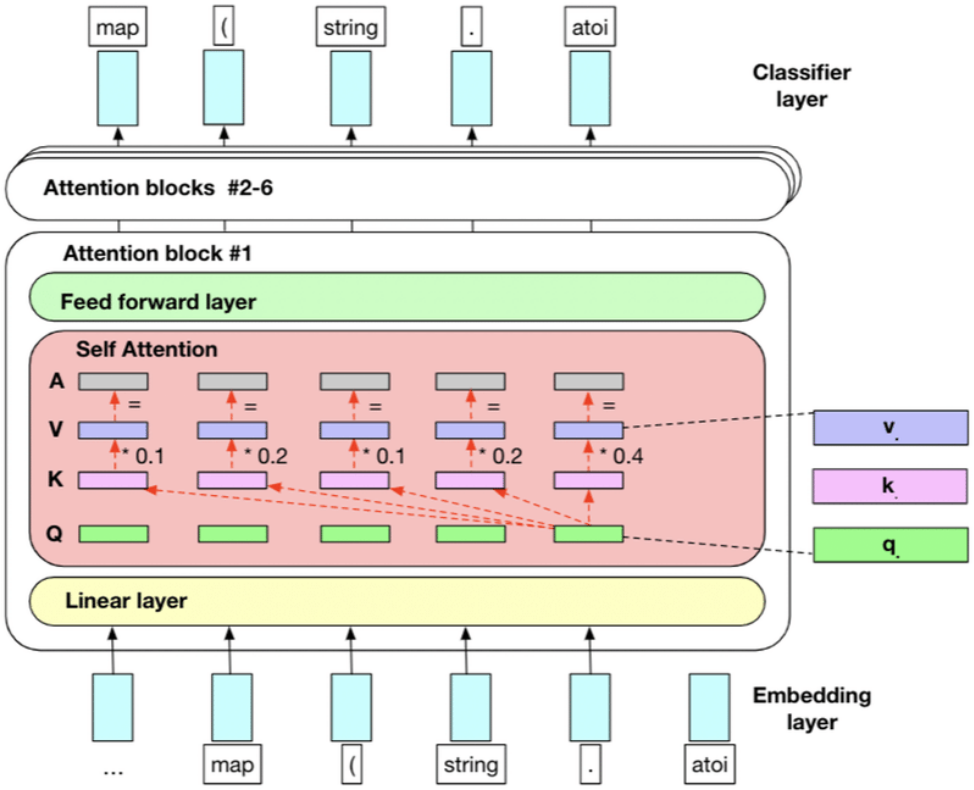
ELMO faces several limitations compared to modern architectures:

- **Poor GPU Parallelization:** Its sequential Bi-LSTM core limits parallel processing on GPUs.
- **Ineffective with Long Data:** Struggles with very long sequences due to LSTM limitations.
- **High Memory Usage:** LSTMs consume significant memory for training and inference.
- **Requires Extra Layers:** Needs additional task-specific layers for downstream applications.

## 4 GPT-2

### 4.1 Architecture

GPT-2 only uses the decoder of the transformer. However, it does not use cross-attention because there is no encoder.




## 4.2 Results

Models pre-trained via next word prediction demonstrate strong capabilities, yielding:

- **Superior Text Generation:** High-quality, coherent, and fluent text generation.
- **Robust Contextual Understanding:** Effective capture of long-range dependencies for comprehension tasks.
- **Efficient Fine-tuning:** Good performance on downstream NLP tasks with minimal examples (zero-shot/few-shot learning).
- **Broad Adaptability:** Valuable for diverse sequence generation tasks like summarization and translation.

## 4.3 My experiment

- Generation Ability is exceptional.
- My result when i fine-tune for generating IELTS Writing Task 2 prove the above ability.
- Upon its release, it is actually SOTA of Generation Task.

=> GPT-2 is also the foundation of today's modern large language models.