VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**

Logo

Description automatically generated

**Deep Learning Final Project Part 1**

*Instructor:* **PGS.TS Lê Anh Cường**

*Student:* **Trương Gia Bảo**

**Hồ Hữu An**

**Trần Nguyễn Duy Bảo**

*Student ID:* **521H0201**

**521H0489**

**521H0493**

*Class:* **21H50302**

**21H50301**

**HO CHI MINH CITY, 2024**

VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**

Logo

Description automatically generated

**Deep Learning Final Project Part 1**

*Instructor:* **PGS.TS Lê Anh Cường**

*Student:* **Trương Gia Bảo**

**Hồ Hữu An**

**Trần Nguyễn Duy Bảo**

*Student ID:* **521H0201**

**521H0489**

**521H0493**

*Class:* **21H50302**

**21H50301**

**HO CHI MINH CITY, 2024**

# ACKNOWLEDGEMENT

I would like to express my sincere thanks to the teachers and lecturers of Ton Duc Thang University and especially Mr. Le Anh Cuong - lecturer of Machine Learnig Group 05 for kindly helping and supporting as well as answering our questions throughout the process of making this report, so that I can get the best results.

**THE PROJECT WAS COMPLETED**

**AT TON DUC THANG UNIVERSITY**

I would like to assure you that this is my own project and guided by Le Anh Cuong. The research contents and results in this topic are honest and have not been published in any form before. The data in the tables for analysis, comments, and evaluations collected by the author himself from different sources are clearly stated in the references section.

In addition, the project also uses some comments, reviews as well as figures of other authors and other organizations with quotes and annotations of origin.

If any fraud is detected, I would like to take full responsibility for the content of my project. Ton Duc Thang University is not involved in copyright or copyright violations caused by me in the process (if any).

*Ho Chi Minh City, 29 March 2024*

*Author*

*(sign and write your full name)*

*Trương Gia Bảo*

*Hồ Hữu An*

*Trần Nguyễn Duy Bảo*

**FACULTY ENDORSEMENTS AND REVIEWS**

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(ký và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(ký và ghi họ tên)

# TABLE OF CONTENTS

[ACKNOWLEDGEMENT 1](#_Toc165326977)

[TABLE OF CONTENTS 1](#_Toc165326978)

[CHAPTER 1: MODEL ARCHITECTURE 2](#_Toc165326979)

[1.1 BERT: ENCODER HALF 2](#_Toc165326980)

[1.1.1 Masked Language Model: 3](#_Toc165326981)

[1.1.2 Next Sentence Prediction (NSP): 5](#_Toc165326982)

[1.2 GPT-3: DECODER HALF 7](#_Toc165326983)

[1.2.1 Autoregressive language model 7](#_Toc165326984)

[1.2.2 Semi-supervised Learning: 7](#_Toc165326985)

[1.3 COMPARISION BETWEEN BERT AND GPT-3: 11](#_Toc165326986)

[CHAPTER 2: CONTINUE PRE-TRAINNING & FINE TUNING 13](#_Toc165326987)

[2.1 CONTINUE PRE-TRAINNING: 13](#_Toc165326988)

[2.2 FINE TUNING: 14](#_Toc165326989)

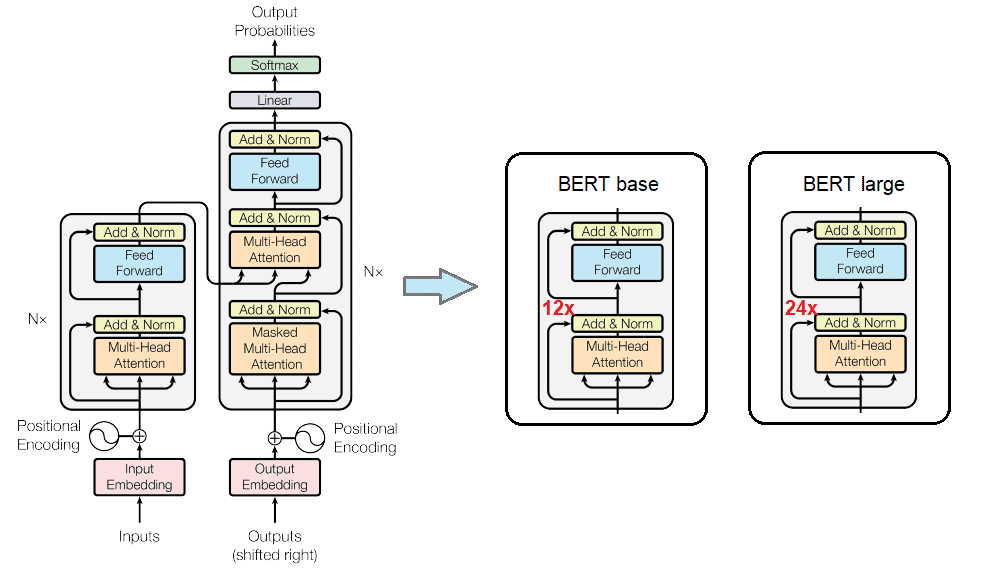
[REFERENCE 16](#_Toc165326990)

# CHAPTER 1: MODEL ARCHITECTURE

## 1.1 BERT: ENCODER HALF

BERT, stands for Bidirectional Encoder Representations from Transformers, is pre-trained using a combination of masked language modeling and next sentence prediction tasks on a large corpus that includes the Toronto Book Corpus and Wikipedia. Engineered to generate deep bidirectional representations from unlabeled text, it considers both left and right context across all layers.

BERT is solely focused on processing input sequences and does not generate any output sequence. It employs a self-attention mechanism on input tokens, enabling it to concentrate on the most relevant parts of the input for tasks such as classification, question answering, and sentence tagging.



**Architecture:**

***BERT****BASE***​: L**=12**, H=**768**, A**=12**,** Total Parameters=110M

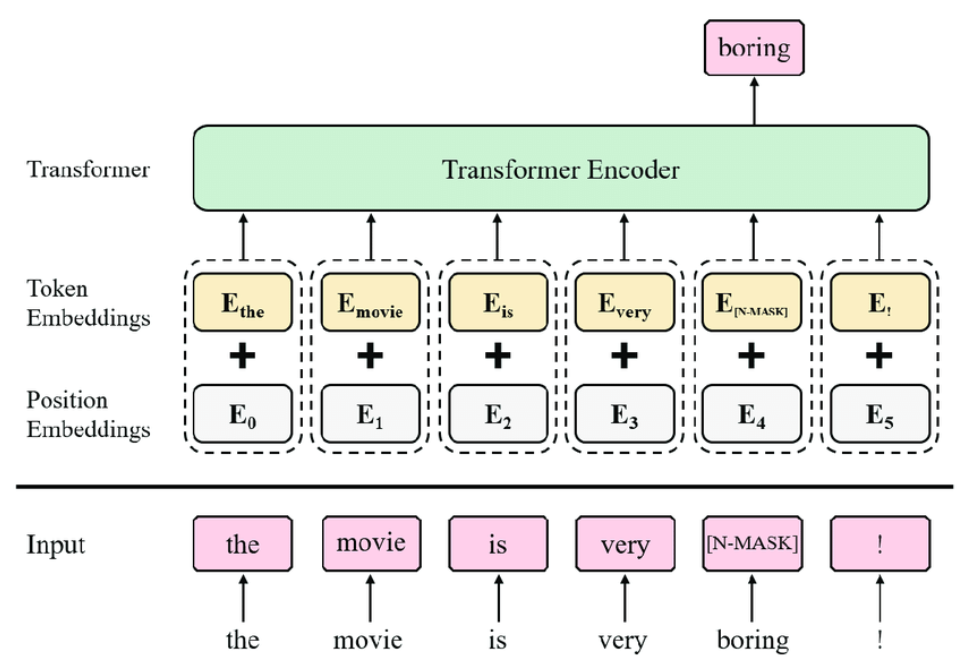
**𝐵𝐸𝑅𝑇***LARGE***​: L**=24**, H**=1024**, A**=16**,** Total Parameters=340M

***L****: the number of Transformer layers(blocks)*

***H****: size of the hidden layers*

***A****: the number of heads in the attention*

### **1.1.1 Masked Language Model:**



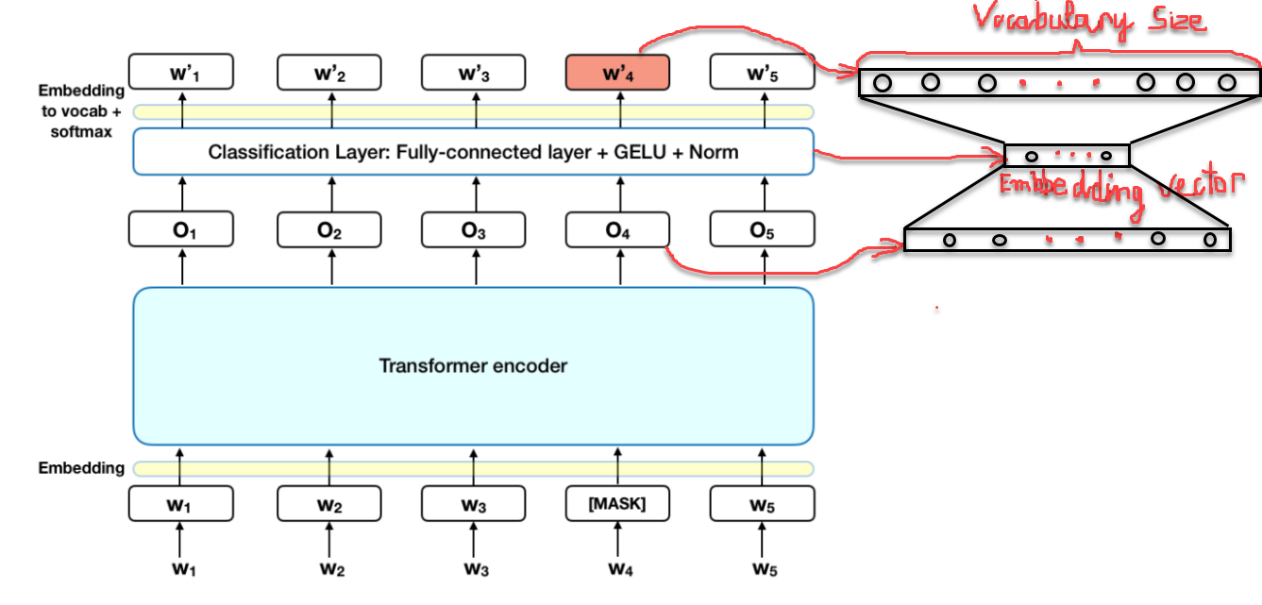
Masked language model is a type of language model used in training neural networks for natural language processing (NLP) tasks. In this model, some words in the input sentence are "masked" before being fed into the network for training. The neural network then tries to predict the masked words based on their surrounding context.

Specifically, during training, some words in the input sentence are randomly selected to be replaced by a special token representing "mask" (e.g., [MASK] in BERT).

The goal of the model is to predict masked words based on information from the remaining words in the sentence.

The MLM model is often used in models such as BERT (Bidirectional Encoder Representations from Transformers), where training in both directions of context (left and right) is combined with the goal of predicting masked words. This helps the model understand and represent the context of each word in a sentence more comprehensively.

The hidden vectors in the last layer corresponding to the hidden tokens are put into a softmax layer over the entire vocabulary for prediction. Google researchers tested masking 15% of all tokens taken from WordPiece's dictionary in sentences at random to only predict masked words.



About 15% of the input sentence tokens are replaced by [MASK] tokens before being passed into the model representing masked words. The model will rely on non-masked words around [MASK] and also the context of [MASK] to predict the original value of the masked word. The number of masked words is chosen to be a small number (15%) so that the context ratio is larger (85%).

The essence of the BERT architecture is still a seq2seq model consisting of 2 phase encoders that help embedding input words and a decoder that helps find the probability distribution of words in the output. The Transformer encoder architecture is retained in the Masked ML task. After performing self-attention and feed forward, we will obtain the embedding vectors in the output

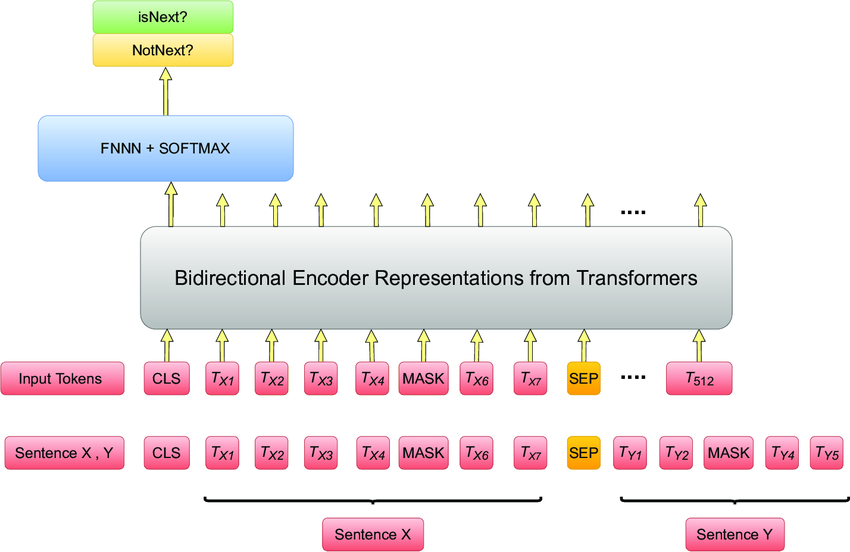
To calculate the probability distribution for the output word, we add a Fully connect layer right after the Transformer Encoder. The softmax function is used to calculate probability distributions. The number of units of the fully connected layer must be equal to the size of the dictionary.

Finally, we obtain the embedding vector of each word at the MASK position which will be the embedding vector reducing the dimensionality of the vector after going through the fully connected layer as described in the drawing on the right.

### **1.1.2 Next Sentence Prediction (NSP):**

This is a supervised classification problem with 2 labels (also known as binary classification). The model's input is a pair-sequence such that 50% of the second sentence is selected as the next sentence of the first sentence and 50% is selected randomly from the text set without any What is the connection with the first sentence? The model's label will correspond to IsNext when the sentence pair is consecutive or NotNext if the sentence pair is not consecutive.

Similar to the Question and Answering model, we need to mark the beginning positions of the first sentence with the token [CLS] and the end positions of the sentences with the token [SEP]. These tokens identify the starting and ending positions of each first and second sentence.



Ex:

*Sentence pair 1: "The sun is shining."*

*Sentence pair 2: "It's a beautiful day."*

* ***isNext***

However, if we have the following pair of sentences:

*Sentence pair 1: "The sun is shining."*

*Sentence pair 2: "Football is a popular sport."*

* ***NotNext***

## 1.2 GPT-3: DECODER HALF

GPT-3 (Generative Pre-trained Transformer 3) is an autoregressive language model that was created by OpenAI. Its purpose is to generate text that closely resembles human language. To achieve this, GPT-3 was trained on a vast dataset consisting of approximately 45TB of text data. This dataset included various sources such as Wikipedia, books, and websites.

Due to its extensive training, GPT-3 has the ability to generate text that mimics human-like language patterns and structures. In addition to text generation, GPT-3 can also be utilized for a wide range of tasks, including answering questions, summarizing information, and translating languages.

### **1.2.1 Autoregressive language model**

An autoregressive language model is a type of Machine Learning model that uses autoregressive techniques to predict the next word in a sequence of words based on the words that have come before it. This can be used for tasks such as natural language processing and machine translation

### **1.2.2 Semi-supervised Learning:**

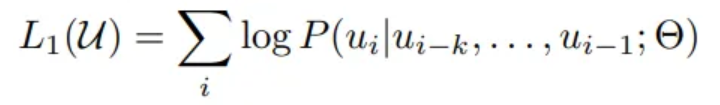
This approach is a ***mixture of supervised and unsupervised learning***. As the cost of making labeled datasets for language tasks is quite high because they require professionals. OpenAI came up with an approach of ***unsupervised pre-training*** and ***supervised fine-tuning***.

Their training procedure consists of two stages:

* The first stage is learning a high-capacity language model on a large corpus of text.
* This is followed by a fine-tuning stage, where they adapt the model to a discriminative task with labeled data.

1. **Unsupervised pre-training:**

In unsupervised pre-training, we have an unlabeled corpus of text and our objective is to maximize the log-likelihood of next word while given previous words. We are using the concept of conditional probability here. Unidirectional pretraining is performed here.



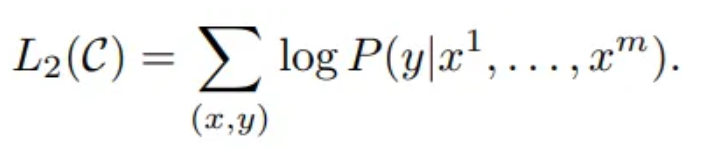
Objective for pre-training

We are taking **k** as the context window. In simple words, we can look back at **k** tokens while predicting **(k+1)th** token.

During the pre-training, a multi-layer Transformer decoder is used. Multi-headed self-attention operation is applied over the input token followed by a position-wise feed-forward network. The output is a distribution over the target tokens.

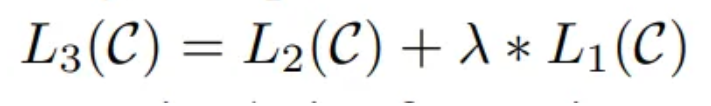
1. **Supervised fine-tuning:**

In supervised fine-tuning, we have a labeled dataset **‘y’** as labels and**‘x’**asinputs. The inputs are passed through the pre-trained model and the output from the final transformer block is fed into an added linear output layer with parameters **Wy** to predict **y.**This gives us the following objective to maximize:



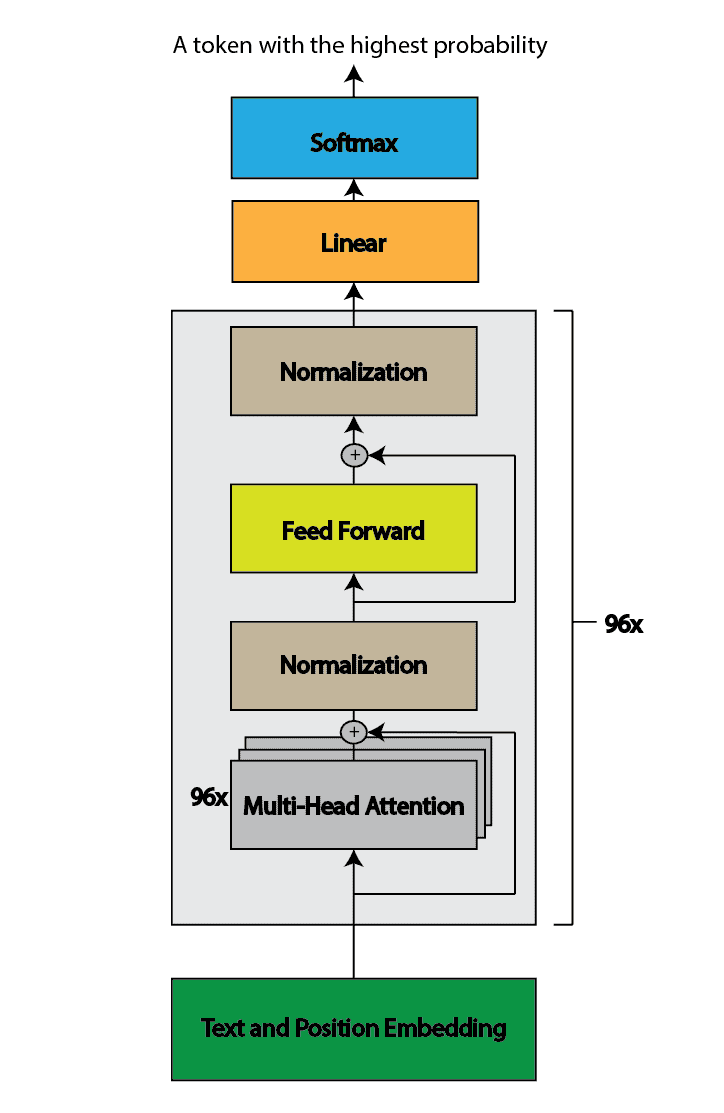
Intermediate objective

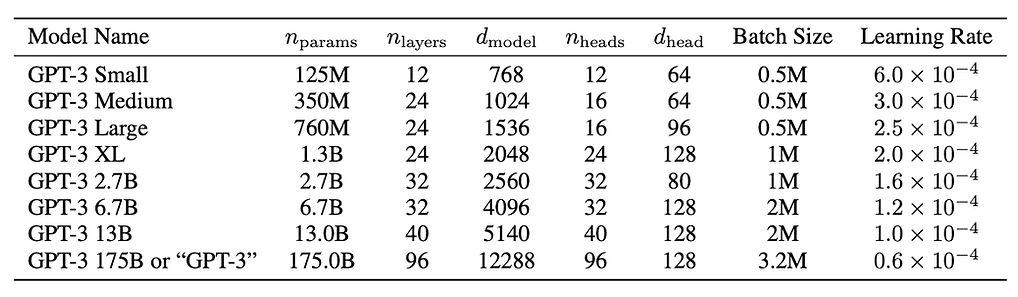
It was found that using the pre-training objective as an auxiliary objective while fine-***tuning improves generalization and accelerates convergence***. So **L1**was made part of the final objective with weight.



Final objective for fine-tuning

**Architecture:**





***nparams****:* *Number of parameters in the model*

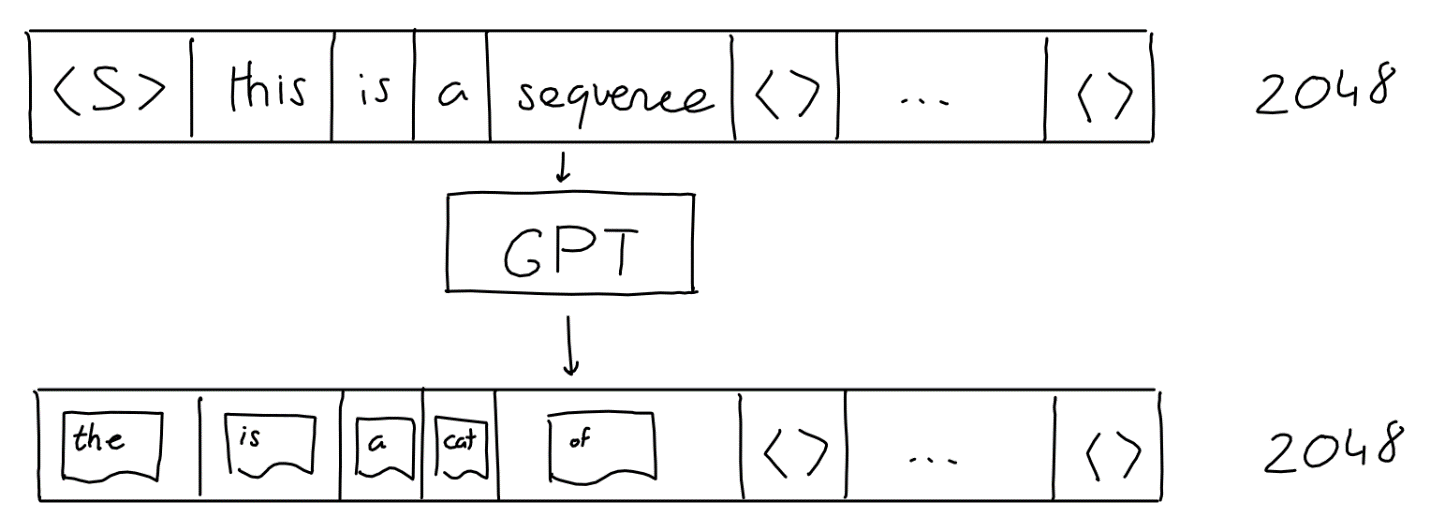
***nlayers****: Number of layers in the model*

***dmodel****:* *Dimensionality of the hidden vector*

***nheads****:* *Number of attention heads*

***dhead****:* *Dimensionality of the attention head vector*

**Input/Output:**



**Input**: The GPT input is a sequence of N words, actually fixed to 2048 words (for GPT-3)

**Output**: The GPT output is not just a single guess, it's a sequence (length 2048) of guesses (a probability for each likely word). One for each 'next' position in the sequence.

Ex: after we get the next word, we add it to the sequence, and get the following word.

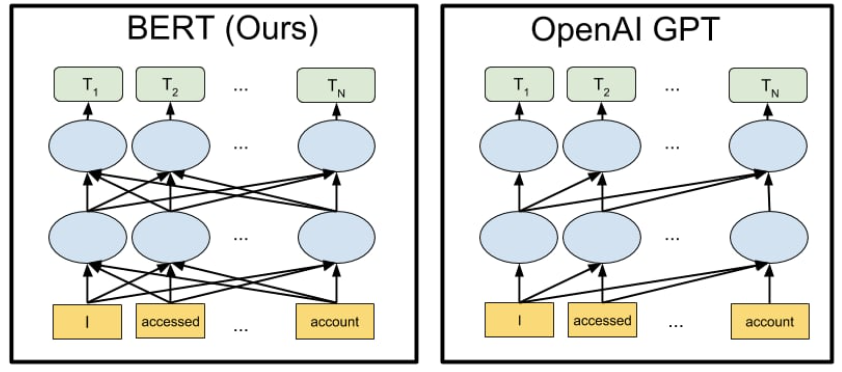
*Not all heroes wear capes ->* ***but***

*Not all heroes wear capes but ->* ***all***

*Not all heroes wear capes but all ->* ***villans***

*Not all heroes wear capes but all villans ->* ***do***

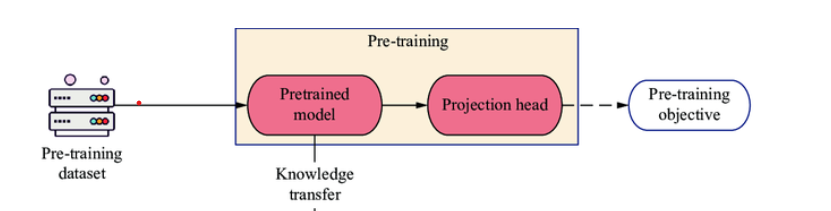
## 1.3 COMPARISION BETWEEN BERT AND GPT-3:



|  |  |  |
| --- | --- | --- |
| **Feature** | **BERT** | **GPT-3** |
| Model Type | BERT is a ***bidirectional*** Transformer-based model. | GPT-3 is a ***unidirectional*** Transformer-based model. |
| Training | BERT is trained using masked language modeling (MLM) and next sentence prediction (NSP) tasks. | GPT-3 is trained using autoregressive language modeling. |
| Part Used | BERT uses the encoder part of the model to generate contextual word representations. | GPT-3 uses the decoder part of the model to generate new text. |
| Used in Tasks | BERT is suitable for tasks such as text classification, entity recognition, and question answering. | GPT-3 is suitable for text generation tasks such as summarization, translation, and text completion. |
| Context Understanding | BERT understands context from both directions, allowing it to comprehend word meanings based on context. | GPT-3 only understands context from the preceding words, hence it has limitations in understanding context. |

# CHAPTER 2: CONTINUE PRE-TRAINNING & FINE TUNING

## 2.1 CONTINUE PRE-TRAINNING:

******

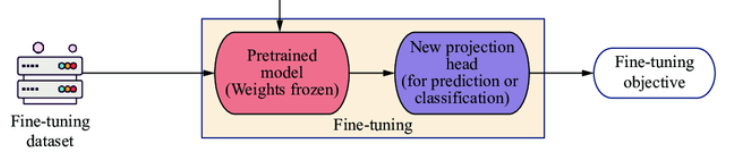
**Definition:**

Continued pre-training extends a model's pre-training using additional data to ***broaden its understanding***. The ***entire model is updated, refining its representations***. It's essential for keeping the model up-to-date with evolving language patterns and domains.

**Objective:**

The aim is to improve the model's ***general performance and understanding*** by exposing it to more varied or specialized data. This can enhance the ***model's ability to generalize across tasks*** ***or improve its performance on tasks*** related to the new data.

## 2.2 FINE TUNING:



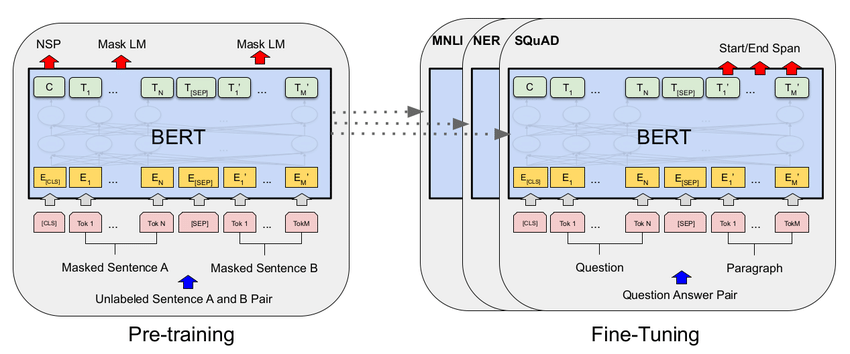
**Definition:**

Fine-tuning entails adapting a pretrained model to a specific task or dataset by adjusting its parameters. Typically, this involves modifying only ***the top layers*** while ***keeping the lower layers fixed****.*

During fine-tuning, parameters of the new, task-specific layers (usually added on top of the pretrained layers) are updated, while the pretrained layers' parameters remain fixed or are updated with a very low learning rate.

**Objective:**

The primary goal of fine-tuning is to leverage the *generic knowledge* the model has acquired during pre-training and adapt it to *a specific task or domain*, thereby improving its accuracy and effectiveness on that task.



After the pre-training phase, the BERT model, armed with its contextual embeddings, is then fine-tuned for specific natural language processing (NLP) tasks. This step tailors the model to more targeted applications by adapting its general language understanding to the nuances of the particular task.

BERT is fine-tuned using labeled data specific to the downstream tasks of interest. These tasks could include sentiment analysis, question-answering, named entity recognition, or any other NLP application. The model’s parameters are adjusted to optimize its performance for the particular requirements of the task at hand.

# REFERENCE

[1] <https://viblo.asia/p/hieu-hon-ve-bert-buoc-nhay-lon-cua-googleeW65GANOZDO#_cac-thu-nghiem-cua-bert-7>

[2] <https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270>

[3] <https://www.geeksforgeeks.org/explanation-of-bert-model-nlp/>

[4] <https://phamdinhkhanh.github.io/2020/05/23/BERTModel.html>

[5] <https://www.linkedin.com/pulse/teaching-old-dog-new-tricks-difference-between-lewis-ms-ccrp-ches-3abqc/>

[6] <https://www.researchgate.net/figure/llustration-of-pre-training-Pre-training-is-a-part-of-transfer-learning-If-the_fig1_368698963>

[7] <https://www.baeldung.com/cs/bert-vs-gpt-3-architecture>

[8] <https://discuss.huggingface.co/t/continue-pretraining-on-a-released-model/67604>

[9] <https://medium.com/@eordaxd/fine-tuning-vs-pre-training-651d05186faf>

[10] <https://krrai77.medium.com/decoding-language-models-a-comprehensive-tutorial-with-gpt-3-bert-and-transformers-4978f04f0115?source=-----4978f04f0115---------------------post_regwall-----------&skipOnboarding=1>

[11] <https://www.toolify.ai/vi/ai-news-vn/gpt3-t-ph-trong-x-l-ngn-ng-t-nhin-1167316>

[12] <https://elixirforum.com/t/gpt3-tokenizer-bpe-encoder-decoder-for-gpt-3-implemented-in-elixir/55642>

[13] <https://blog.invgate.com/gpt-3-vs-bert#:~:text=For%20example%2C%20GPT%2D3%20is,you%20are%20looking%20to%20accomplish>

[14] <https://medium.com/nerd-for-tech/gpt3-and-chat-gpt-detailed-architecture-study-deep-nlp-horse-db3af9de8a5d>