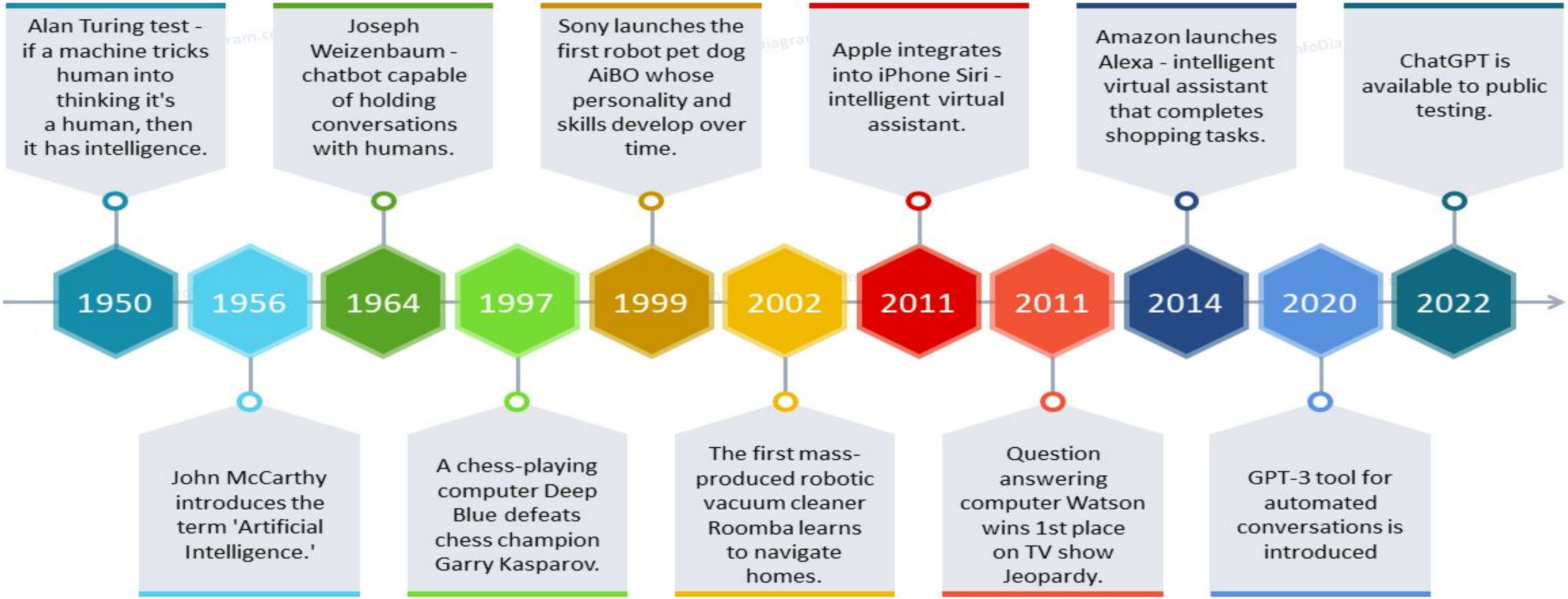
All about Transformers

Aivalis Theodoros

22/09/2023

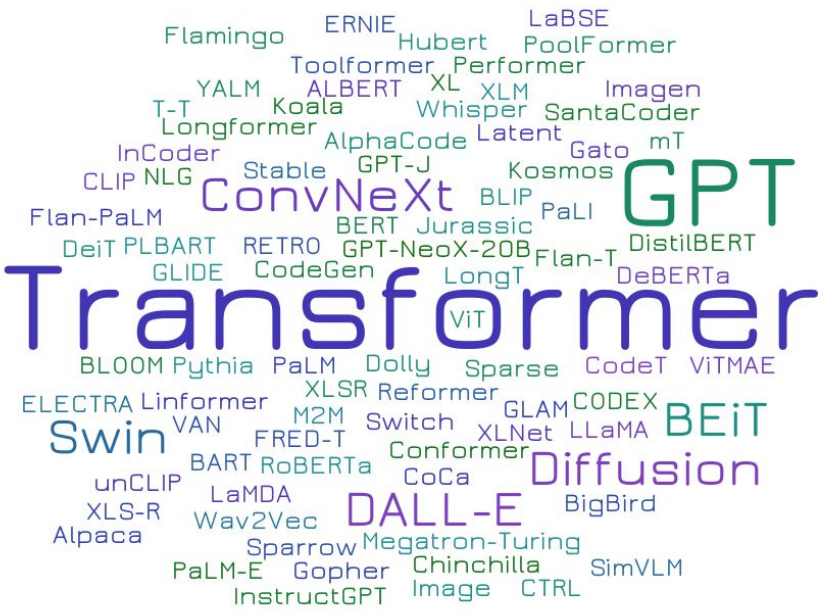
AI’s History Timeline



AI’s History Timeline [1]

Attention is All you need!

- Imports an attention mechanism to draw global dependencies between input and output sequences.[3]



Popular Transformer Models [2][3]

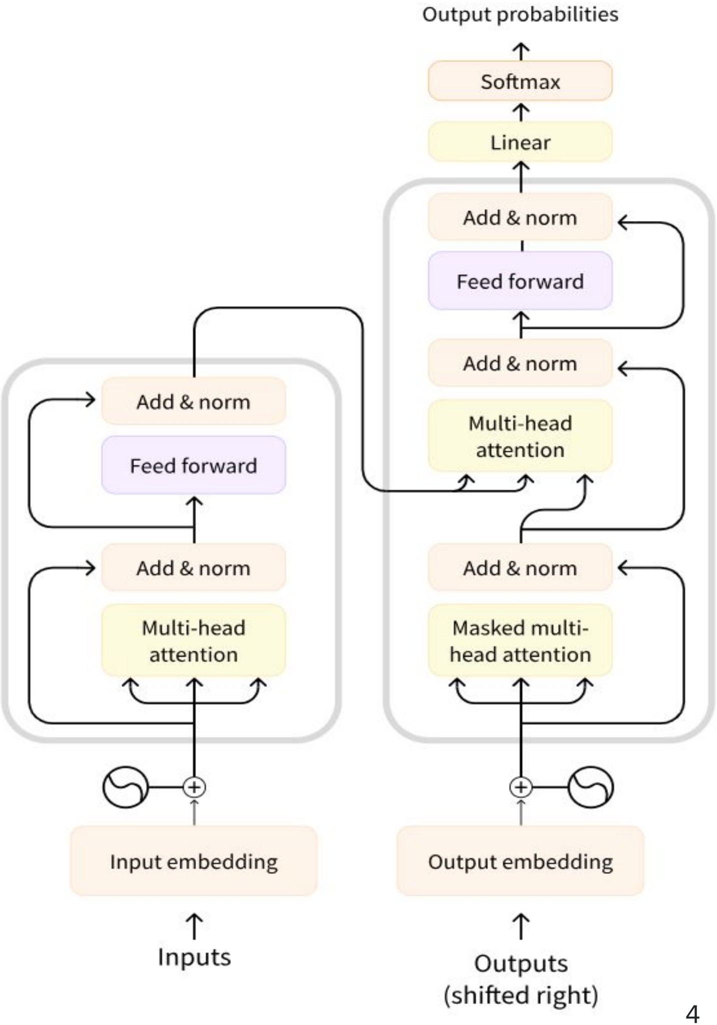
Structure and Architecture

A Transformer model is composed of two blocks:

- Encoder (left)
- Decoder (right)

Each of parts can be used independently, depending on the task:

- Encoder-only models
- Decoder-only models
- Encoder-decoder models or sequence-to-sequence models. [4]

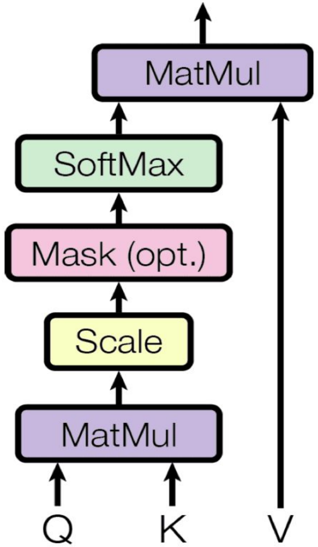


Transformer Architecture [4]

Attention Mechanism

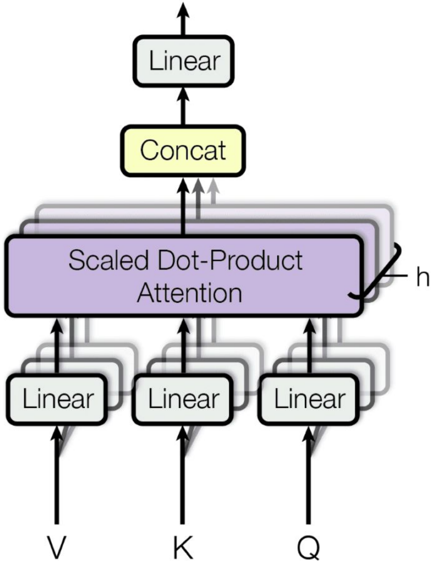
''The attention function can be considered as a mapping between a query and a set of key-value pairs to an output.''

- Scaled Dot-Product Attention [5]



$$attention(Q,K,V) = soft\max\left(\frac{QK^T}{\sqrt{d_K}}\right)V$$

- Multi-Head Attention [5]

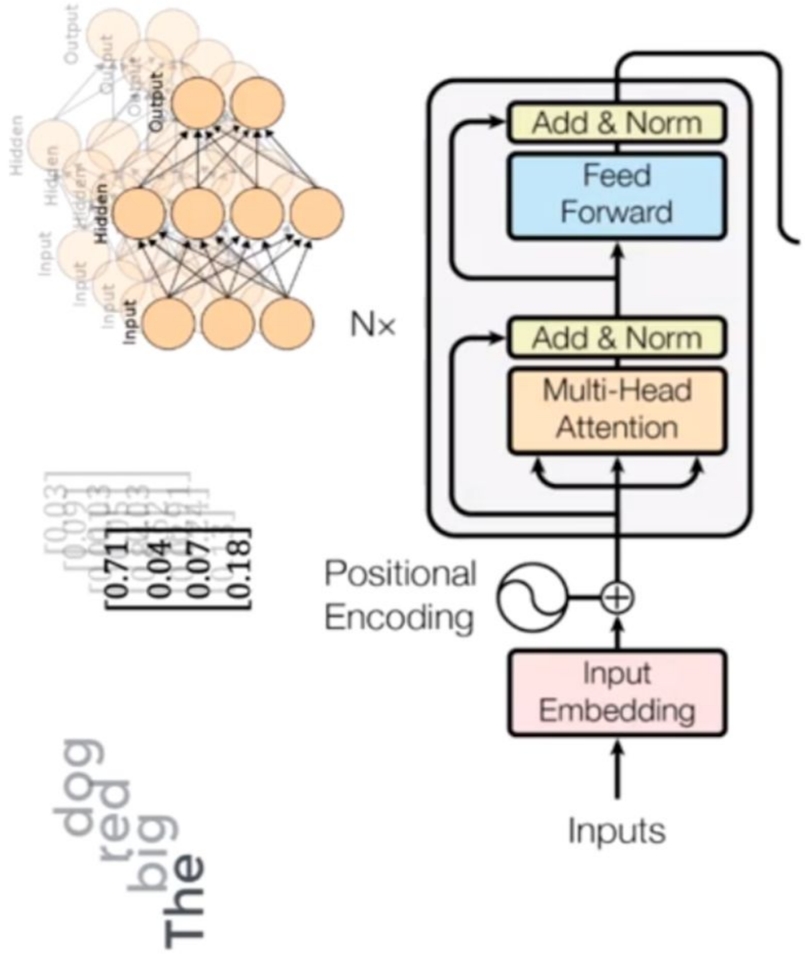


$$multihead(Q,K,V) = concat(head_1,\ldots,head_h)\,W^O$$

$$head_i = attention\left(QW_i^Q, KW_i^K, VW_i^V\right)$$

Encoder Block

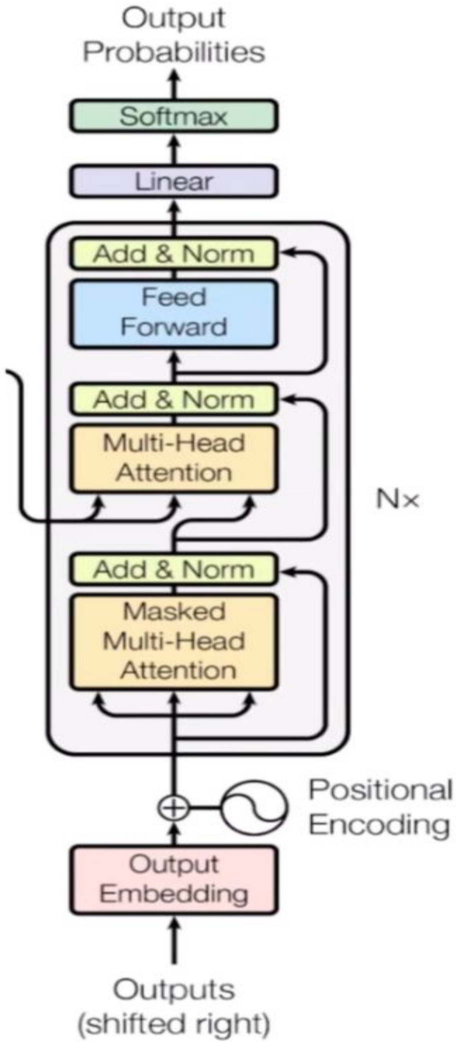
- Words as input to create numerical vectors (Embeddings).
- Positional Encoding for words with multiple meanings (depending of their position).
- Attention layer to create attention vectors (about the relations between the words).
- Feed-forward Neural Network (convert into an acceptable form for the next layer). [6]



Encoder Block [6]

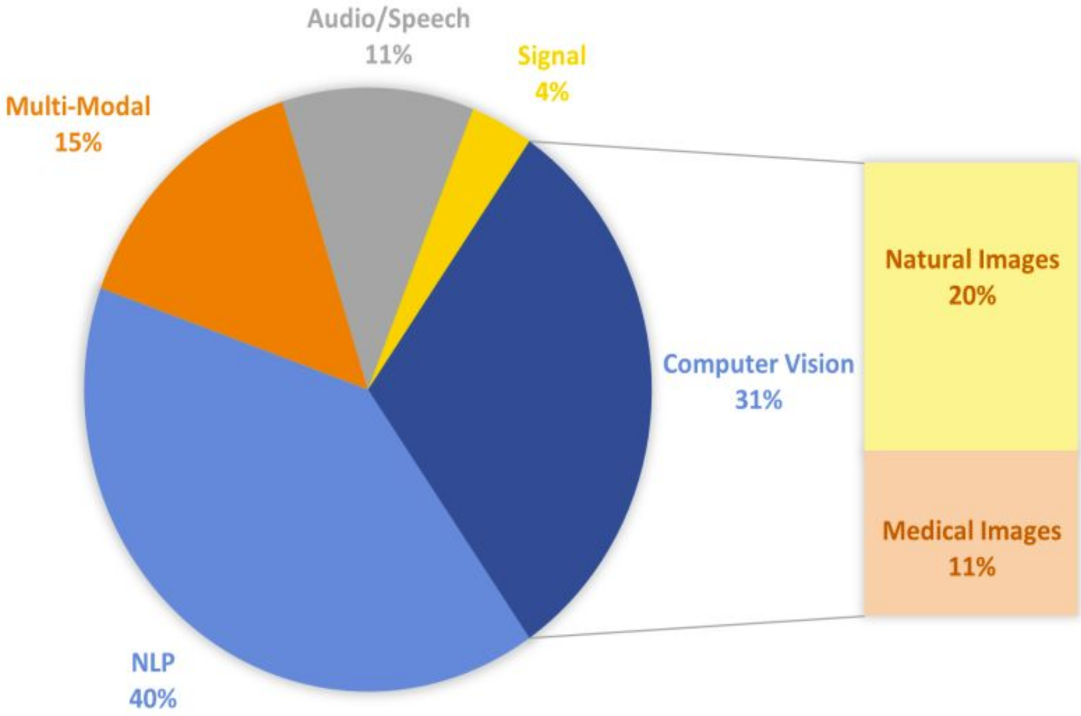
Decoder Block

- Give outputs as decoder's input in order to learn.
- Create vectors from words in Embedding and Positional Encoding Part similar to the Encoder.
- Create attention vectors for every word to represent relations of every word in the sentence.
- Attention vector that combines vectors from both encoder and decoder blocks.
- Use a feed-forward layer to form it for a Linear layer where it expands the dimensions of the output.
- Softmax layer that transforms the input into a probability distribution. [6]



Decoder Block [6]

Application-Based Classification of Transformers

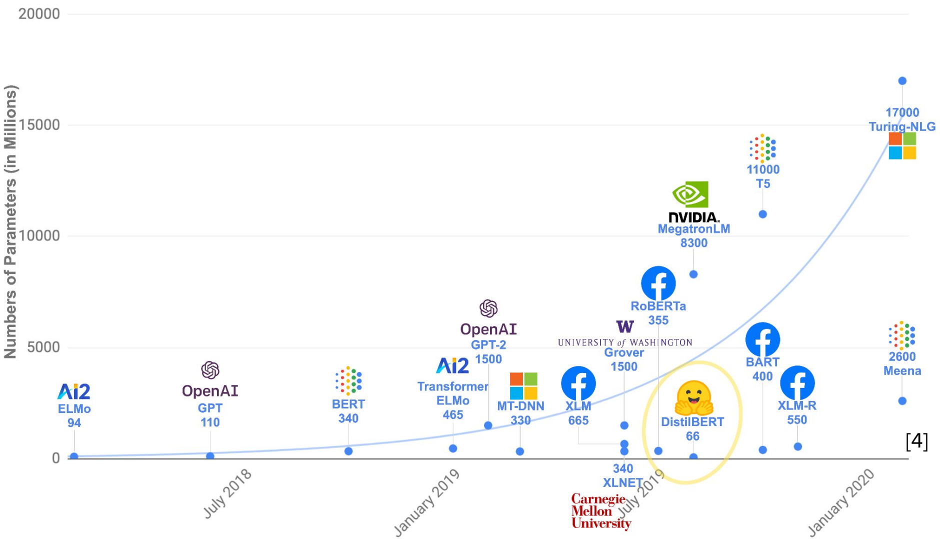


Applications of Transformers [7]

- Natural Language Processing (NLP)
  - Language Translation
  - Questioning Answering
  - Text Summarization & Generation
  - Natural Language Reasoning
- Computer Vision
  - Natural & Medical Image Processing
  - Image Generation & Segmentation
  - Image Classification
  - Object Detection
- Multi-Modal
- Audio/Speech
- Signal [7]

Popular Transformer Models

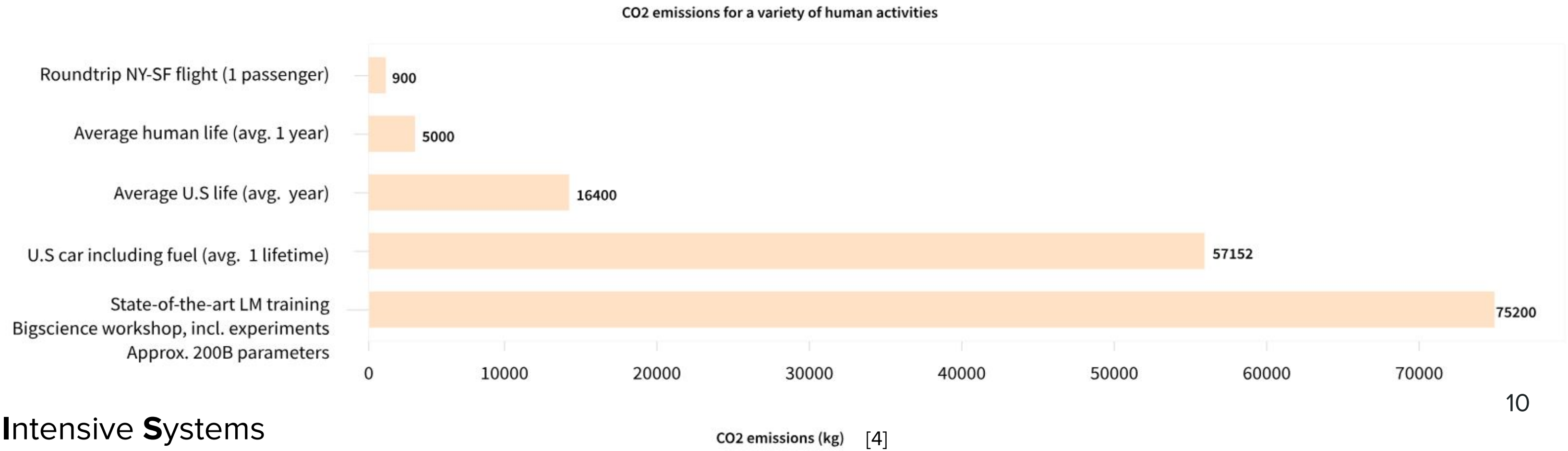
Transformer	**BERT** [8],[9],[10]	**GPT-3** [8],[10],[11]	**T-5** [8],[10],[12]
Main Uses	Questioning Answering, Text Classification, Text Generation, Sentiment Analysis	Natural Language Processing (NLP), Natural Language Generalization(NLG)	Language Translation, Question Answering
Popular Applications	Google Search (since 2020)	Chat-GPT, Dall-E	
Parameters	110 M(based) 340 M(large)	175 B	60 M to 11B
Training Time	1-130 min (fine-tuning on a GPU) 4 days on 4 Cloud TPUs (base) 4 days on 16 Cloud TPUs (large)	355 years on a single GPU (theoretically)	few days on Cloud TPUs



[4]

Challenges and Limitations

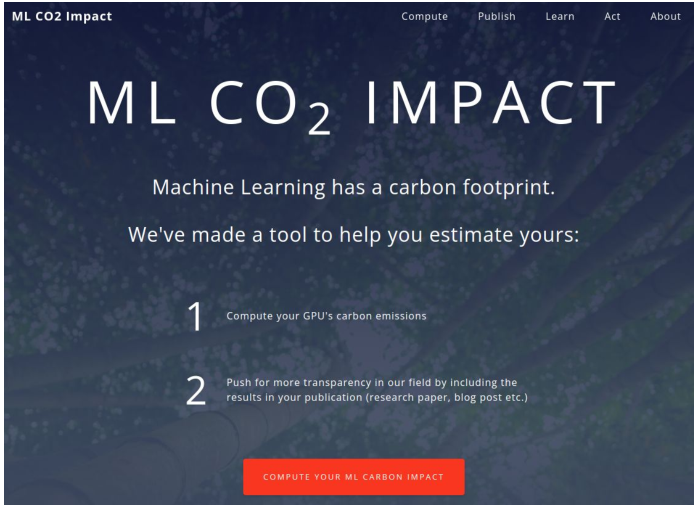
- Computational resources and memory to run and train.
- Large cost of deployment and scaling(especially on LLM's)
- Overfitting, generalization and robustness issues (noisy or incomplete data).
- Lack interpretability and explainability.
- Carbon footprint of transformer models. [13]



[4]

Possible solutions to consider

- Using pretrained models when they are available
- Fine-tuning vs Training from scratch
- Starting with small experiments and debugging
- Doing a review to choose hyperparameter ranges
- Random search vs Grid Search [4], [14]



Tool for estimate carbon footprint of your ML model. [14]

Fine-tune a pre-trained Transformer

- Load the model and the tokenizer
- Prepare Datasets (Train, Validate, Test)
- Compute metrics
- Combine everything in a trainer
- Start Training [10], [15]

```
[ ] from transformers import AutoTokenizer, AutoModelForSequenceClassification, DataCollatorWithPadding
    from torch.utils.data import DataLoader

    BASE_MODEL = "camembert-base"
    LEARNING_RATE = 2e-5
    MAX_LENGTH = 256
    BATCH_SIZE = 16
    EPOCHS = 20

    tokenizer = AutoTokenizer.from_pretrained(BASE_MODEL)
    model = AutoModelForSequenceClassification.from_pretrained(BASE_MODEL, id2label=id2label, label2id=label2id)
```

```
[ ] import numpy as np
    from datasets import load_metric

    metric = load_metric("accuracy")

    def compute_metrics(eval_pred):
        logits, labels = eval_pred
        predictions = np.argmax(logits, axis=-1)
        return metric.compute(predictions=predictions, references=labels)
```

```
[ ] from transformers import Trainer

    trainer = Trainer(
        model=model,
        args=training_args,
        train_dataset=ds["train"],
        eval_dataset=ds["validation"],
        compute_metrics=compute_metrics
    )

    trainer.train()
```

References

1. AI & Machine Learning Presentation Diagrams (PPT template)
2. 5 Top Most Powerful Transformer Models 2023 | Codementor
3. Attention Is All You Need, https://arxiv.org/pdf/1706.03762.pdf
4. How do Transformers work? - Hugging Face NLP Course
5. The Transformer Attention Mechanism - MachineLearningMastery.com
6. Transformer Neural Networks: A Step-by-Step Breakdown | Built In
7. A Comprehensive Survey on Applications of Transformers for Deep Learning Tasks, https://arxiv.org/pdf/2306.07303.pdf
8. Top 9 Transformer Models | Saturn Cloud Blog
9. How does BERT work and what are the use cases of BERT ? | by Shushant Pudasaini | Medium
10. 🤗 Transformers
11. What is GPT-3? Everything You Need to Know - TechTarget
12. Exploring Google's T5 Text-To-Text Transformer Model | T5_transformer – Weights & Biases
13. What are the current and future trends and directions of transformer models for AI?
14. Machine Learning CO2 Impact Calculator
15. Regression with Text Input Using BERT and Transformers | by La Javaness R&D