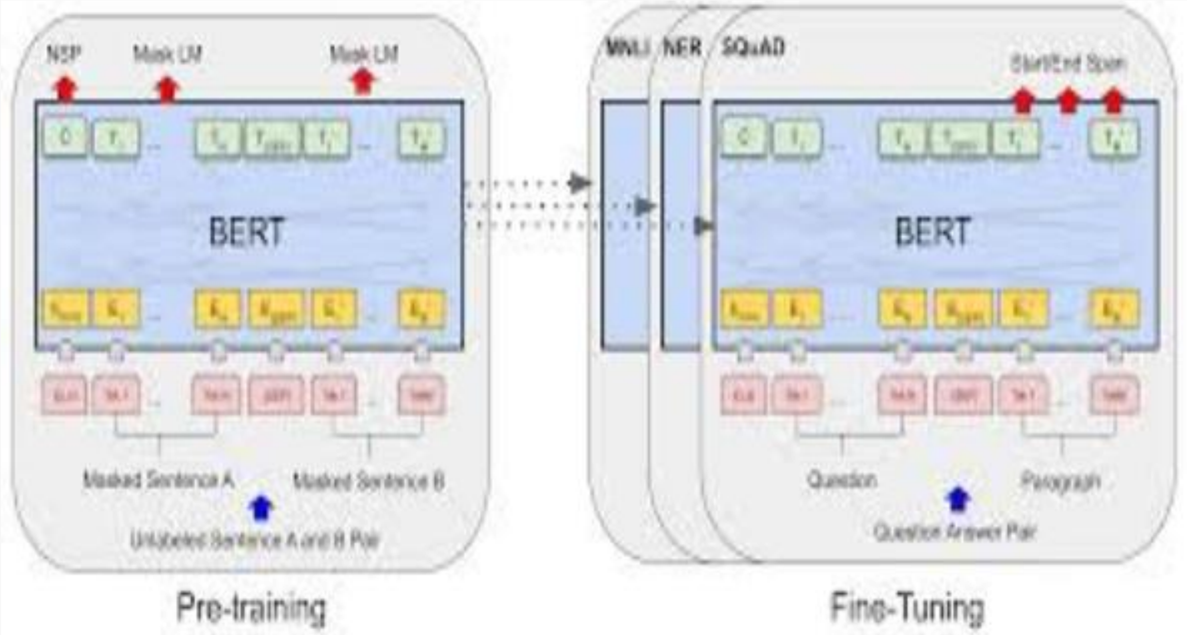
Project Mental - 2024 : Bert



BERT

- Bidirectional Encoder Representation from Tranfomer

BERT?

BERT'S KEY FEATURES

- Bidirectional Encoding
- Transformer Architecture

BERT

COMPARE WITH PREVIOUS MODEL

PREVIOUS

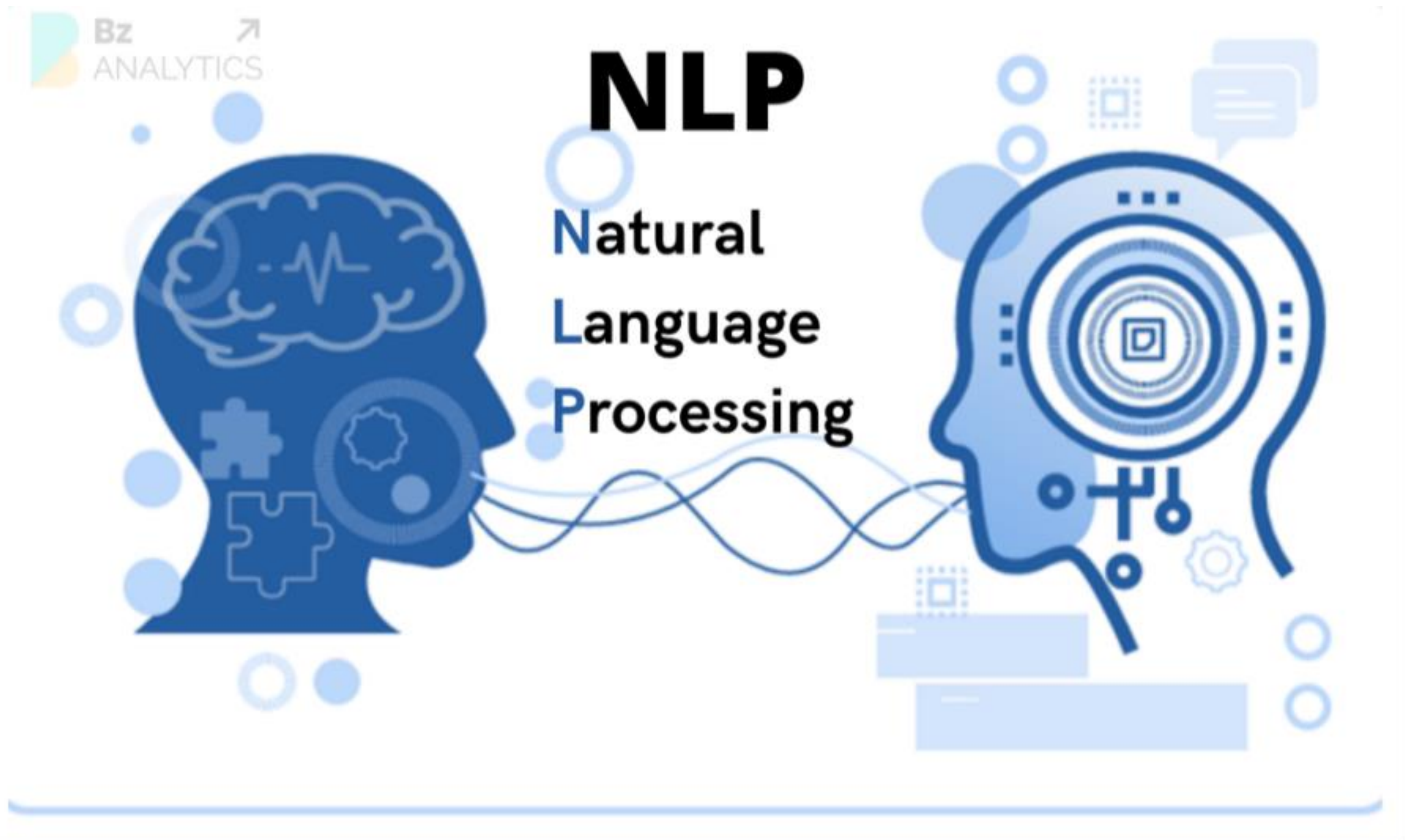
- Sequential Processing
- Unidirectionality
- Word Embeddings limits

BERT

- Bidirectionality
- Pretraining + Fine-Tuning
- Contextualized Word Embeddings
- Use of Transformers

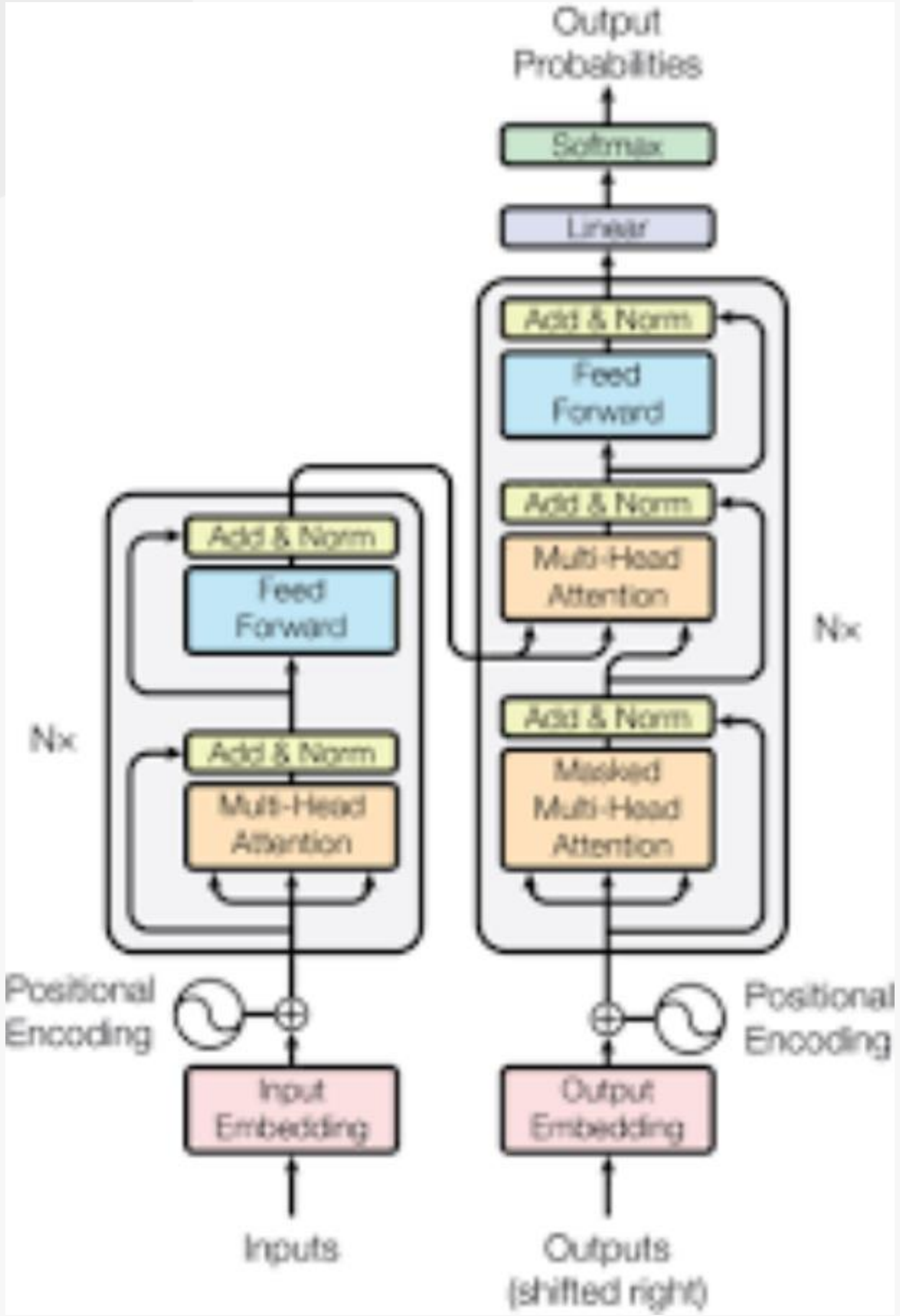
NLP

NLP



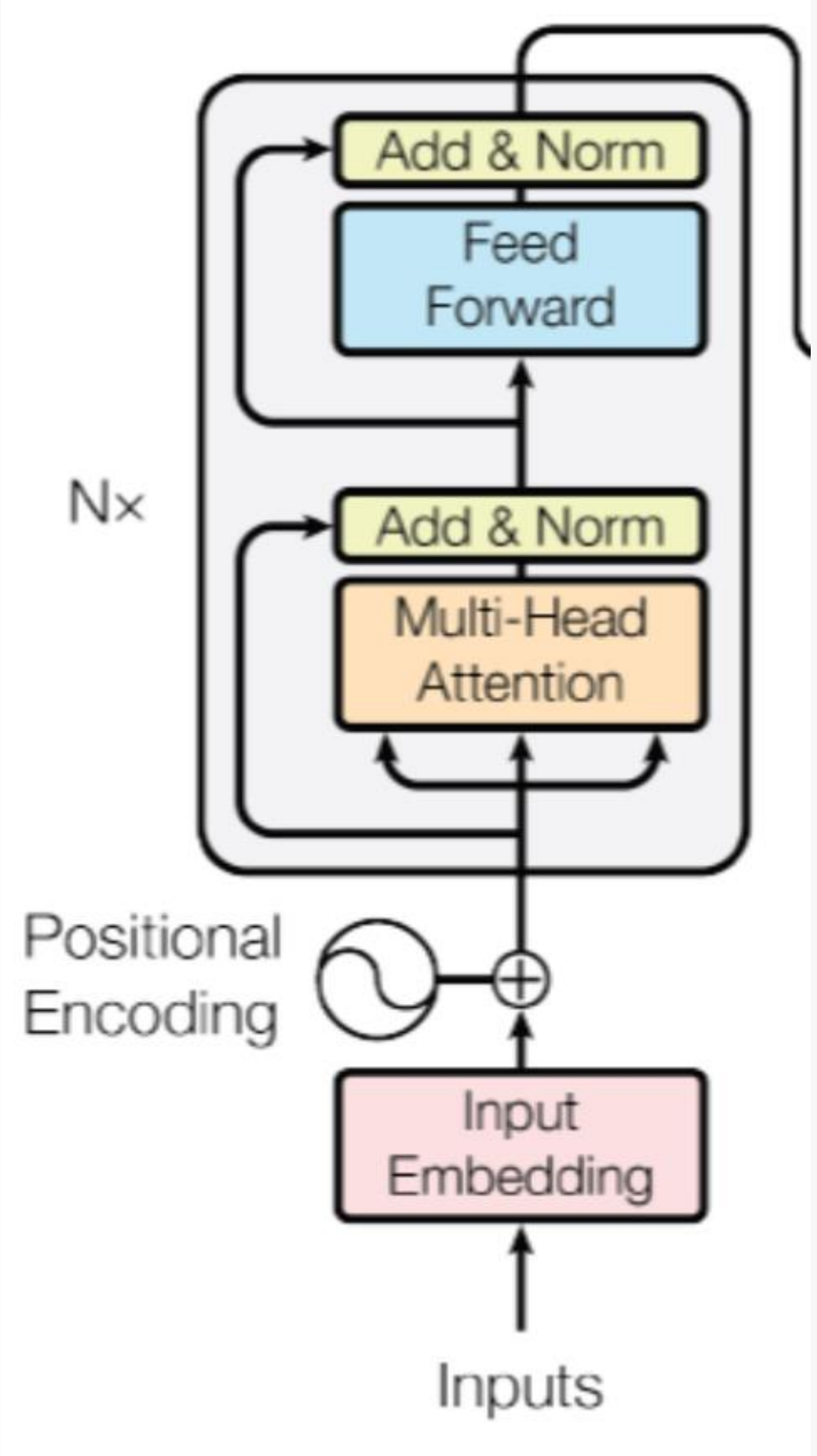
- Machine Translation (Google Search Engine,DeepL)
- Speech Recognition
- Predictive Text
- Spam filtering
- Voice assistants

Transformer



Transformer

- Transformer: A model that replaces RNN/LSTM
- Self-Attention: Dynamically calculates relationships between words in a sentence
- Multi-Head Attention: Learns various relationships through parallel processing
- Positional Encoding: Incorporates word order information
- Encoder-Decoder Structure: Handles input-output sequence processing

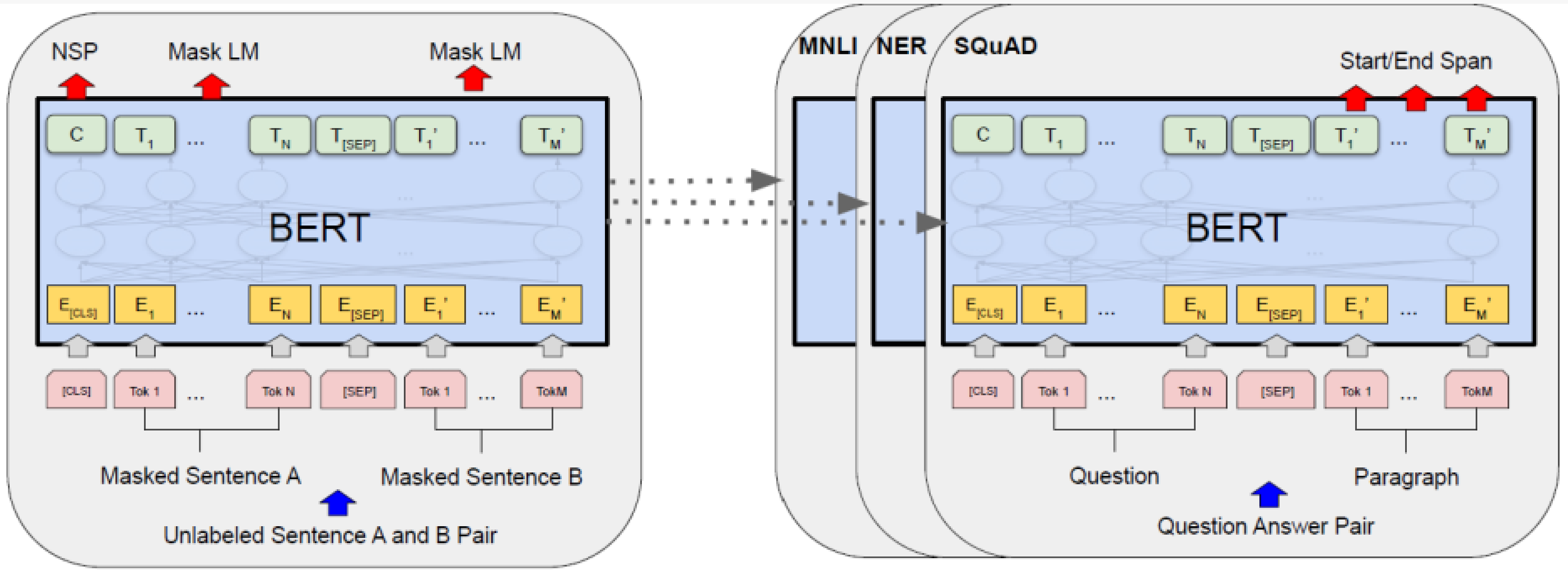


Encoder

- Input Embedding
- Positional Encoding
- Self-Attention
- Multi-Head Attention
- Feedforward Neural Network
- Layer Normalization
- Residual Connection

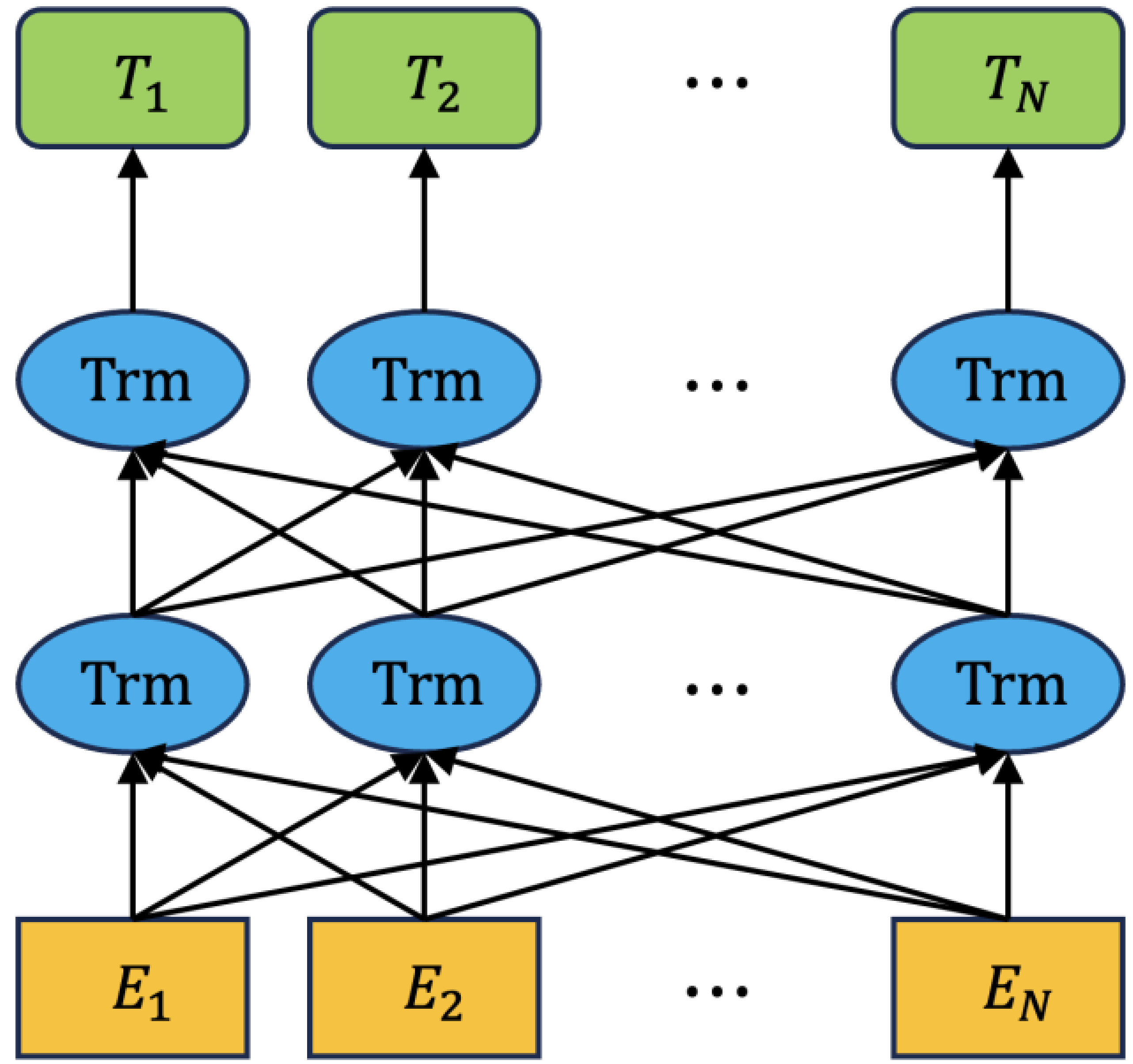
Bert Structure

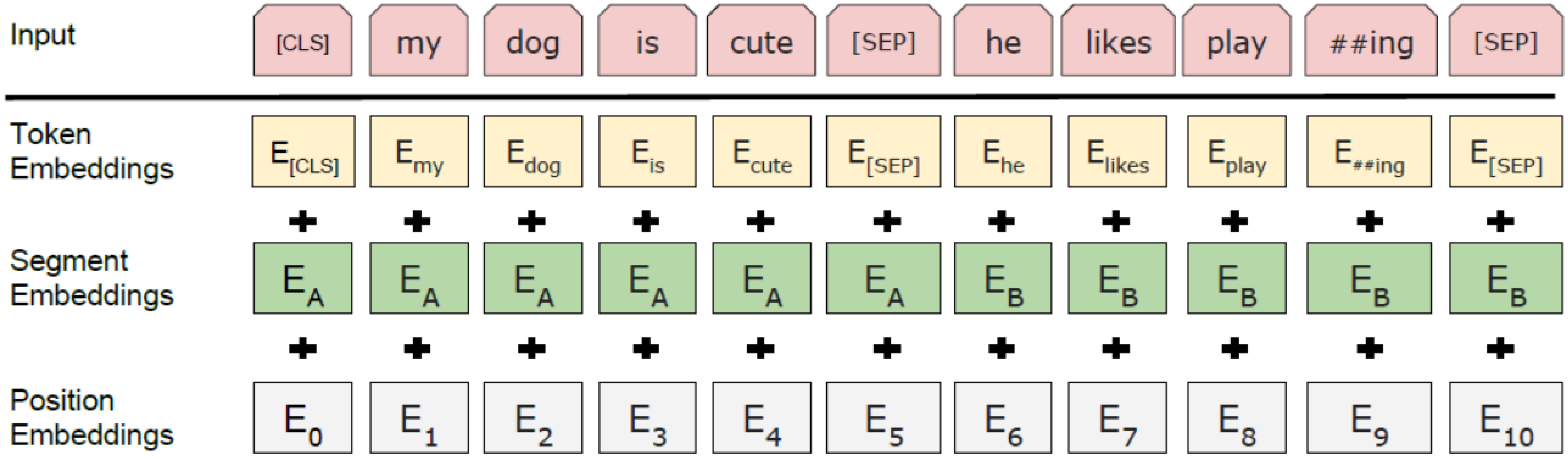
BERT Model



Pre-training

Fine-Tuning





BERT Input Representation

- Token Embedding: Token Embeddings use the WordPiece method to break words into smaller sub-words and employ [CLS] and [SEP] tokens to capture sentence-level meaning and distinguish between sentences.
- Segment Embedding: Adds unique vectors for different segments (e.g., Sentence A, Sentence B) to distinguish between them.
- Position Embedding: Represents the position of each word in a sentence to help the model understand the word order by adding position information.

Pre-Traing

MLM

Masked Language Model

1. MASKING
2. MASKED TOKEN PROCESSING
3. USING TRANSFORMER ARCHITECTURE
4. PRE-TRAINING
5. IMPROVING CONTEXT UNDERSTANDING

NSP

Next Sentence Prediciton

1. SENTENCE PAIRING
2. PREDICTION TASK
3. SENTENCE PAIR LABELS
4. TRANSFORMER STRUCTURE

TL & Fine-Tuning

TL

Transfer Learning

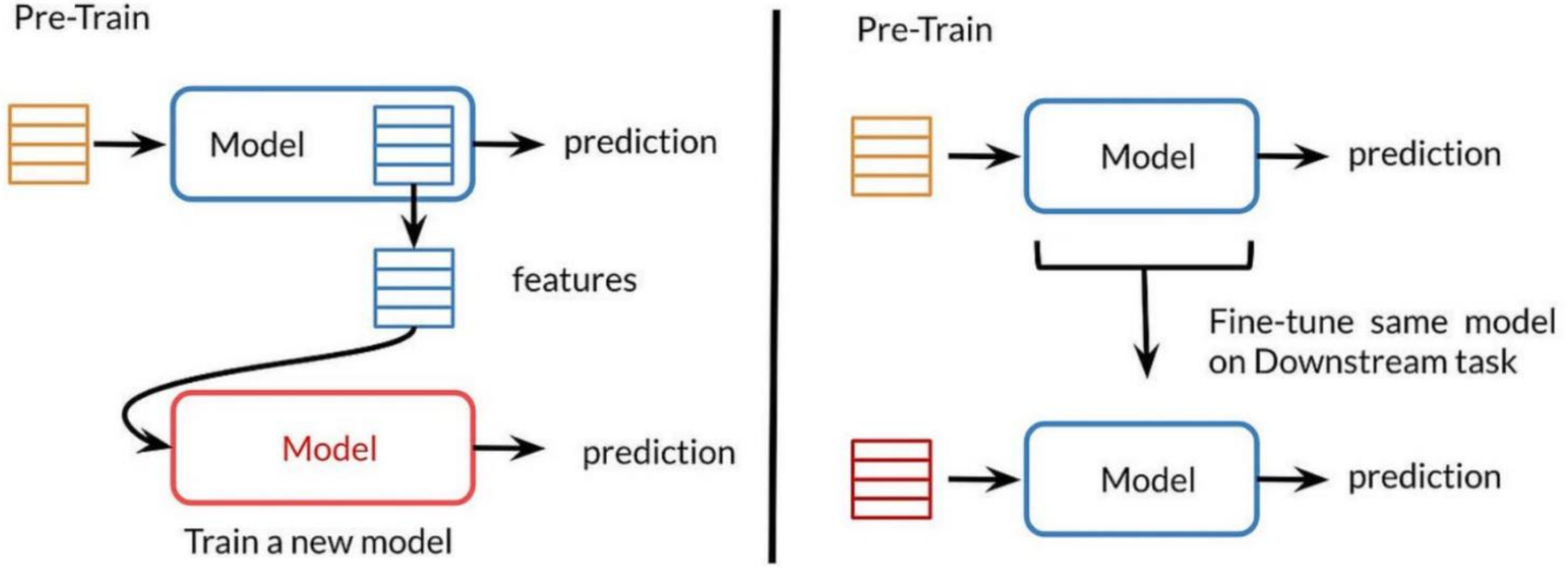
1. Models trained on large datasets can be used for learning even in areas with limited data.
2. Features learned from one model can be applied to a new model.
3. Use the existing model as a feature extractor.
4. Leverage the feature extraction capability of the existing model to easily build a new model.

Fine-Tuning

Fine-Tuning

1. Fine-tuning is the process of further training a pre-trained model to adapt it for a specific task.
2. Based on the existing model used as a feature extractor, fine-tuning involves training on a smaller domain-specific dataset to create a model optimized for that task.
3. The weights of the trained model are adjusted to adapt to the new task.

TL & Fine-Tuning



System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Table 1: GLUE Test results, scored by the evaluation server (https://gluebenchmark.com/leaderboard). The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.[8] BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

Bert Evalutaion

- GLUE
- SQuAD

System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
Published				
BiDAF+ELMo (Single)	-	85.6	-	85.8
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

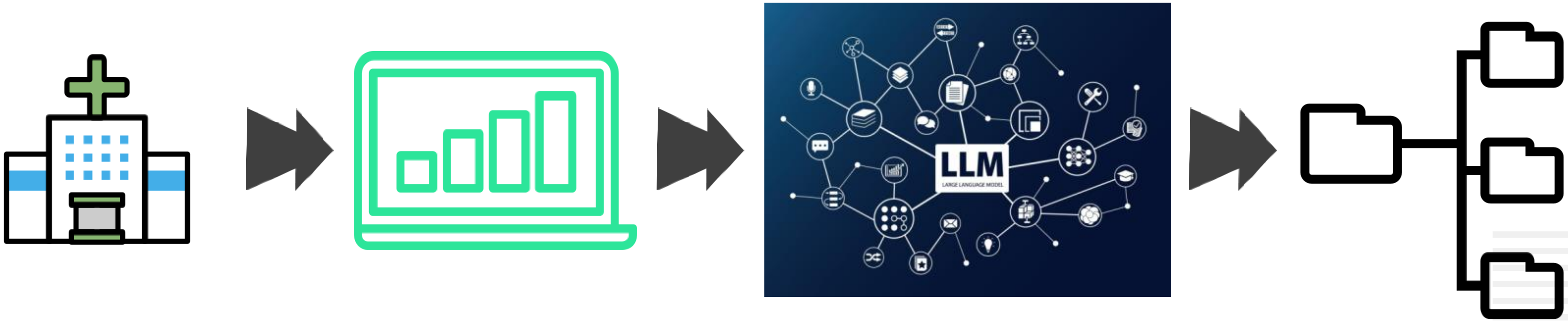
Table 2: SQuAD 1.1 results. The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds.

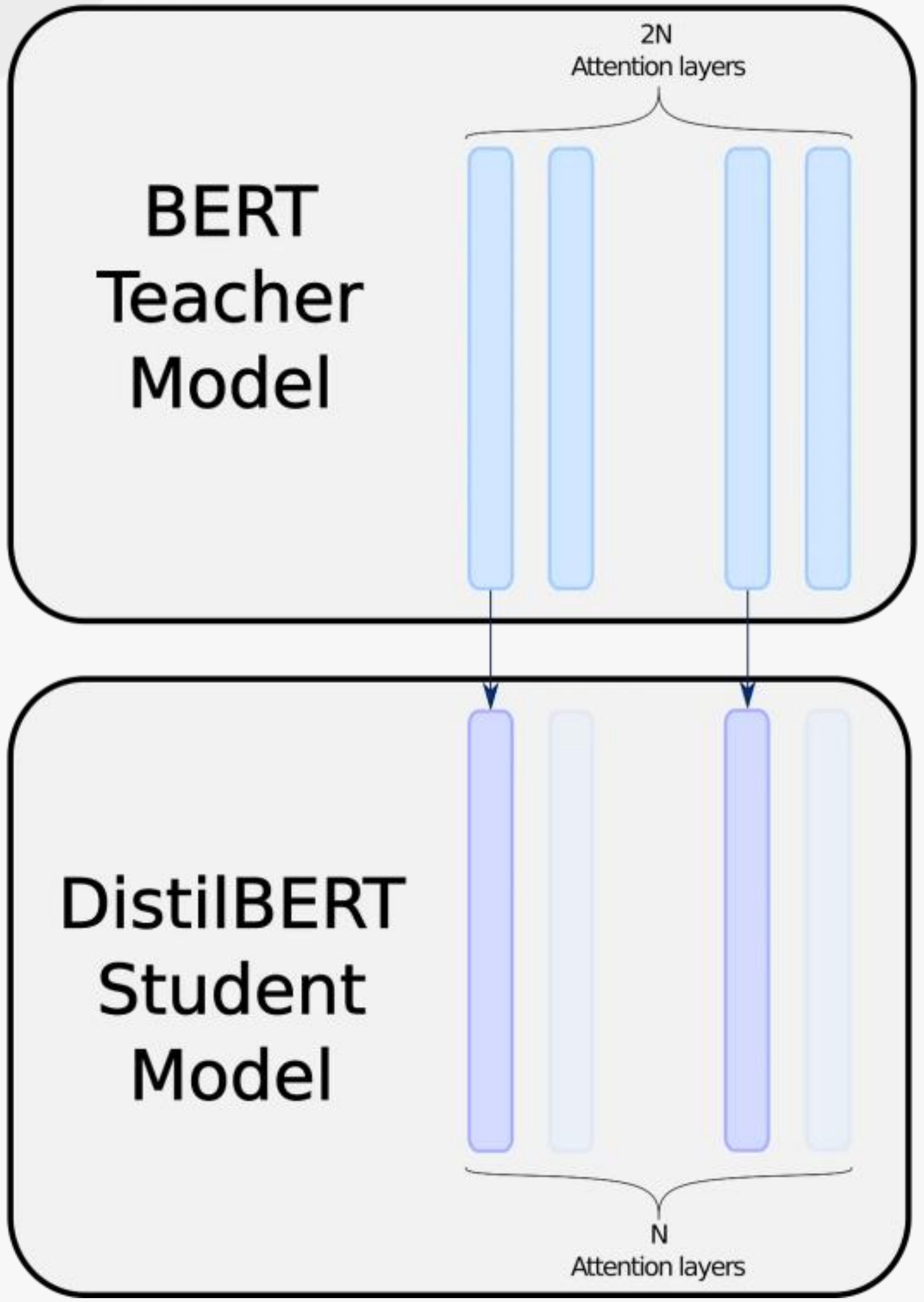
System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	86.3	89.0	86.9	89.5
#1 Single - MIR-MRC (F-Net)	-	-	74.8	78.0
#2 Single - nlnet	-	-	74.2	77.1
Published				
unet (Ensemble)	-	-	71.4	74.9
SLQA+ (Single)	-		71.4	74.4
Ours				
$\text{BERT}_{\text{LARGE}}$ (Single)	78.7	81.9	80.0	83.1

Table 3: SQuAD 2.0 results. We exclude entries that use BERT as one of their components.

Reserch Overview

BackGround





DistilBERT

- DistilBERT is 60% smaller than BERT
- Faster Processing
- Retains High Performance
- Uses Knowledge Distillation
- Lower Resource Requirements

Code Strucure

Structure(Debug)

Main_Complaints_input_ids (Inp utLayer)	[(None, 128)]	0	[]
Memory_input_ids (InputLayer)	[(None, 128)]	0	[]
Language_input_ids (InputLayer)	[(None, 128)]	0	[]
Orientation_input_ids (InputLa yer)	[(None, 128)]	0	[]
Judgment_and_Problem_Solving_i nput_ids (InputLayer)	[(None, 128)]	0	[]
Home_and_Hobbies_input_ids (In putLayer)	[(None, 128)]	0	[]
Personality_and_Behavior_input _ids (InputLayer)	[(None, 128)]	0	[]
Age (InputLayer)	[(None, 1)]	0	[]

```
[DistilBERTEmbeddingLayer] input_ids shape: [4 128]
[DistilBERTEmbeddingLayer] attention_mask shape: [4 128]
[DistilBERTEmbeddingLayer] hidden_state shape: [4 128 768]
[DistilBERTEmbeddingLayer] CLS embedding shape: [4 768]

[DistilBERTEmbeddingLayer] input_ids shape: [4 128]
[DistilBERTEmbeddingLayer] attention_mask shape: [4 128]
[DistilBERTEmbeddingLayer] hidden_state shape: [4 128 768]
[DistilBERTEmbeddingLayer] CLS embedding shape: [4 768]

[DistilBERTEmbeddingLayer] input_ids shape: [4 128]
[DistilBERTEmbeddingLayer] attention_mask shape: [4 128]
[DistilBERTEmbeddingLayer] hidden_state shape: [4 128 768]
[DistilBERTEmbeddingLayer] CLS embedding shape: [4 768]
```

```
Debug [Personality_and_Behavior_bert_out] shape: [4 768]
Debug [Home_and_Hobbies_bert_out] shape: [4 768]
Debug [Judgment_and_Problem_Solving_bert_out] shape: [4 768]
Debug [Orientation_bert_out] shape: [4 768]
Debug [Language_bert_out] shape: [4 768]
Debug [Memory_bert_out] shape: [4 768]
Debug [Main_Complaints_bert_out] shape: [4 768]
Debug [Age_num_in] shape: [4 1]
```

Code-Conclusion

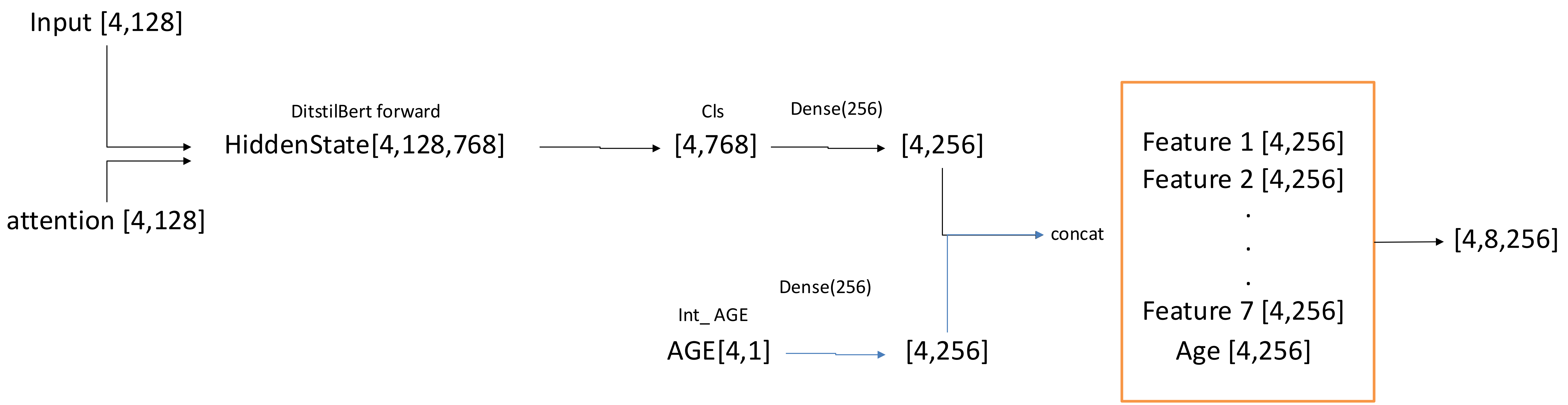
```
Epoch 1/10
1/4 [======>.......................] - ETA: 1:14 - loss: 0.9355 - categorical_accuracy: 0.5000 - 2/4
[==============>..............] - ETA: 1s - loss: 0.9936 - categorical_accuracy: 0.4375 - pr3/4
[=====================>........] - ETA: 0s - loss: 0.8326 - categorical_accuracy: 0.4583 -
pr4/4 [==============================] - ETA: 0s - loss: 0.7897 - categorical_accuracy:
0.4688 - precision_2: 0.5000 - precision_3: 0.3125 - recall_2: 0.4958 - recall_3: 0.4375
model save [True, False]: [loss: 0.643076]    [f1: 0.000000 --> 0.000000]    [f1_0: 0.000000 -->
0.000000]    [f1_1: 0.000000 --> 0.000000]    [pr0: 0.000000 --> 0.000000]    [pr1: 0.000000 -->
0.000000]
4/4 [==============================] - 34s 3s/step - loss: 0.7897 -
categorical_accuracy: 0.4688 - precision_2: 0.5000 - precision_3: 0.3125 - recall_2: 0.4958 -
recall_3: 0.4375 - val_loss: 0.6431 - val_categorical_accuracy: 0.5500 - val_precision_2: 0.5714
- val_precision_3: 0.5385 - val_recall_2: 0.4000 - val_recall_3: 0.7000

Epoch 10/10
1/4 [======>.......................] - ETA: 1s - loss: 6.1612e-04 - categorical_accuracy: 0.8967 2/4
[==============>..............] - ETA: 1s - loss: 5.0488e-04 - categorical_accuracy: 0.8978 3/4
[=====================>........] - ETA: 0s - loss: 6.1528e-04 - categorical_accuracy: 0.8989
4/4 [==============================] - ETA: 0s - loss: 5.0549e-04 -
categorical_accuracy: 0.8999 4/4 [==============================] - 3s 719ms/step -
loss: 5.0549e-04 - categorical_accuracy: 0.8999 - precision_2: 0.8963 - precision_3: 0.9037 -
recall_2: 0.9057 - recall_3: 0.8941 - val_loss: 0.2554 - val_categorical_accuracy: 0.9000 -
val_precision_2: 0.9000 - val_precision_3: 0.9000 - val_recall_2: 0.9000 - val_recall_3: 0.9000
```

```
accuracy                         1.00       4
macro avg       1.00      1.00      1.00       4
weighted avg       1.00      1.00      1.00       4

Confusion matrix, without normalization
[[2 0]
[0 2]]
```

CodeStructure



CodeStructure

MultiHeadAttention

Layer Nomerization

$[4,8,256] \xrightarrow{\text{Flatten}} [4,2048] \xrightarrow{\text{Fully Connected}} [4,64] \xrightarrow{\text{Fully Connected}} [4,16] \xrightarrow{\text{Fully Connected}} [4,2]$

fedtherapist: mental health monitoring with user-generated linguistic expressions on smartphones via federated learning