BERT and Its Variants

AI with Deep Learning

EE4016

Prof. Lai-Man Po

Department of Electrical Engineering
City University of Hong Kong

https://medium.com/@lmpo/from-static-to-contextual-the-bert-revolution-in-nlp-46732eb7abf0

EE4016 Mid-Term Exam (Week 11)

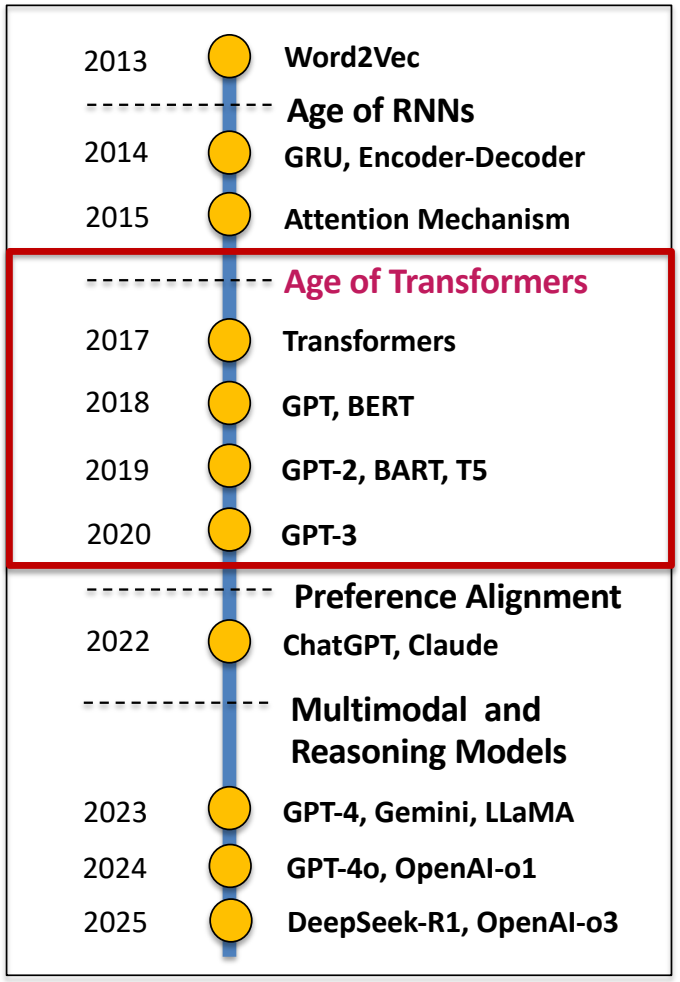
- The Mid-Term Exam will be held on **Monday of Week 11, which falls on March 30, 2026**. Here is important information about the mid-term exam:
  - **The mid-term exam will begin at 1:50pm and conclude at 2:50pm**, lasting for 1 hour.
  - Similar to the quiz, it is a semi-open book exam:
    - The first 40 minutes will be closed book. Students are not allowed to use any materials except an electronic calculator.
    - The last 20 minutes will be open book. You are permitted to use your devices to access the course PPT lecture notes and Medium Articles.
  - Students may utilize physical copies of handouts or electronic devices such as smartphones, tablets, iPads, or laptops to access their notes. However, electronic devices must be set to airplane mode during the exam. Investigators will periodically check compliance with this requirement.
  - Communication with others and the use of ChatGPT or any other language model services during the exam are strictly prohibited. Investigators will conduct periodic checks to ensure that students adhere to these rules throughout the exam.
  - Students are responsible for bringing their own answer sheet, preferably in A4 size.
  - The exam questions will follow a similar style to Assignment 2. It will consist of approximately 20 short questions covering topics such as MLPs, CNNs, Word2Vec, RNNs, and Transformers. Additionally, there may be some long questions focusing specifically on MLPs, CNNs and RNNs and Transformers.

Content

1. **Types of Transformer-based Models**
   - **Encoder-Only, Decoder-Only and Encoder-Decoder**
2. **BERT** (**B**i-directional **E**ncoder **R**epresentation from **T**ransformer)
3. **BERT Pretraining, Fine Tuning (3 pass explanation)**
4. **BERT's variants – RoBERTa, DistillBERT, ALBERT, etc**

LLMs: From Word2Vec to DeepSeek-R1 (2012-2025)

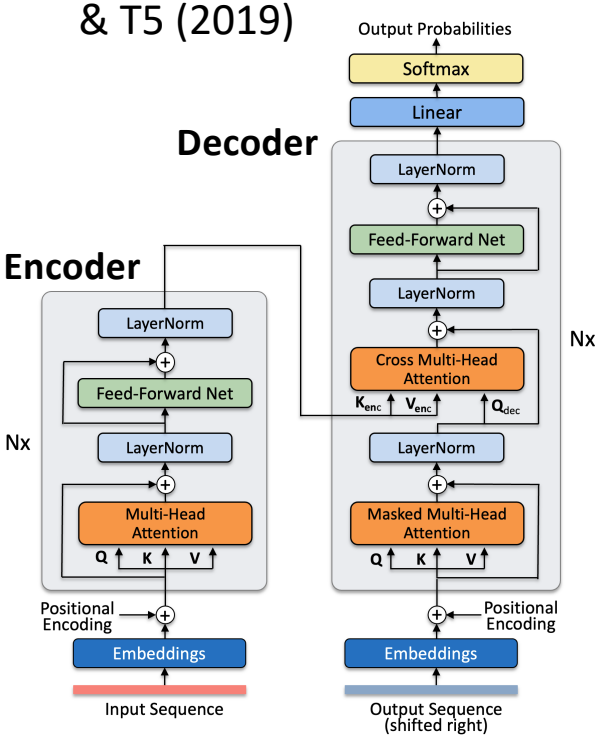
1. Tokenization and Word2Vec: BPE, CBOW, Skip-Gram
2. RNNs, LSTM, GRU, Seq-to-Seq (Encoder-Decoder) and Attention Mechanisms
3. Transformers with Self-Attention
4. Lager Language Models (LLMs): BERT, GPT, BART, T5
5. Preference Alignment by SFT and RLHF: ChatGPT, Claude, LLaMA
6. Multimodal Models: GPT-4, GPT-4o, Gemine, LLaVA
7. Reasoning Models: OpenAI-o1, DeepSeek-R1



The Three Ways to Build a Transformer

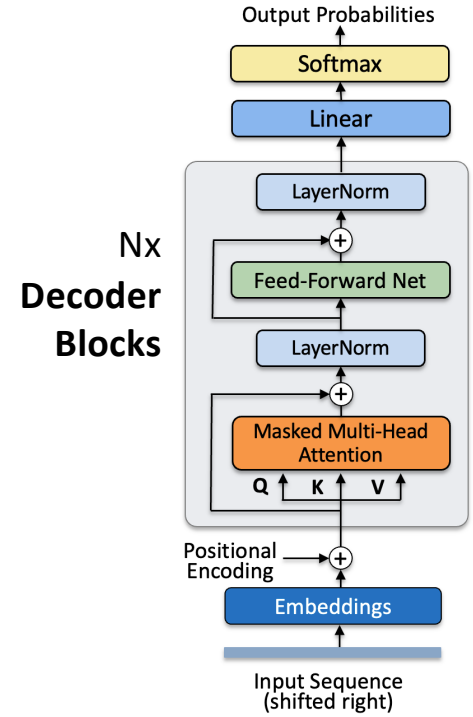
Encoder-Decoder

- Seq-to-Seq
- Function: Translation
- Examples: Transformer (2017) & T5 (2019)



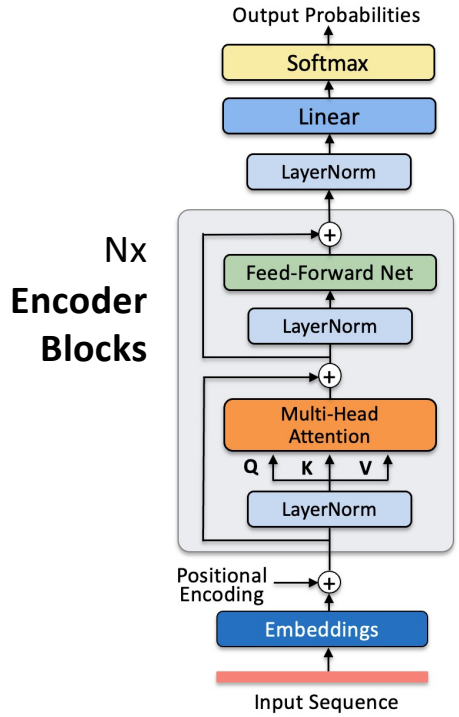
Decoder-Only

- Auto-regressive
- Function: Generation
- Example: GPT (2018-06)

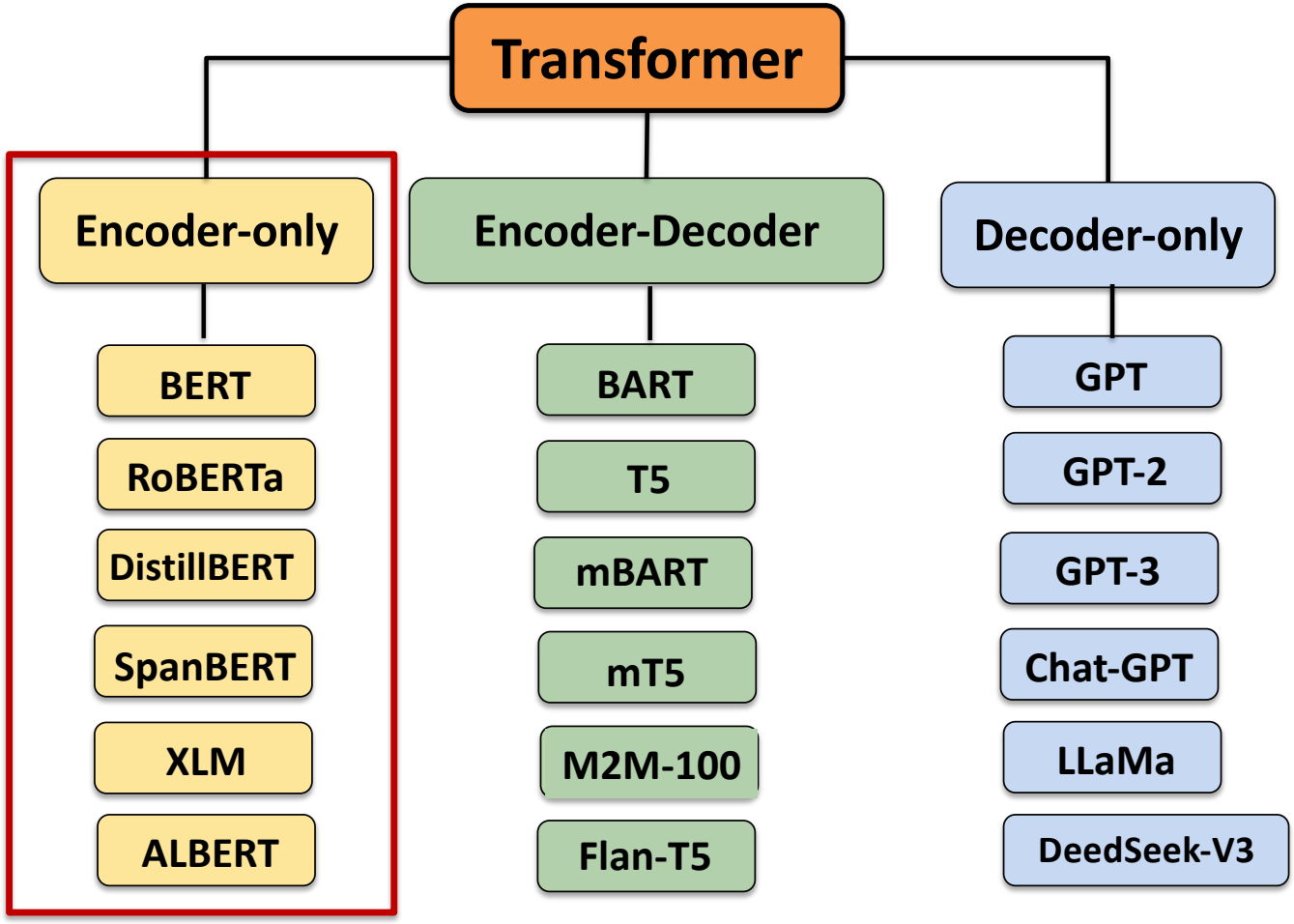


Encoder-Only

- Auto-encoding
- Function: Understanding
- BERT (2018-10)



Well-known Transformer-based Models



Transformer-based Language Models

- **Encoder-only Transformer Models**
  - **Bidirectional context**: Good for classification, language understanding
  - **BERT Series**: **BERT** (2018-10), **RoBERTa**, **DistillBERT**, **XLM**, **SpanBERT**, **ALBERT**
- **Decoder-only Transformer Models**
  - **Autoregressive Language Modeling**: Good for text generation
  - **GPT Series**: **GPT-1** (2018-6), **GPT-2** (2019-02), **GPT-**3 (2020-06), **InstructGPT** (2022-02), **ChatGPT** (2022-11), **GPT-4** (2023-03), **LLaMA** (2023-02), **DeekSeek-V3** (2024-12)
- **Encoder-Decoder Transformer Models**
  - **Sequence-to-Sequence Models**
  - Original **Transformer** (2017-06), **T5** (2019-10), **BART** (2019-11), **BigBird** (2012-06), **M2M-100** (2021-07), **Flan-T5** (2022-10)

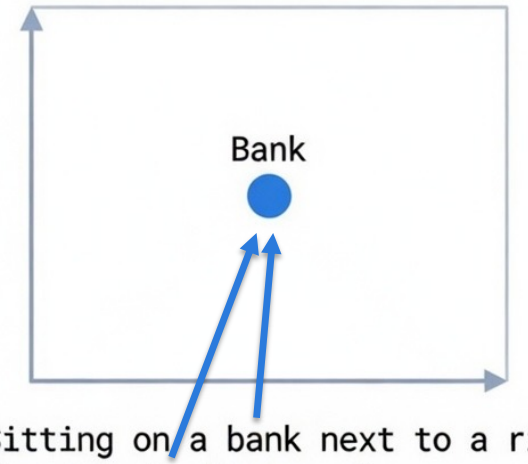
BERT

Bidirectional Encoder Representation from Transformer

The Polysemy Problem: One Word, Multiple Meanings

Why Context is King

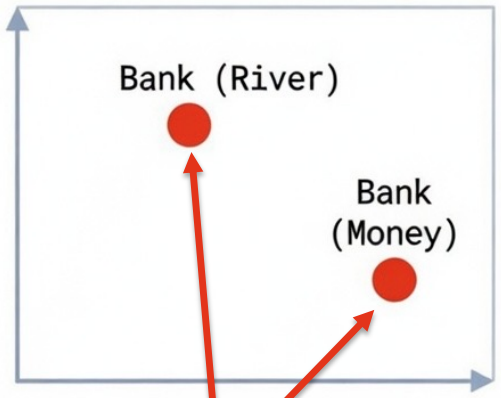
Static Embeddings / Word2Vec



1. Sitting on a bank next to a river.
2. My empty bank account.

One fixed vector for every instance.
Ambiguity fails.

Contextual Embeddings / BERT



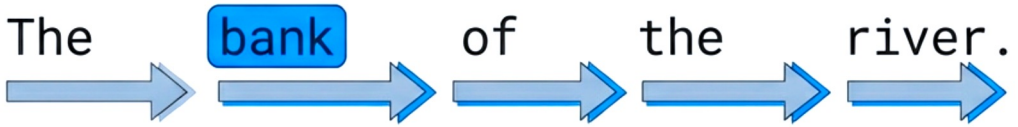
1. Sitting on a bank next to a river.
2. My empty bank account.

Vector changes based on surrounding words. Ambiguity solved.

Breaking the Linear Barrier

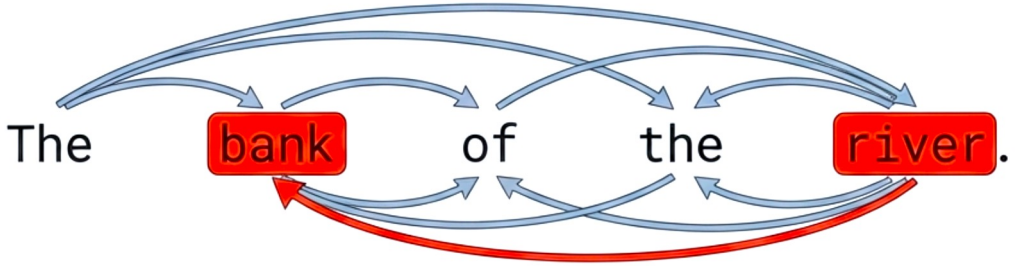
Prior to 2018, NLP models were **unidirectional**. They read text sequentially, limiting their ability to understand context based on future words. **ELMo** and **BERT** introduced bidirectionality, allowing the model to see the entire sentence at once.

Unidirectional



Constraint: Can only see **past context**.

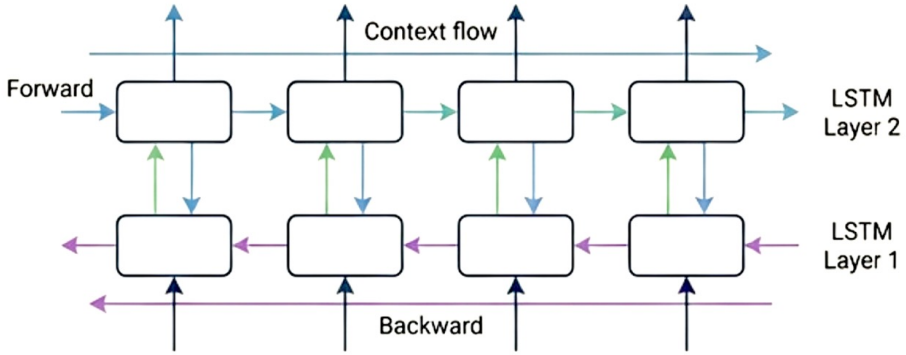
Bidirectional



Advantage: See **river** simultaneously to define **bank**.

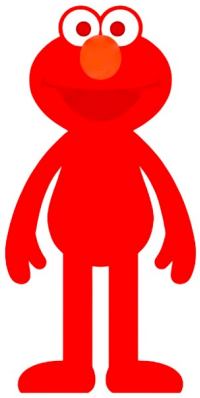
ELMo (2018-02): The Bridge to Contextual Understanding

Embeddings from Language Models (**ELMo)** learns **Contextualized Word Representations based on** two Bi-LSTMs layers.

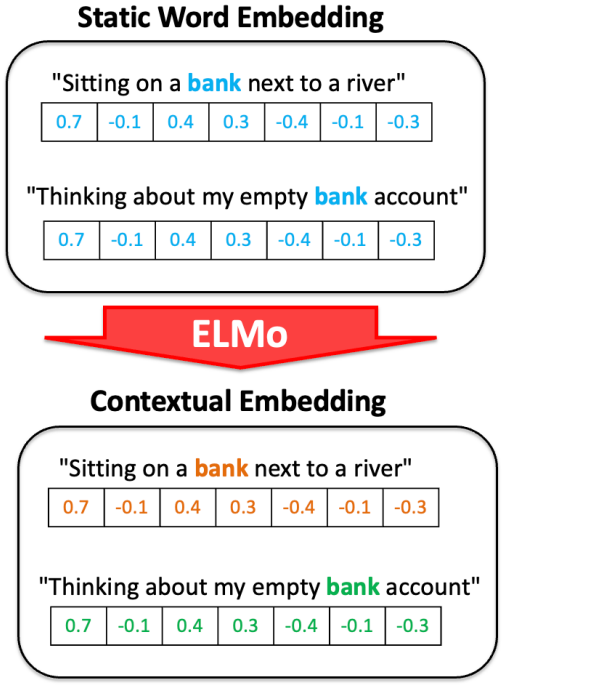


Bi-directional LSTM

The Bridge to Context: ELMo finally distinguished between 'River Bank' and 'Money Bank' using bidirectional LSTMs.

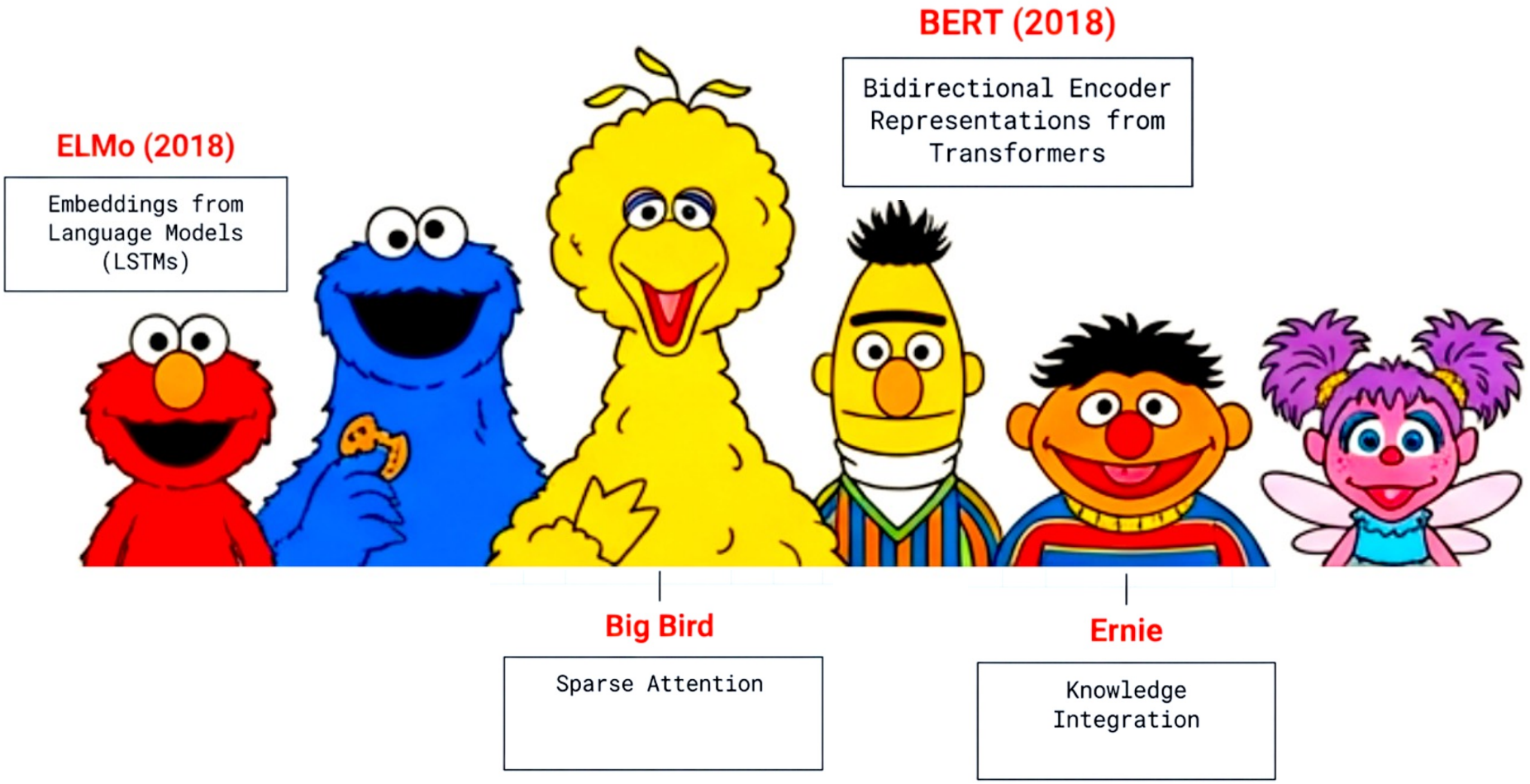


ELMo



The vector for “Bank” now changes based on its neighbors.

The 'Sesame Street' Era of NLP

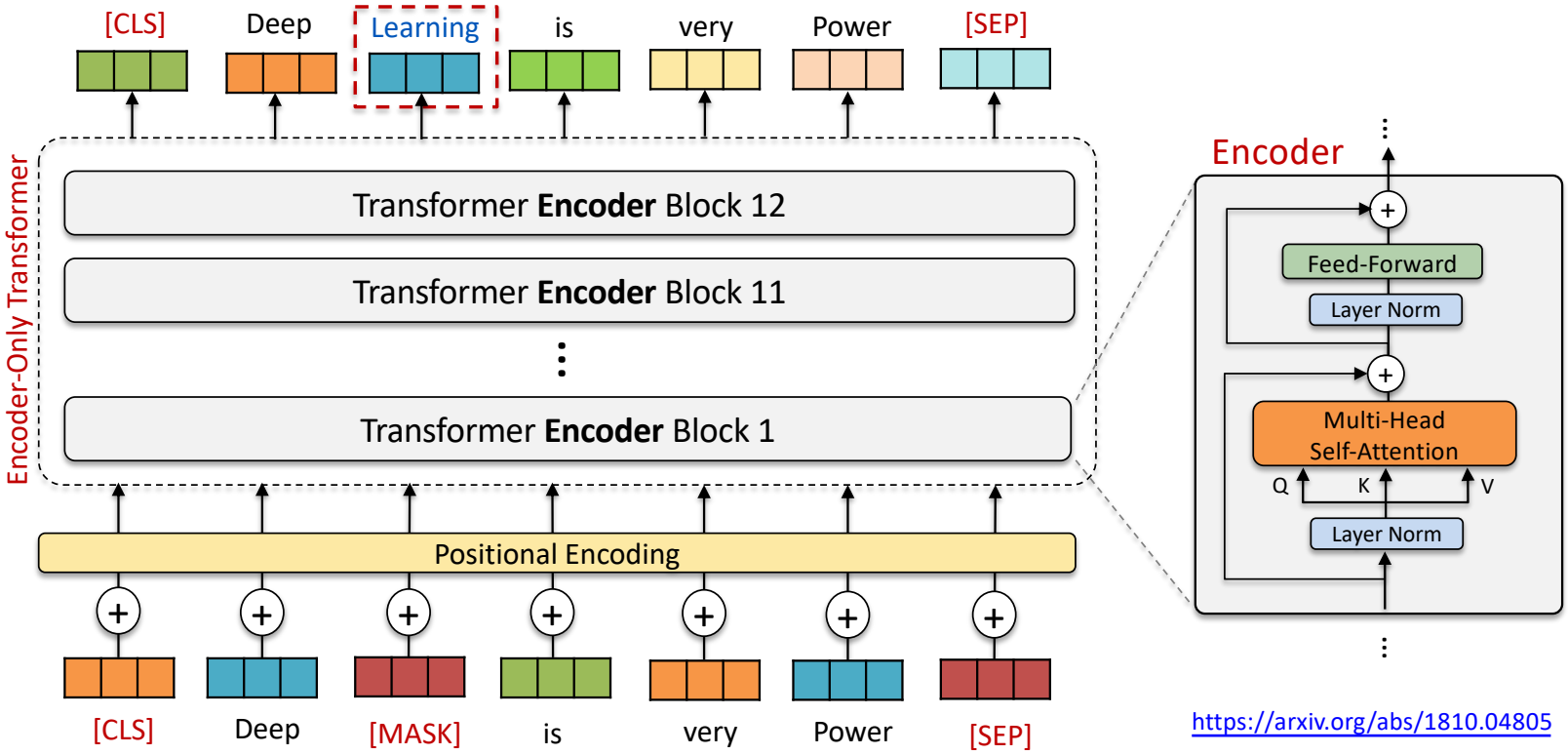


Google's BERT (2018-10)

Bidirectional **E**ncoder **R**epresentations from **T**ransformer (**BERT**)

- BERT is the **first encoder-only Transformer model**, capable of learning from large amounts of unlabeled text.

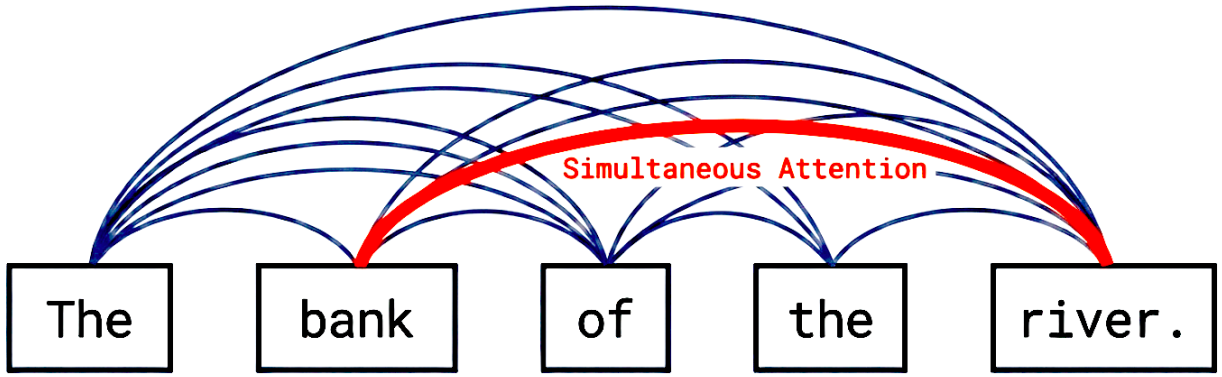
BERT



https://arxiv.org/abs/1810.04805

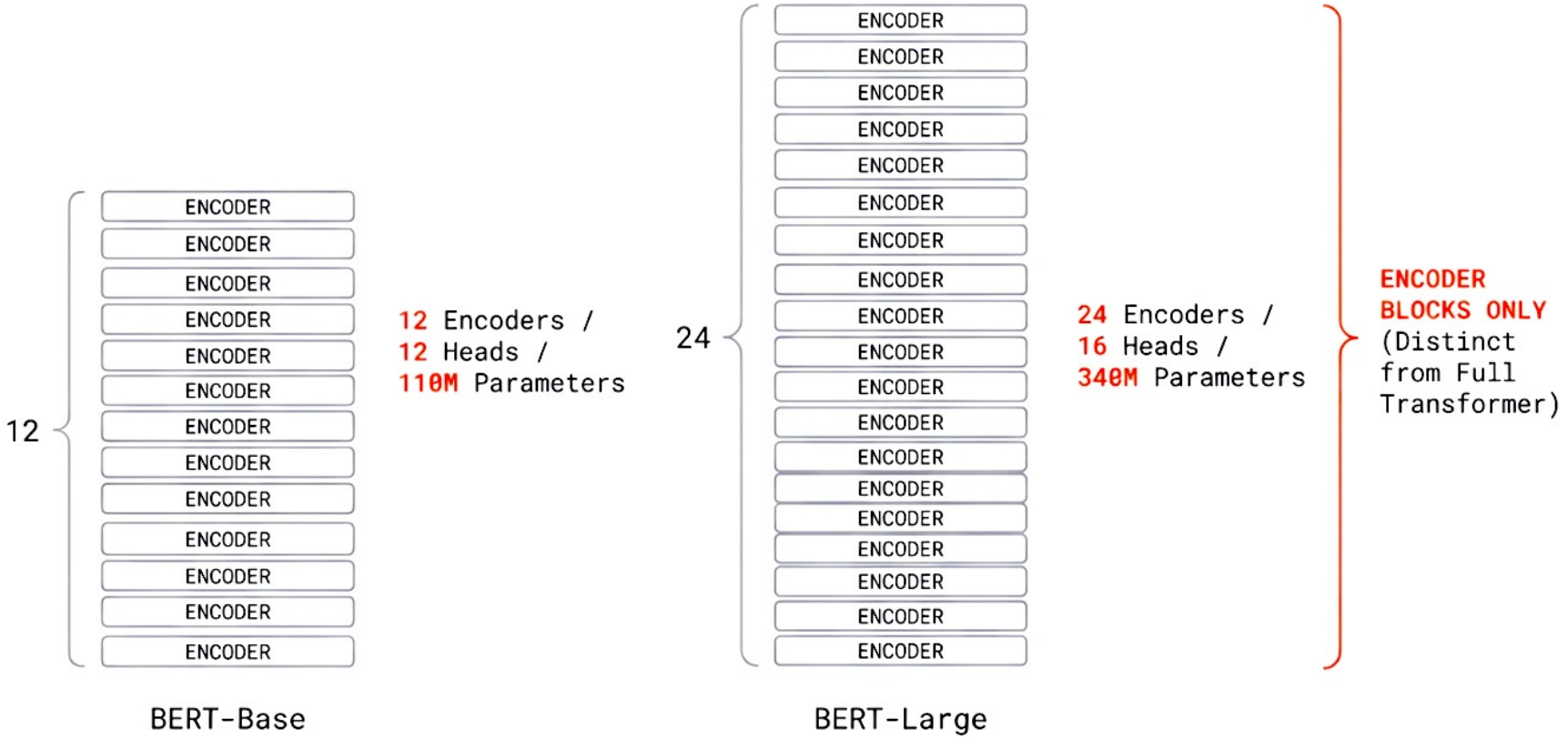
BERT: Bidirectional Encode Representations

- Similar to ELMo, BERT provides **contextualized word representations** to address polysemy, but it uses Transformers instead of LSTMs.



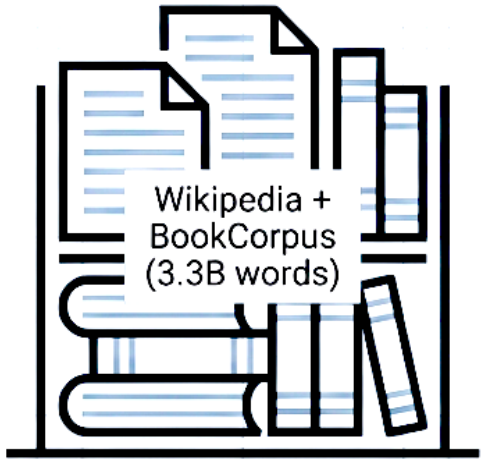
"Deep bidirectionality allows the model to see the future and the past simultaneously."

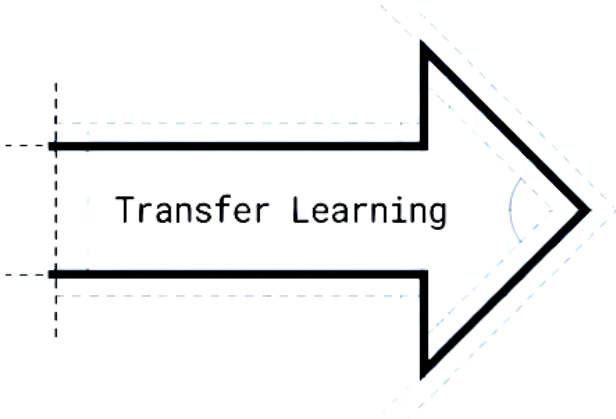
The Architecture: Stacking the Encoders



The Two-Stage Paradigm: Pre-training & Fine-tuning

Stage 1: Pre-training





Stage 2: Fine-tuning

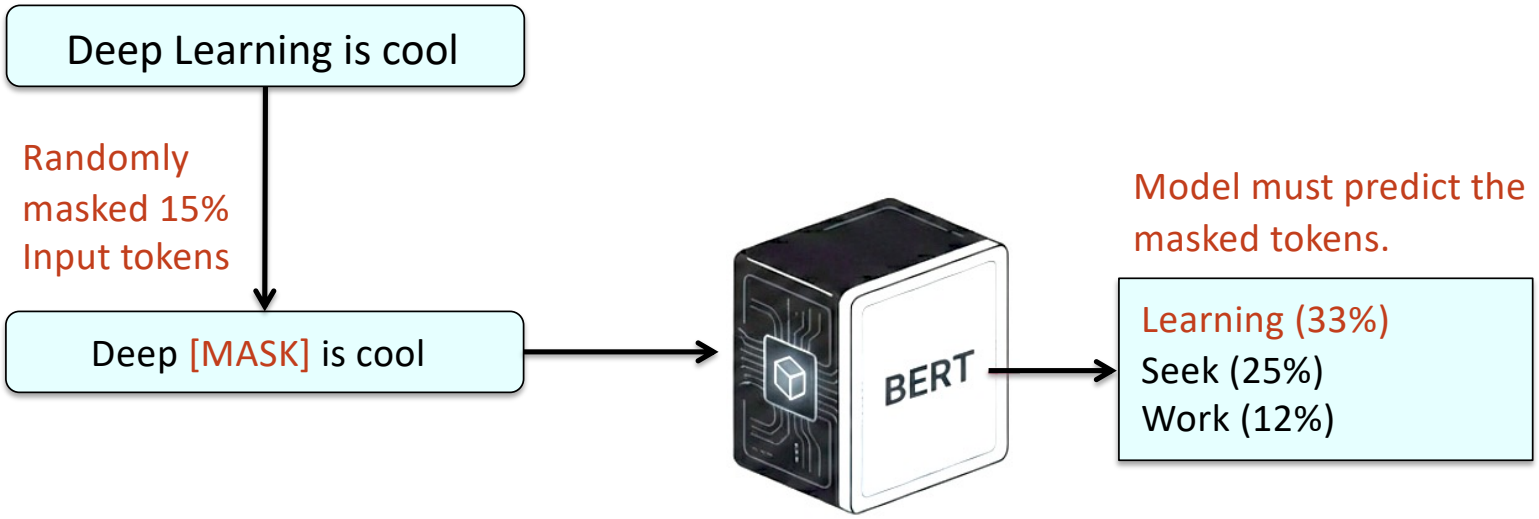
Goal: **Self-supervised learning** of language structure (Unlabeled Data).

- Wikipedia (2.5 billion words) + BookCorpus (800 millions words)

Goal: **Supervised learning** for Task-specific refinement (Labeled Data).

Pre-Training Objective 1: Masked Language Model (MLM)

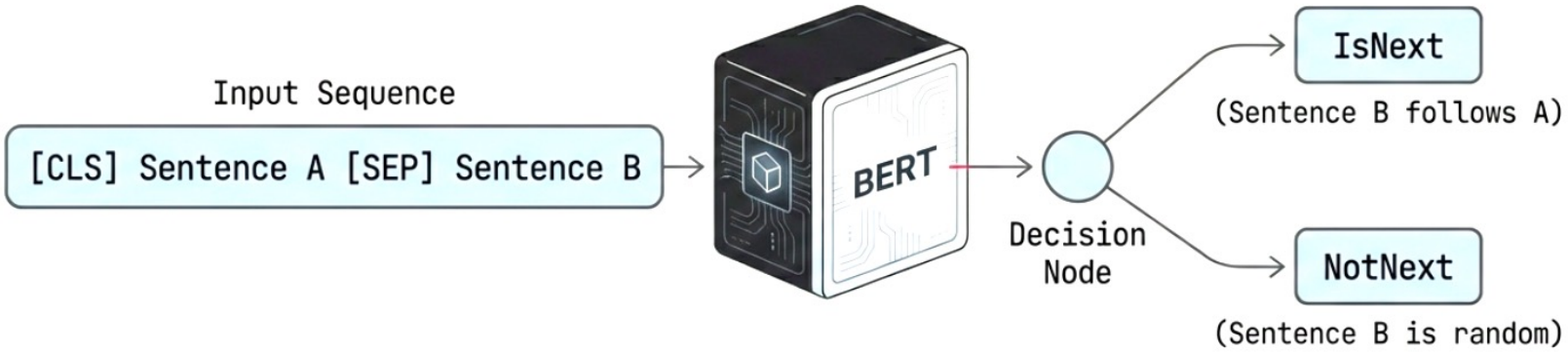
The Cloze Task (填空)



We hide 15% of the words. To fill the blank, the model MUST understand the bidirectional context.

Pre-Training Objective 2: Next Sentence Prediction (NSP)

- Use two special tokens of [CLS] as a classification token and [SEP] as a separation token for two input sentences
- Use the embedding of the [CLS] to predict whether the second sentence follows the first sentence



Goal: Understand the relationship between sentences.

- True Pair: [A] He went to the store. [B] He bought milk.
- False Pair: [A] He went to the store. [B] Penguins imply flight.

The Anatomy of an Input

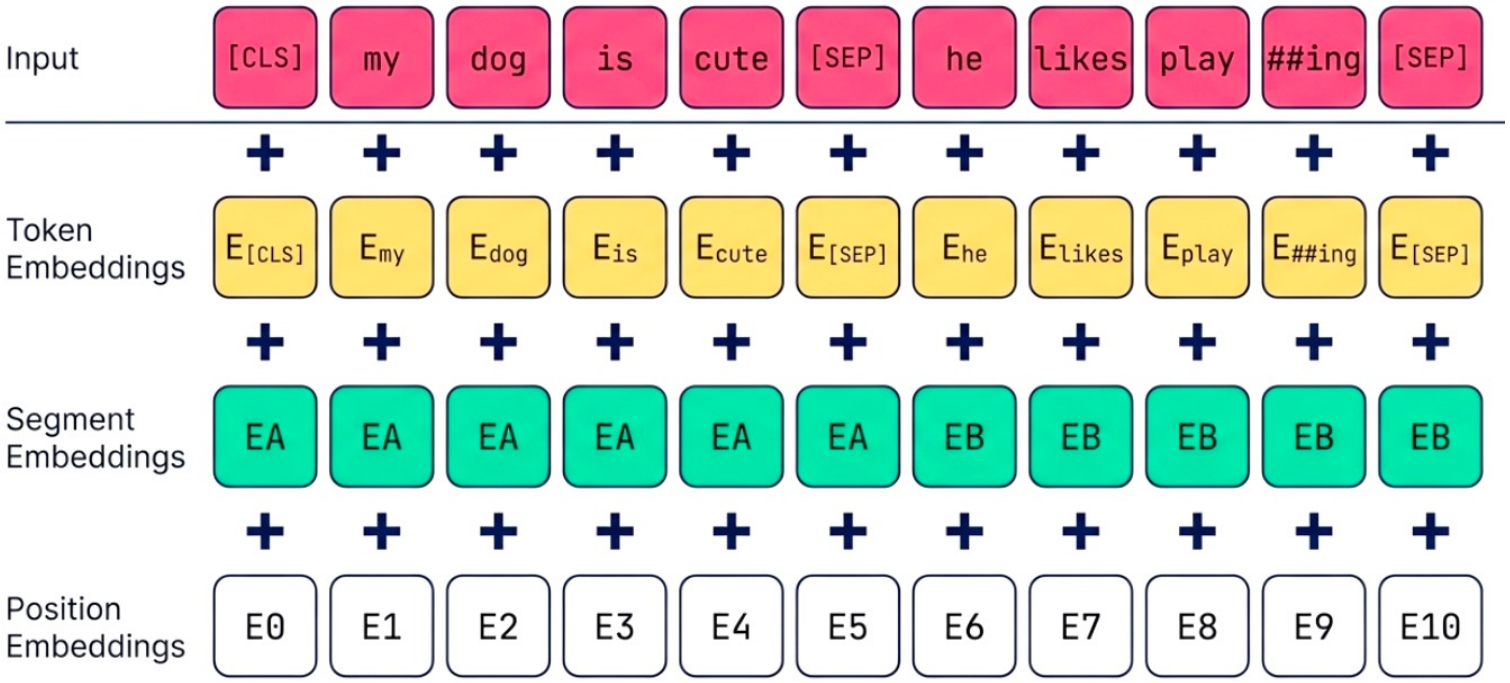
BERT reads "WordPieces" rather than full words. It uses a 30,000-token vocabulary and specific control characters to structure the input.

- **[CLS]** Classification Token:
  - Added to the start of every sequence. Holds the aggregate representation.
- **[SEP]** Separator Token:
  - Marks the boundary between sentence A and sentence B.
- **[MASK]** Mask Token:
  - Used during training to hide words for prediction.

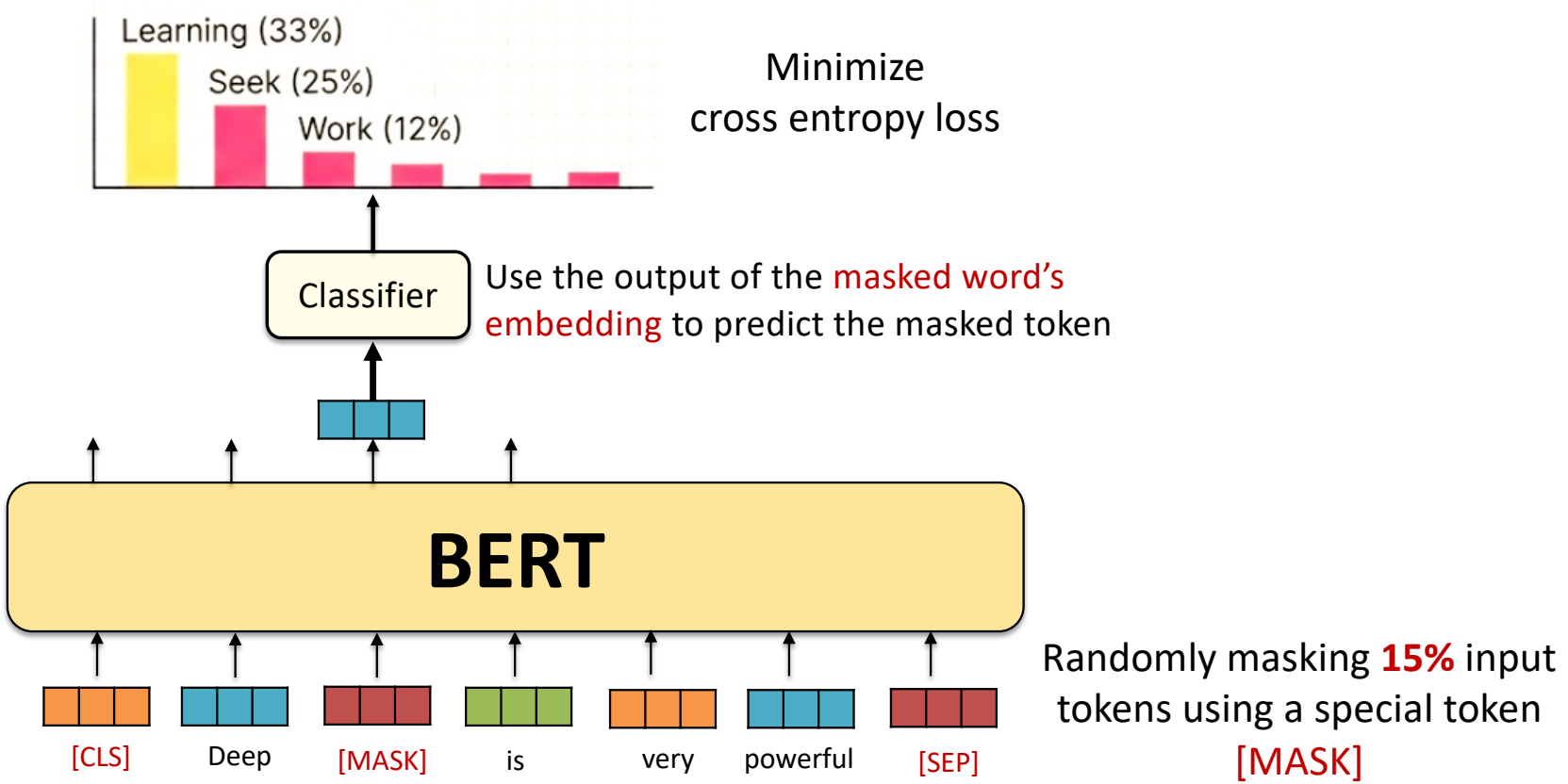
BERT requires these structural cues to process raw text into mathematical vectors.

The Three-Layer Embedding Stack

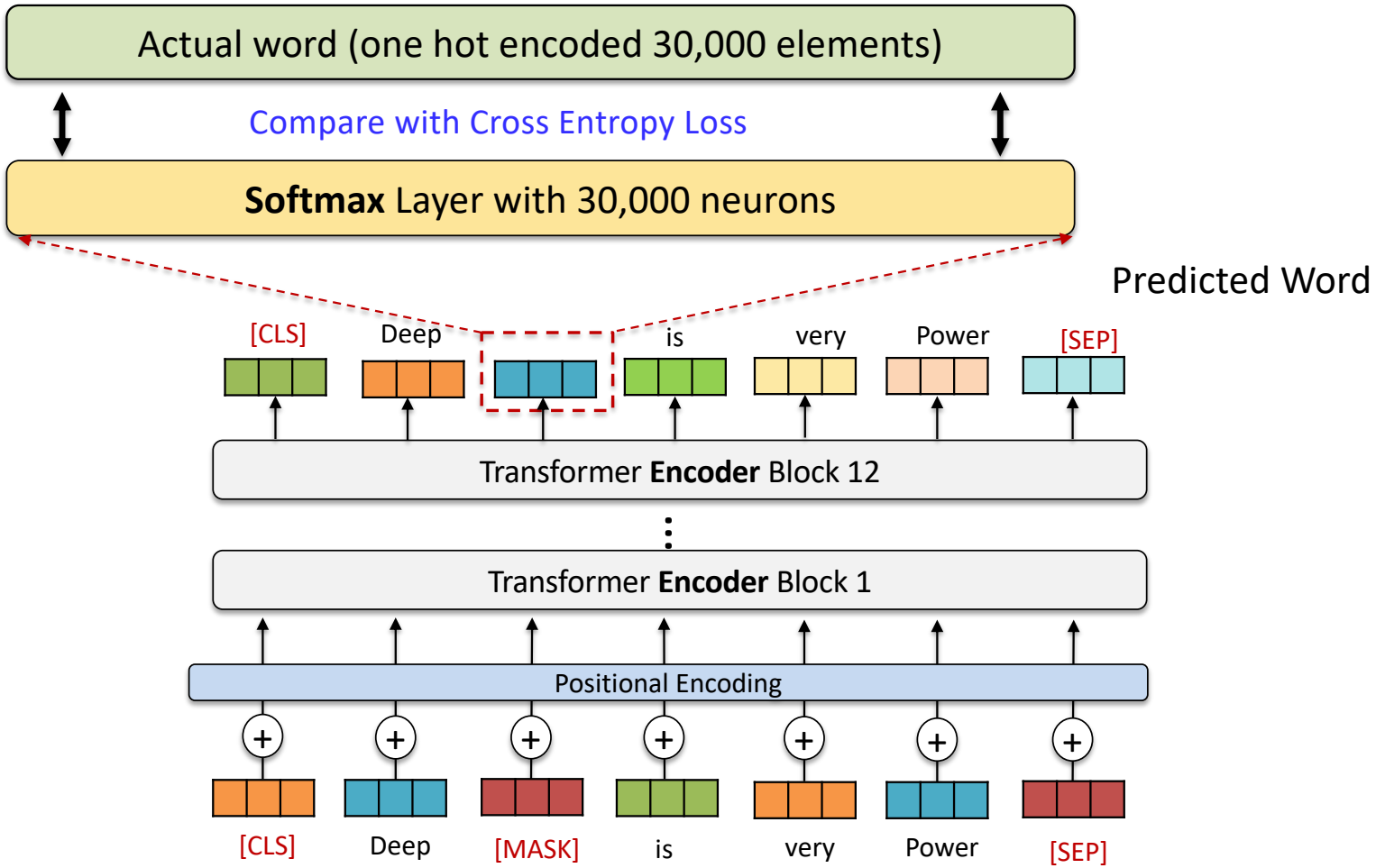
- Input Representation = Token + Segment + Position



Masked Language Model (MLM) Per-training



BERT's Output



Next Sentence Prediction (NSP) Per-training

- **NSP** is used to model relationship between sentences
  - Question Answering, Natural Language Inference etc. are based on **understanding inter-sentence relationship**

Input: "[CLS] calculus is a branch of math [SEP] it was developed by Newton and Leibniz"

Label: IsNext (or True)

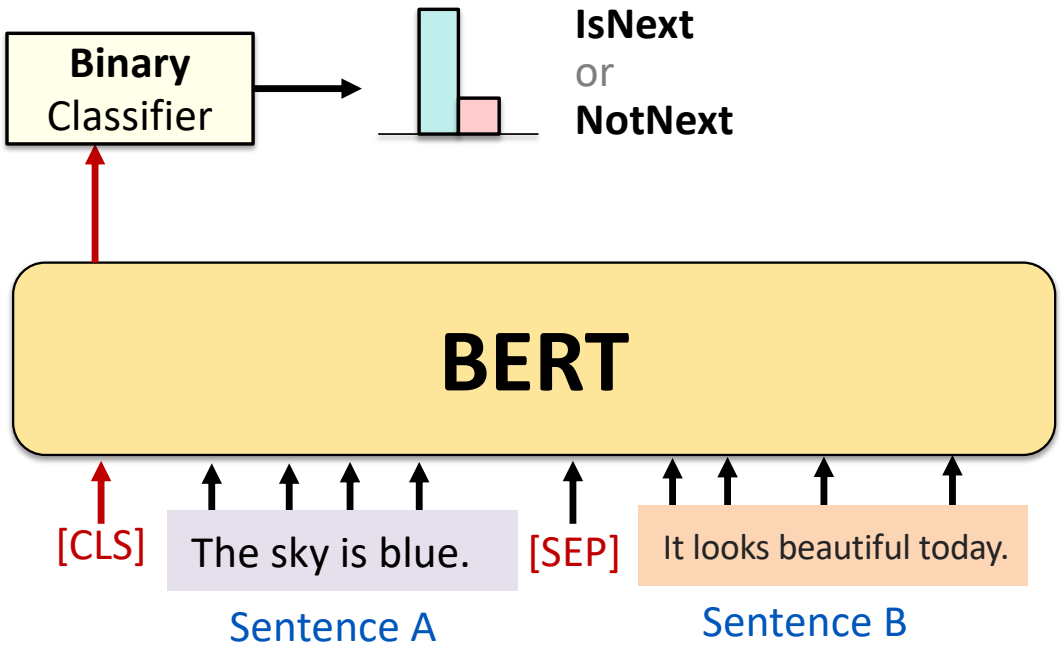
Input: "[CLS] calculus is a branch of math [SEP] panda is native to south central China"

Label: NotNext (or false)

Next Sentence Prediction (NSP) Data Labeling

NSP models **relationship** between 2 sentences by forming the input with two special tokens:

- [CLS] : the position that outputs classification results
  - **IsNext** or **NotNext**
- [SEP] : the boundary of two sentences



Combining MLM and NSP Methods

- **Input:** "[CLS] calculus is a [MASK] of math [SEP] it [MASK] developed by Newton and Leibniz"
  - **Target:** True, "branch", "was"

- **Input:** "[CLS] calculus is a branch of math [SEP] panda is native to [MASK] central China"
  - **Target:** Fales, "south"

BERT: Pre-training Cost Function

- **The BERT pre-training cost function is the sum of the two loss functions:**
  - **[MASK]** : $\mathcal{L}_{\text{MLM}}$ is the loss for multi-class classification (i.e., predicting the masked words)

$$\mathcal{L}_{\text{MLM}} = -\sum_{i \in \text{Masked}} \log P(w_i | w_{\text{context}})$$

where w_i is the masked token, and $w_{context}$ represents the surrounding tokens.

  - **[CLS]** : Loss 1 is for binary classification (i.e., predicting the next sentence.)

$$\mathcal{L}_{\text{NSP}} = -\log P(\text{IsNext} \mid S_A, S_B)$$

where S_A and S_B are the two sentences.

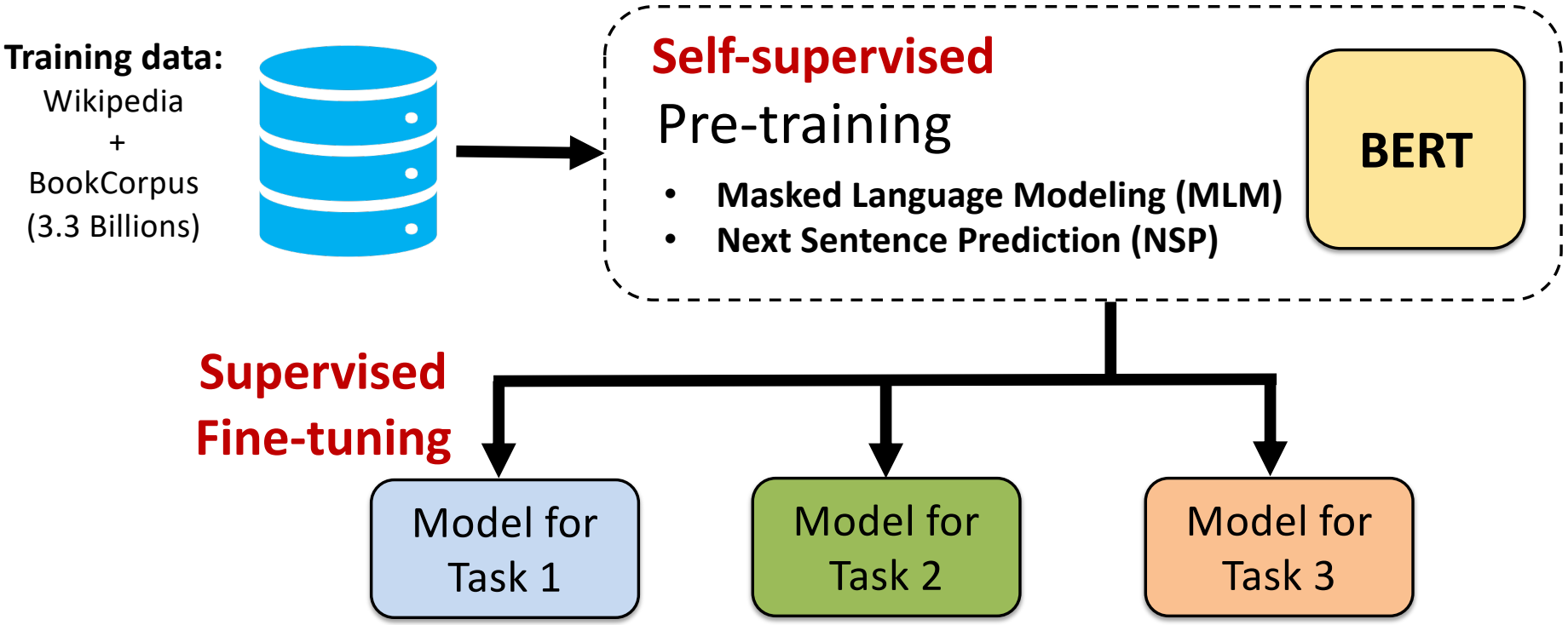
- The total pretraining loss is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{NSP}}$$

- This joint optimization ensures that BERT learns both contextualized representations and relationships between sentences.

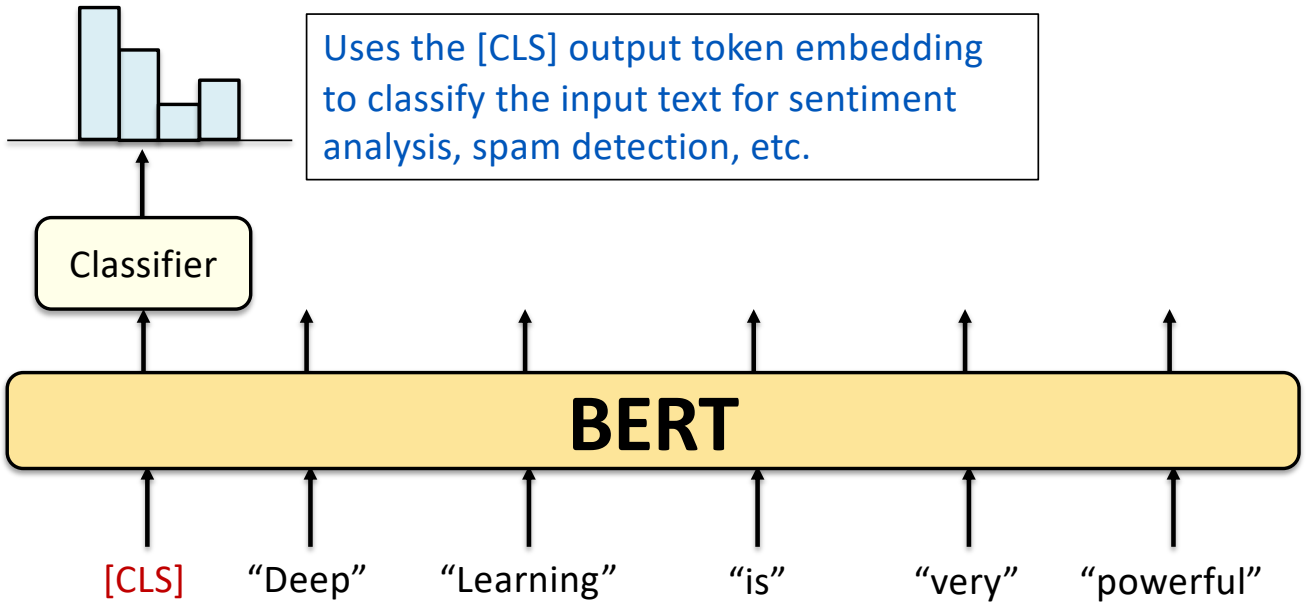
BERT needs Finetuning to Understand New Tasks

- After pre-training on a large corpus, **BERT can be fine-tuned for various NLP tasks** by adding task-specific layers and training on labeled data. This leverages BERT's rich contextualized representations, enabling strong performance with minimal task-specific data.



BERT: Text Classification

- For tasks like sentiment analysis or spam detection, a classification layer is added on top of the [CLS] token's output. The [CLS] token serves as a fixed-size representation of the entire input sequence.



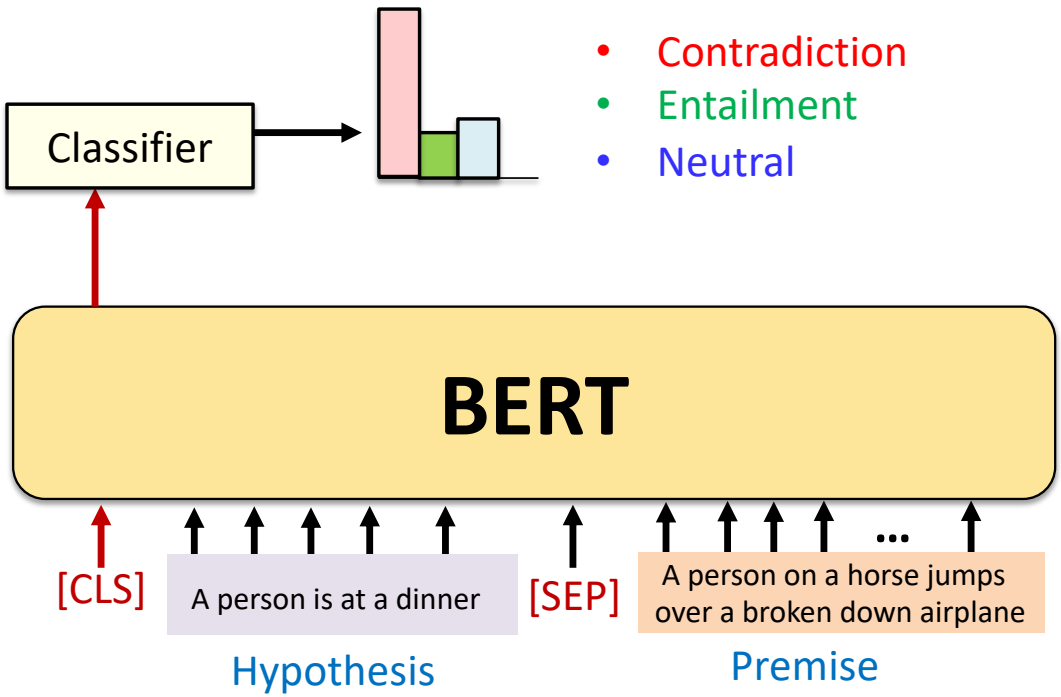
BERT – Sentiment Analysis (Example)

```
from transformers import pipeline


sentiment_pipeline = pipeline("sentiment-analysis",
                              model='distilbert-base-uncased-finetuned-sst-2-english'
data = ["It was not bad",
        "I expected to love it but I was wrong"]
sentiment_pipeline(data)

#> [{'label': 'POSITIVE', 'score': 0.9995607733726501},
#>  {'label': 'NEGATIVE', 'score': 0.997614860534668}]
```

https://towardsdatascience.com/natural-language-processing-for-absolute-beginners-a195549a3164

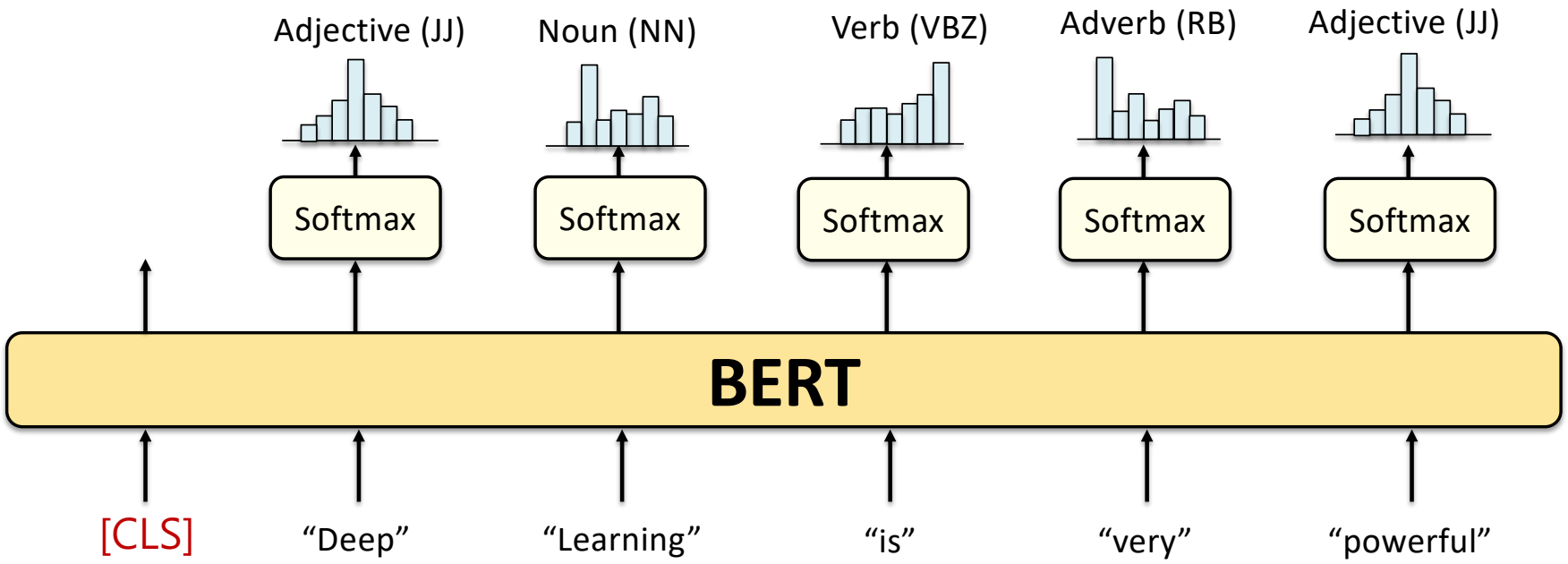
BERT – Natural Language Inference (NLI)

- **Natural Language Inference (NLI)** is a task in natural language processing that involves determining the logical relationship between two given statements.
- **Input**: 2 sentences
  - **Hypothesis:** A person is at a diner
  - **Premise:** A person on a horse jumps over a broken down airplane
- **Output**: 3 options
  - Contradiction
  - Entailment
  - Neutral



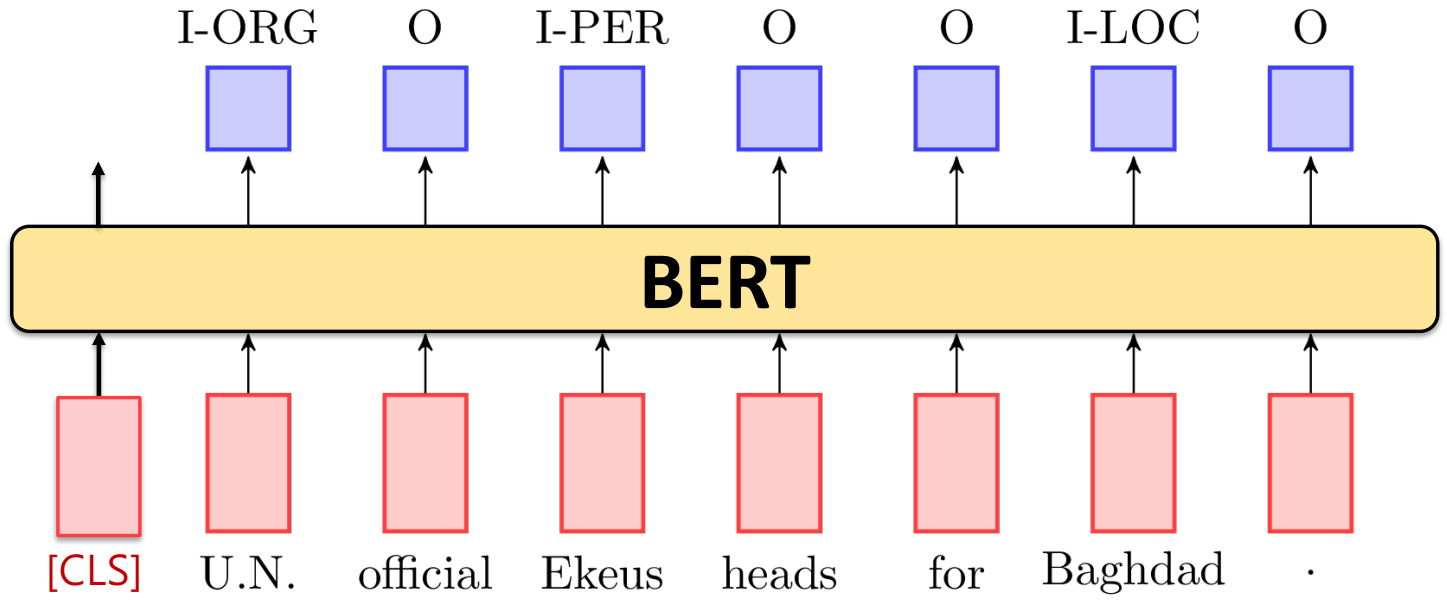
Part-of-Speech (POS) Tagging

- For sequence labeling tasks, BERT outputs a label for each token in the input sequence, typically using a softmax layer over predefined entity categories.



Named Entity Recognition (NER)

- **NER** is also known as **entity identification or entity extraction**. It is a process of identifying predefined entities present in a text such as person name, organisation, location, etc. (Aligned Sequential Paris)



BERT – Named Entity Recognition (NER)

```
from transformers import pipeline


body = "Hi, my name is Dmitrii. I am in London, I work in Super Company, " \
       "I have a question about the hotel reservation."

ner = pipeline("ner",
               model="dslim/bert-base-NER",
               aggregation_strategy='average')
ner(body)

#> [{'entity_group': 'PER',
#>   'score': 0.7563127,
#>   'word': 'Dmitrii',
#>   'start': 15,
#>   'end': 22},
#> {'entity_group': 'LOC',
#>   'score': 0.99956125,
#>   'word': 'London',
#>   'start': 32,
#>   'end': 38},
#> {'entity_group': 'ORG',
#>   'score': 0.99759734,
#>   'word': 'Super Company',
#>   'start': 50,
#>   'end': 63}]
```

Dmitrii => Person

London => Location

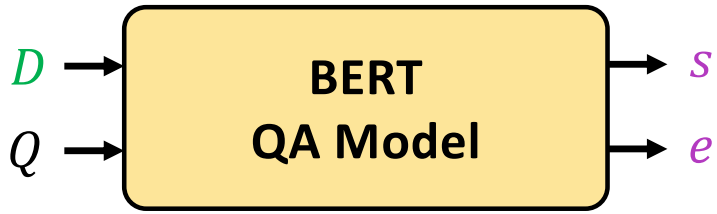
BERT: Extraction Question Answering (QA)

- **Extraction Question answering** is the task of answering questions given a context.
- For example:
  - **Context**: "Hong Kong, a captivating fusion of Eastern and Western cultures, is a thriving international financial hub and gateway to Asia's economic prowess.
  - **Question**: "What is the international financial hub of China?"
- **Two problems**:
  1. We need to find a way for BERT to understand which part of the input is the context, which one is the question.
  2. We also need to find a way for BERT to tell us where the answer starts and where it ends in the context provided.

BERT – Extraction Question Answering (QA)

Document: $D = \{d_1, d_2, \cdots, d_N\}$

Query: $Q = \{q_1, q_2, \cdots, q_M\}$



Output: two integers (s, e)

Answer: $A = \{d_s, \cdots, d_e\}$

In meteorology, precipitation is any product of the condensation of 17 spheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain 77 atte 79 cations are called "showers".

What causes precipitation to fall?
gravity $s = 17, e = 17$

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

Where do water droplets collide with ice crystals to form precipitation?
within a cloud $s = 77, e = 79$

Extraction Question Answering Example

```
question_answerer = pipeline("question-answering",
                             model='distilbert-base-cased-distilled-squad')

result = question_answerer(question="Which surface has collision with meteorites
                           context=body)
print(f"Answer: '{result['answer']}', score: {round(result['score'], 4)}, start:

#> Answer: 'the Moon', score: 0.4401, start: 93, end: 101

result = question_answerer(question="How many craters were formed",
                           context=body)
print(f"Answer: '{result['answer']}', score: {round(result['score'], 4)}, start:

#> Answer: 'at least 220', score: 0.5302, start: 600, end: 612

result = question_answerer(question="Is there atmosphere on the moon",
                           context=body)
print(f"Answer: '{result['answer']}', score: {round(result['score'], 4)}, start:

#> Answer: 'There is no', score: 0.2468, start: 220, end: 231
```

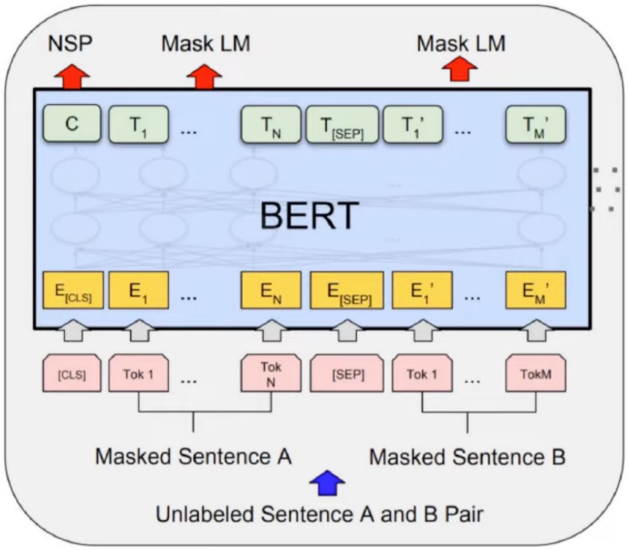
```
body = '''Obviously, the lunar surface is covered with craters, left
from previous collisions of meteorites with the Moon. Where does math go?
While a meteorite collision is a random event, its frequency
obeys probability theory laws. There is no atmosphere on the Moon's
surface, no erosion, and no wind. Therefore the lunar surface is an
ideal "book" in which the events of the last tens of thousands of
years are recorded. By studying the Moon, we can calculate how often
such objects fall on its surface.

A study of the lunar surface with high-resolution cameras is ongoing.
It has been estimated that at least 220 new craters have formed on the
Moon over the past 7 years. This check is also vital because
these calculations can help assess the danger to the Earth.'''
```

https://towardsdatascience.com/natural-language-processing-for-absolute-beginners-a195549a3164

BERT Fine-Tuning for Understanding New Tasks

Pretraining

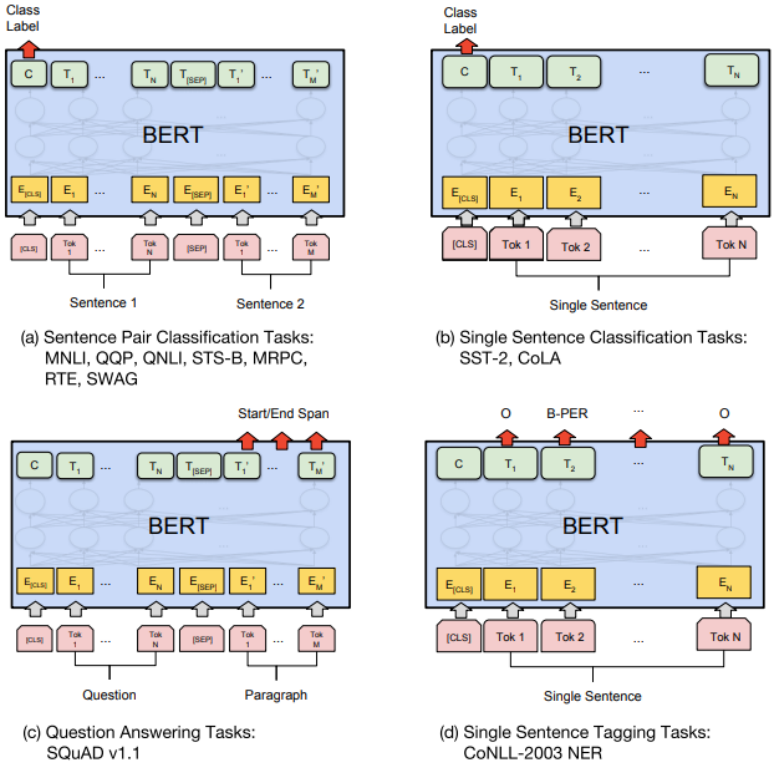
Self-Supervised with **unlabeled** data



Language Understanding

Finetuning Training

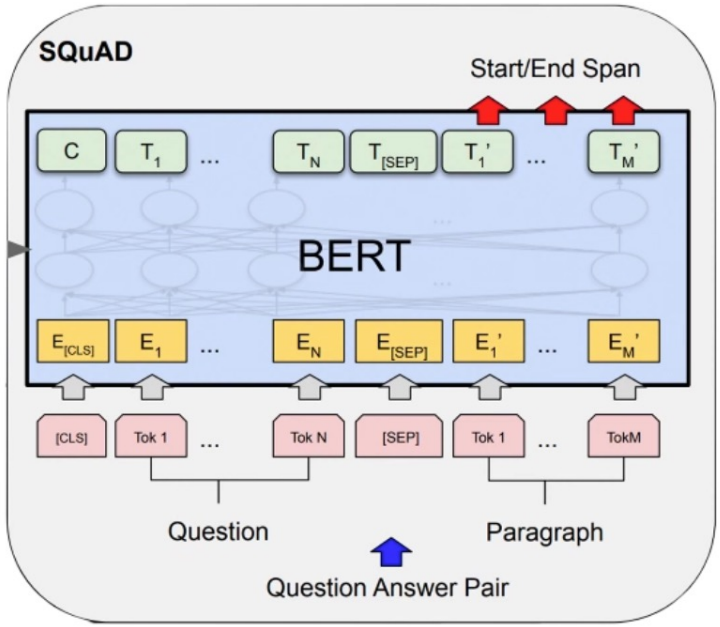
Supervised with labeled data



Adapting to different tasks

Fine-Tuned BERT's Performance

- **SQuAD** : **S**tanford **Qu**estion & **A**nswer **D**ataset
- Only needs 30 minute to fine tune on single cloud TPU with 91% F1 score



GLUE: General Language Understanding Evaluation

1. Corpus of Linguistic Acceptability (CoLA)
2. Stanford Sentiment Treebank (SST-2)
3. Microsoft Research Paraphrase Corpus (MRPC)
4. Quora Question Pairs (QQP)
5. Semantic Textual Similarity Benchmark (STS-B)
6. Multi-Genre Natural Language Inference (MNLI)
7. Question-answering NLI (QNLI)
8. Recognizing Textual Entailment (RTE)
9. Winograd NLI (WNLI)

https://gluebenchmark.com/

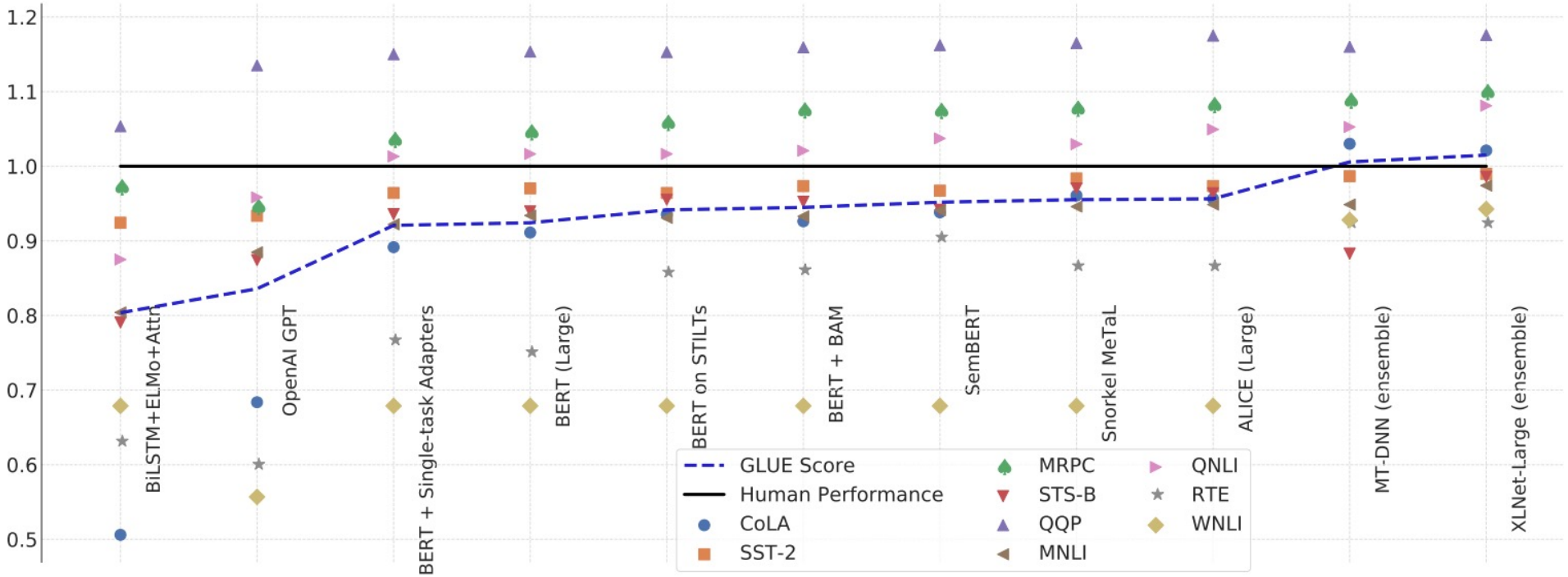
GLUE also has Chinese version (https://www.cluebenchmarks.com/)

GLUE Benchmark

GLUE (General Language Understanding Evaluation) is a benchmark designed to evaluate and analyze natural language understanding systems. It consists of a diverse set of natural language processing tasks, including:

- **CoLA** - The Corpus of Linguistic Acceptability consists of English acceptability judgments.
- **SST-2** - The Stanford Sentiment Treebank is a binary single-sentence classification task consisting of sentences extracted from movie reviews with human annotations of their sentiment.
- **MRPC** - Microsoft Research Paraphrase Corpus consists of sentence pairs automatically extracted from online news sources, with human annotations for whether the sentences in the pair are semantically equivalent.
- **STS-B** - The Semantic Textual Similarity Benchmark contains sentence pairs drawn from news headlines, video and image captions, and natural language inferences. Human raters assigned a similarity score from 1 to 5 for each pair.
- **QQP** - Quora Question Pairs is a binary classification task to determine if two questions posted on Quora are semantically equivalent.
- **MNLI** - Multi-Genre Natural Language Inference has sentence pairs with textual entailment annotations.
- **QNLI** - Question Natural Language Inference contains question-paragraph pairs, labeled for whether the context paragraph contains the answer to the question.
- **RTE** - Recognizing Textual Entailment contains sentence pairs labeled for entailment.

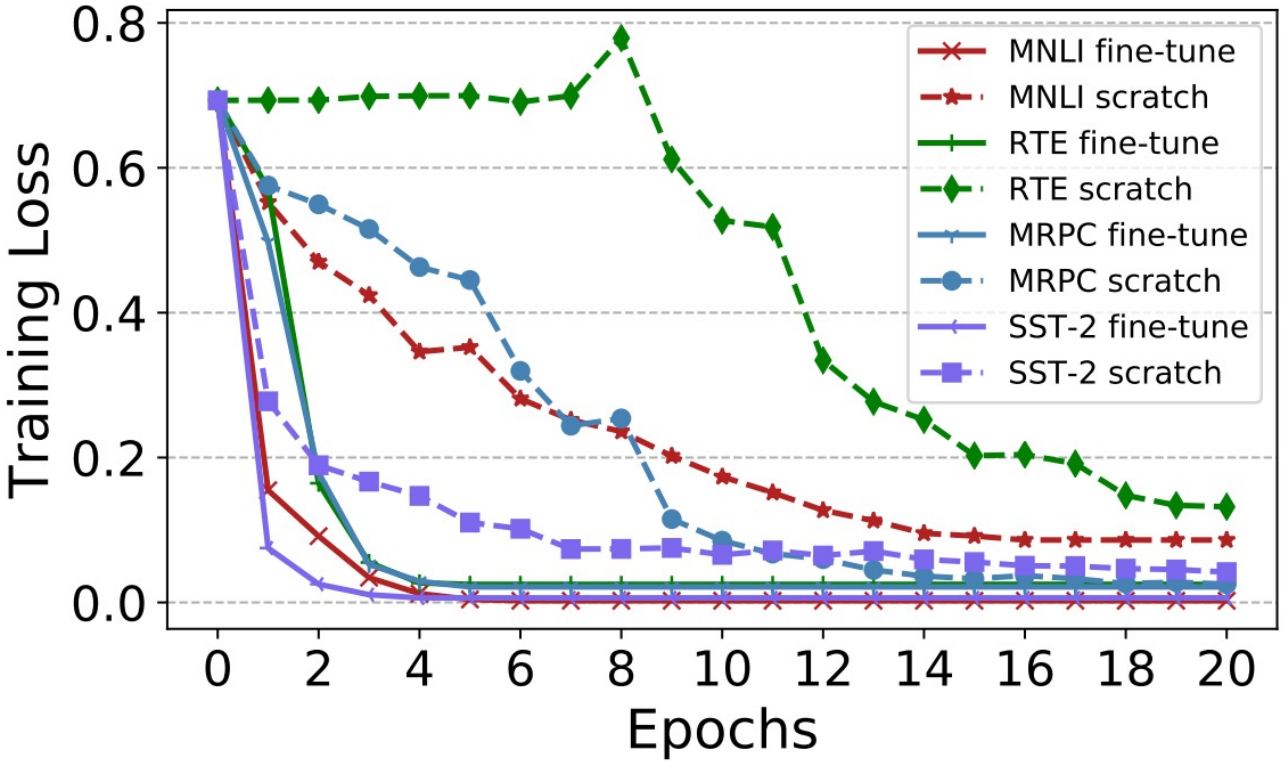
GLUE Scores of BERT and its Family



https://arxiv.org/abs/1905.00537

Pre-Train v.s. Random Initialization

(fine-tune) (scratch)



https://arxiv.org/abs/1908.05620

BERT Results on NER

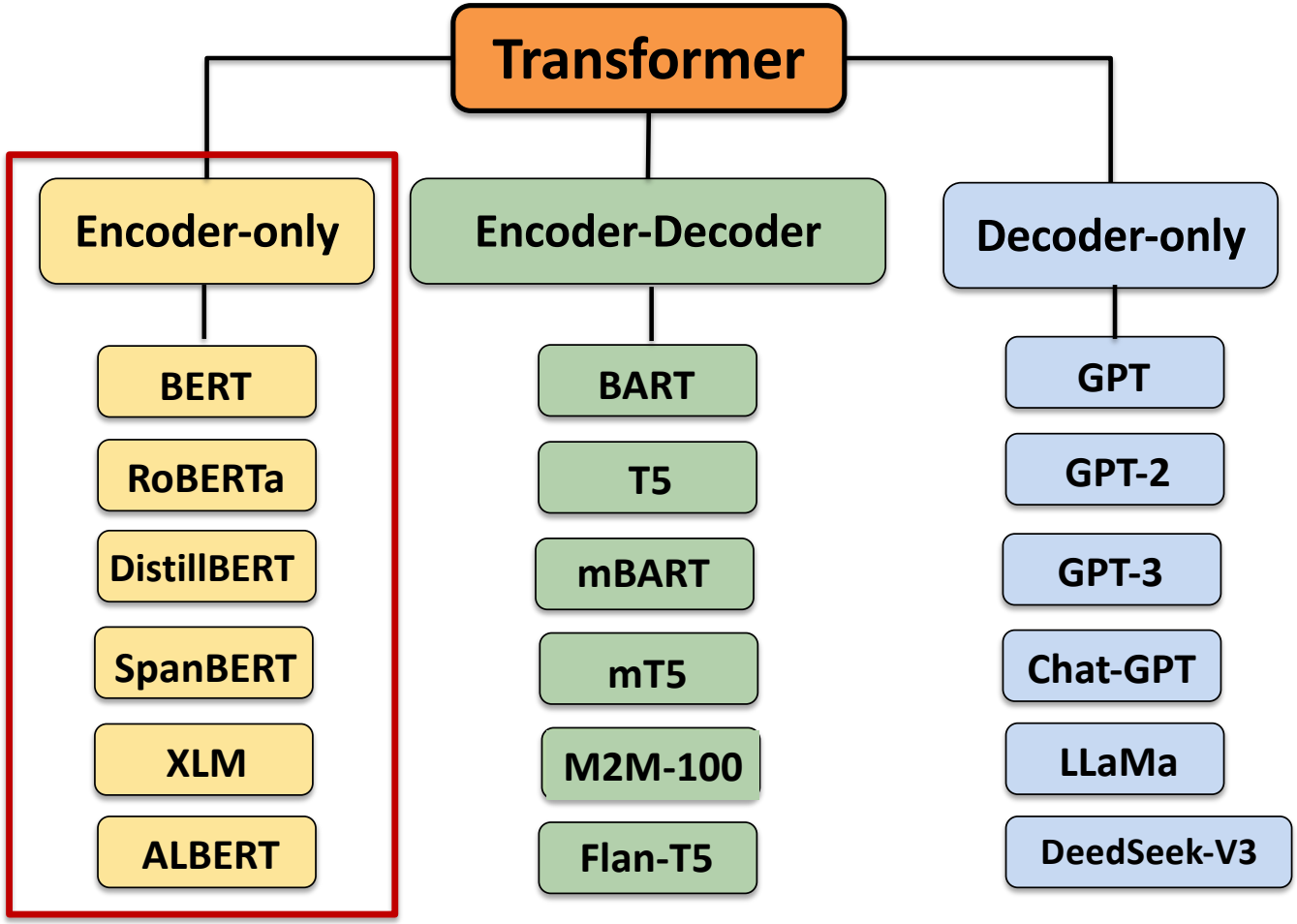
- **Name Entity Recognition (NER)**

Model	Description	CONLL 2003 F1
TagLM (Peters+, 2017)	LSTM BiLM in BLSTM Tagger	91.93
ELMo (Peters+, 2018)	ELMo in BLSTM	92.22
BERT-Base (Devlin+, 2019)	Transformer LM + fine-tune	92.4
CVT Clark	Cross-view training + multitask learn	92.61
BERT-Large (Devlin+, 2019)	Transformer LM + fine-tune	92.8
Flair	Character-level language model	93.09

BERT Variants

(Optional)

Well-known Transformer-based Models



BERT Variants

BERT (2018-10) outperformed various existing SOTA models on different evaluation metrics. Variants of BERT models were proposed starting at 2019.

- **RoBERTa (2019)**
- **DistilBERT (2019)**
- **SpanBERT (2019)**
- **XLM (2019)**
- **ALBERT (2019)**

RoBERTa (2019-07)

Robustly **o**ptimized **BERT** Pre-Training **a**pproach (RoBERTa) from Facebooks, which robustly optimized BERT's training and architecture.

Key Optimizations

- Dynamic Masking

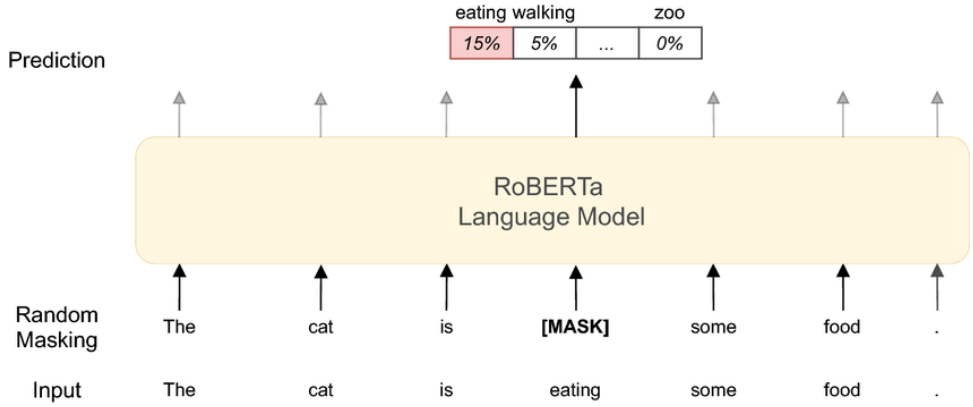
Masking patterns regenerate every epoch.

- NSP Removed

Next Sentence Prediction was found to hurt performance.

- Scale

Trained on 160GB of data (vs BERT's 16GB).



https://arxiv.org/pdf/1907.11692.pdf

RoBERTa vs BERT Performance

Comparison	BERT October 11, 2018	RoBERTa July 26, 2019
Parameters	**Base:** 110M **Large:** 340M	**Base:** 125 **Large:** 355
Layers / Hidden Dimensions / Self-Attention Heads	**Base:** 12 / 768 / 12 **Large:** 24 / 1024 / 16	**Base:** 12 / 768 / 12 **Large:** 24 / 1024 / 16
Training Time	**Base:** 8 x V100 x 12d **Large:** 280 x V100 x 1d	1024 x V100 x 1 day (4-5x more than BERT)
Performance	Outperforming SOTA in Oct 2018	88.5 on GLUE
Pre-Training Data	BooksCorpus + English Wikipedia = 16 GB	BERT + CCNews + OpenWebText + Stories = 160 GB
Method	Bidirectional Trans-former, MLM & NSP	BERT without NSP, Using Dynamic Masking

RoBERTa's GLUE Results

	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
Single-task single models on dev										
$\text{BERT}_{\text{LARGE}}$	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-	-
$\text{XLNet}_{\text{LARGE}}$	89.8/-	93.9	91.8	83.8	95.6	89.2	63.6	91.8	-	-
RoBERTa	**90.2/90.2**	**94.7**	**92.2**	**86.6**	**96.4**	**90.9**	**68.0**	**92.4**	**91.3**	-
Ensembles on test (from leaderboard as of July 25, 2019)										
ALICE	88.2/87.9	95.7	**90.7**	83.5	95.2	92.6	**68.6**	91.1	80.8	86.3
MT-DNN	87.9/87.4	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0	87.6
XLNet	90.2/89.8	98.6	90.3	86.3	**96.8**	**93.0**	67.8	91.6	**90.4**	88.4
RoBERTa	**90.8/90.2**	**98.9**	90.2	**88.2**	96.7	92.3	67.8	**92.2**	89.0	**88.5**

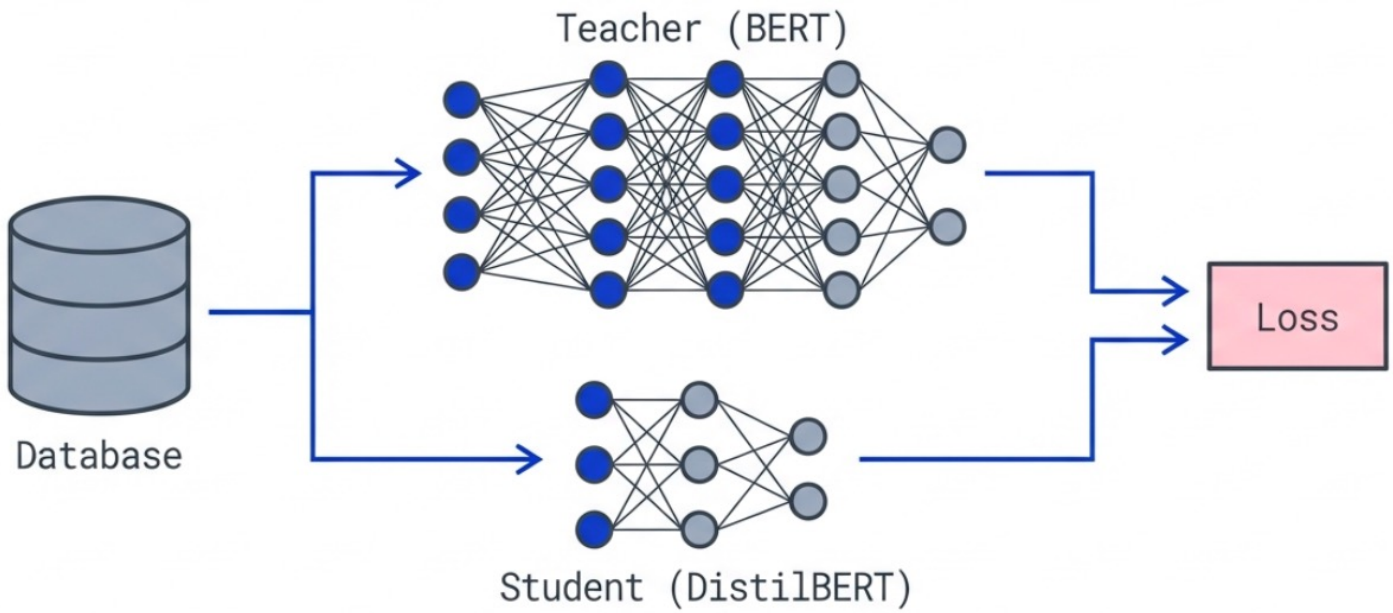
GLUE: General Language Understanding Evaluation

DistilBERT (2019)

- **DistilBERT** – **Distil**led and Lightweight version of **BERT**
  - **Deigned for a lightweight model** to deploy in production environment
  - Trained using **Knowledge Distillation Technique**
  - 60% faster than BERT
  - 40% less memory than BERT
  - Retains 97% of BERT's performance

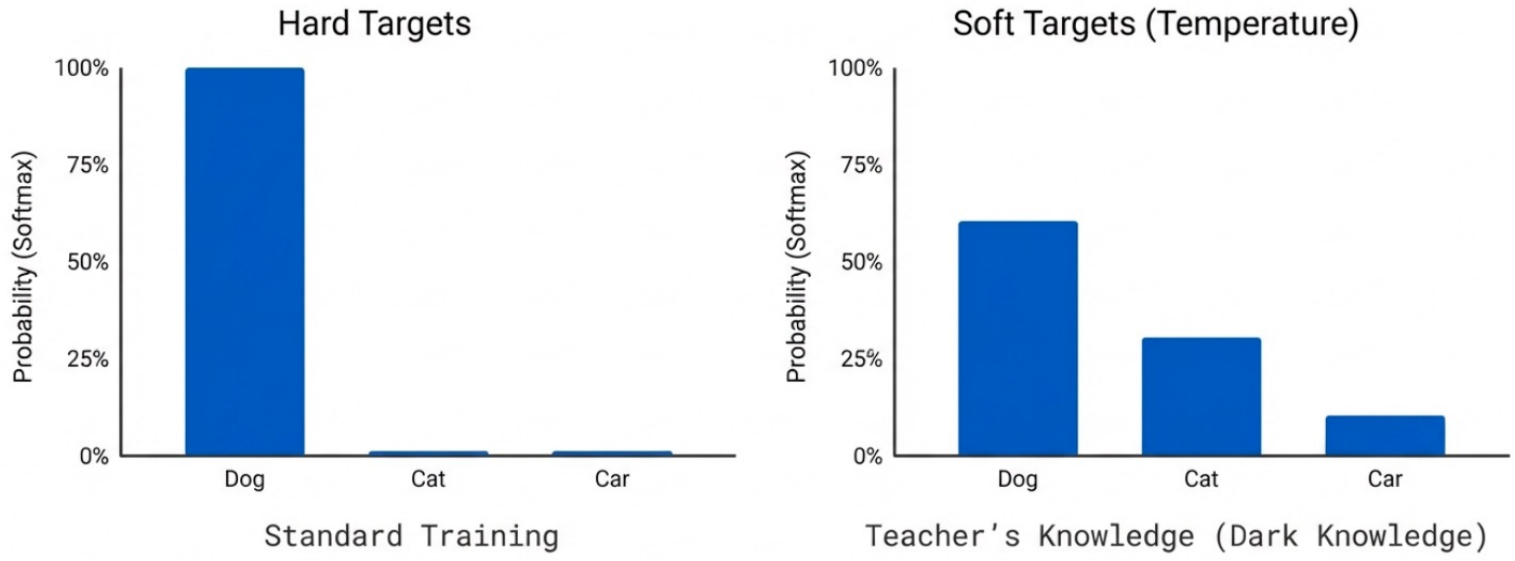
Knowledge Distillation

The Student is trained to mimic the probability distributions (soft targets) of the Teacher, not just the raw data.



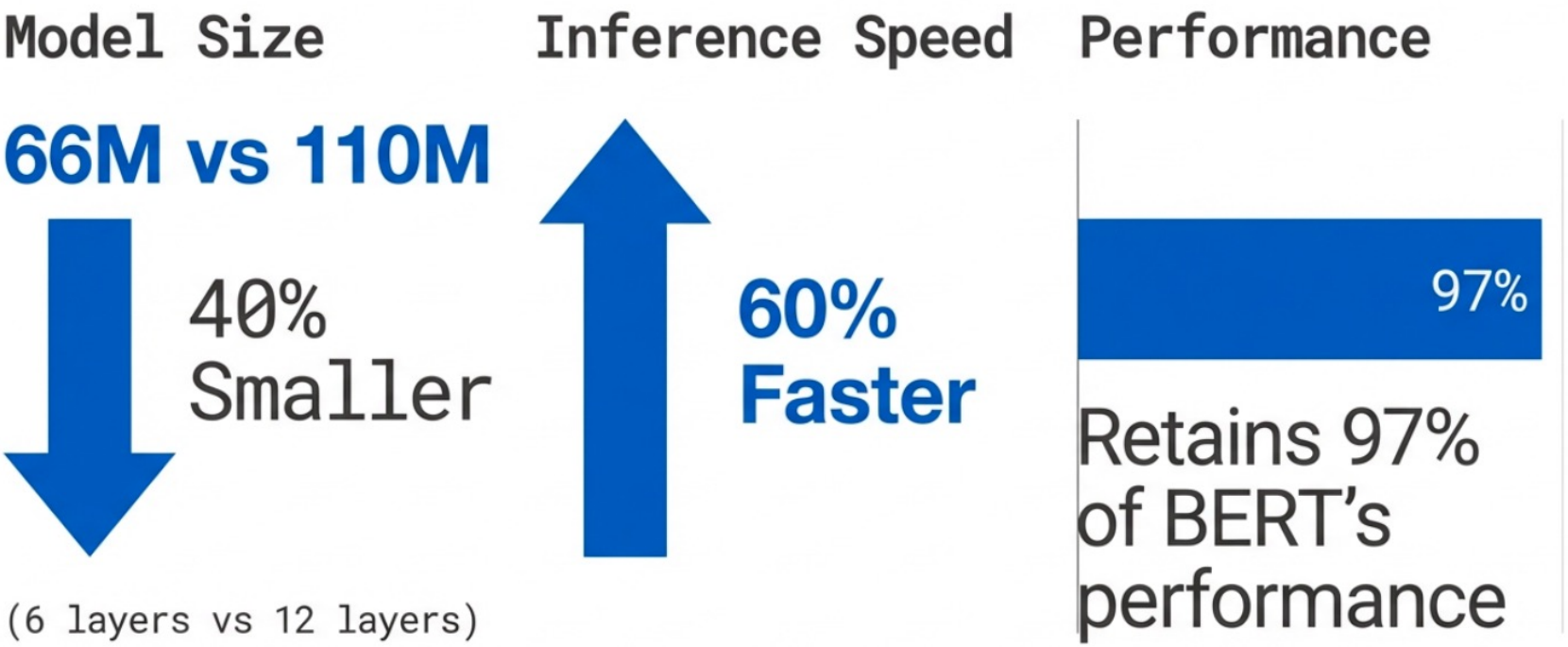
Distillation Objectives: Soft vs. Hard Targets

The objective of the student model will be to minimize the final distillation loss which is a weighted sum of **Loss (hard)** and **Loss (soft)**



$$\text{Final Loss} = \alpha \cdot \text{Loss(soft)} + (1 - \alpha) \cdot \text{Loss(hard)}$$

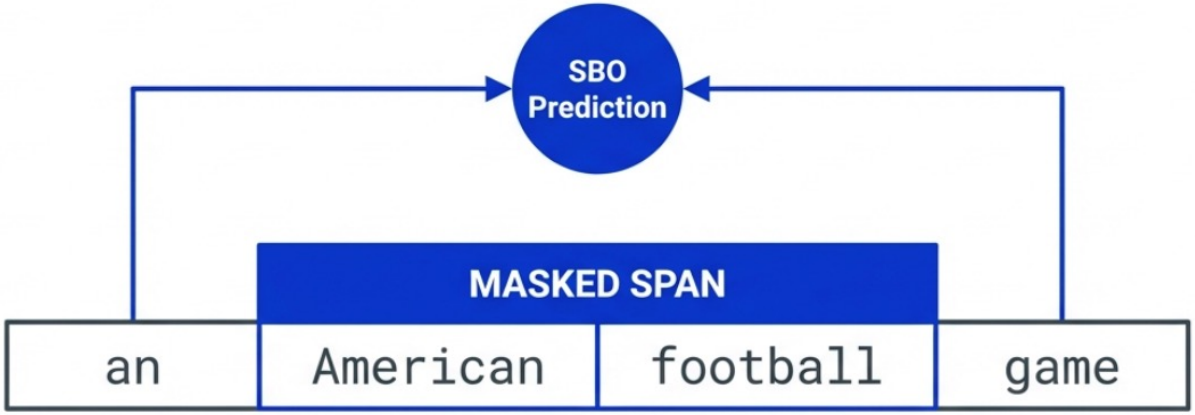
DistilBERT Results

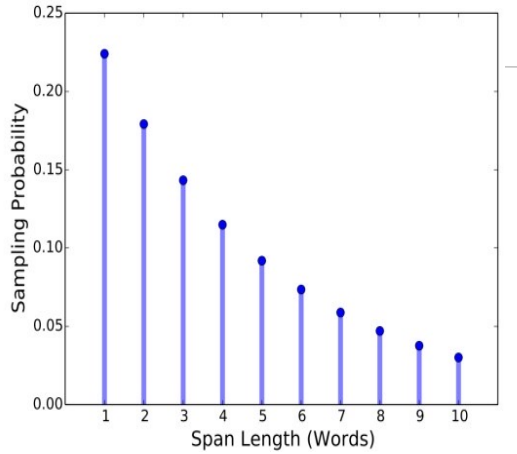


SpanBERT (2019): Better Masking

Instead of masking random tokens, SpanBERT **masks contiguous spans**.

The model must predict the span using only the boundary tokens.





https://arxiv.org/pdf/1907.10529.pdf

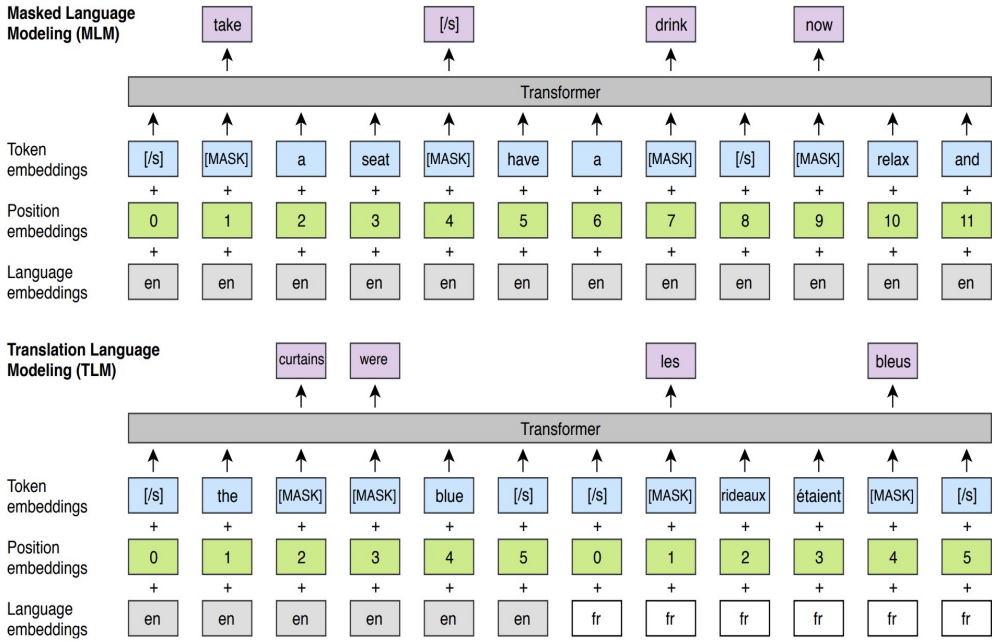
SpanBERT Results

Method	SQuAD 2.0 Score
Random Token Masking (BERT)	83.8
Random Span Masking + SBO (SpanBERT)	**86.8**

SpanBERT achieves State-of-the-Art on Extractive Question Answering by learning phrase-level structure.

XLM – Cross-Lingual Language Model (2019-01)

- Designed for **Cross-lingual Language Tasks** like text classification and machine translation.
- **Learns to map words from different languages** by using BPE and a dual-language training mechanism
- Training objective with both:
  - Masked Language Modeling (MLM)
  - Translation Language Modeling (TLM)



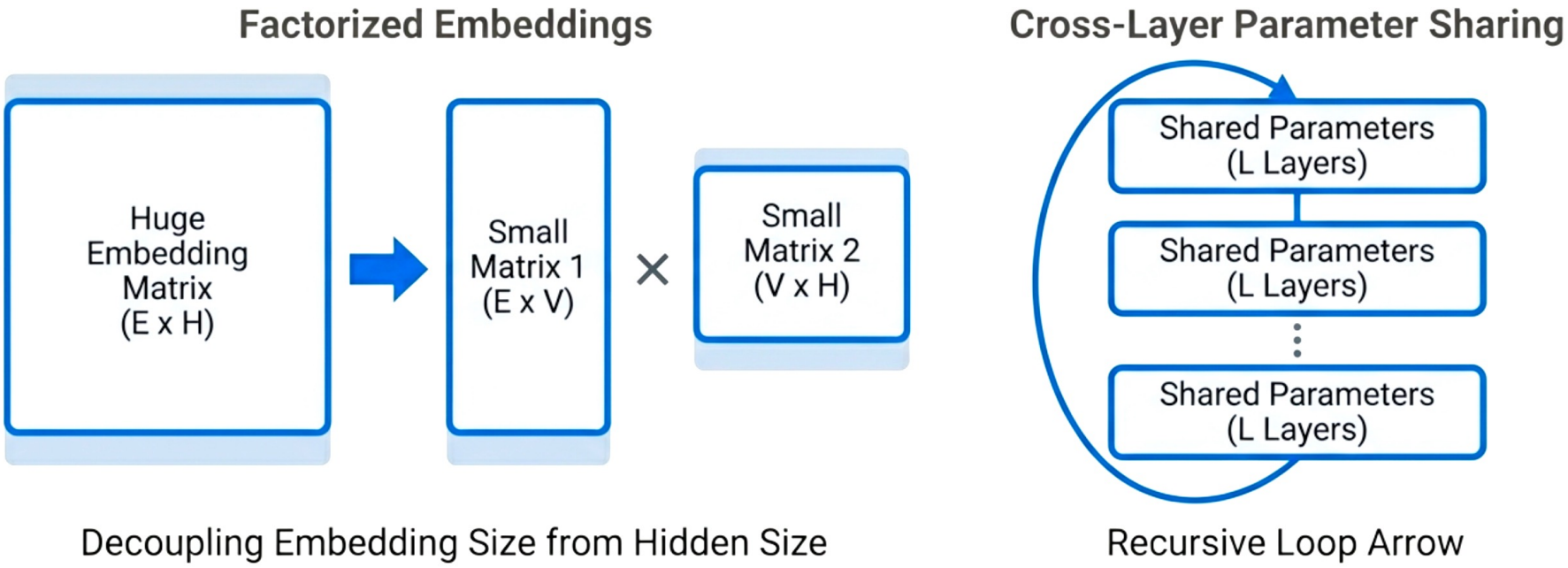
https://arxiv.org/pdf/1901.07291.pdf

ALBERT : A Lite BERT (2019-09)

- A Lite BERT for Self-supervised Learning of Language Representation
  1. Token embedding is decoupled from the hidden dimension
  2. All layers share parameters
  3. NSP objective is replaced with sentence ordering prediction
- Modifications made ALBERT to train for larger models with few parameters.

https://arxiv.org/pdf/1909.11942.pdf

ALBERT: A Lite BERT



Drastic parameter reduction: 12M (ALBERT) vs 110M (BERT-Base).

ALBERT's Objective Shift: SOP

NSP (Topic Matching)

Easy for BERT to distinguish

SOP (Sentence Order Prediction)

Positive	She cooked dinner.	She ate the food.
Negative	She ate the food.	She cooked dinner.

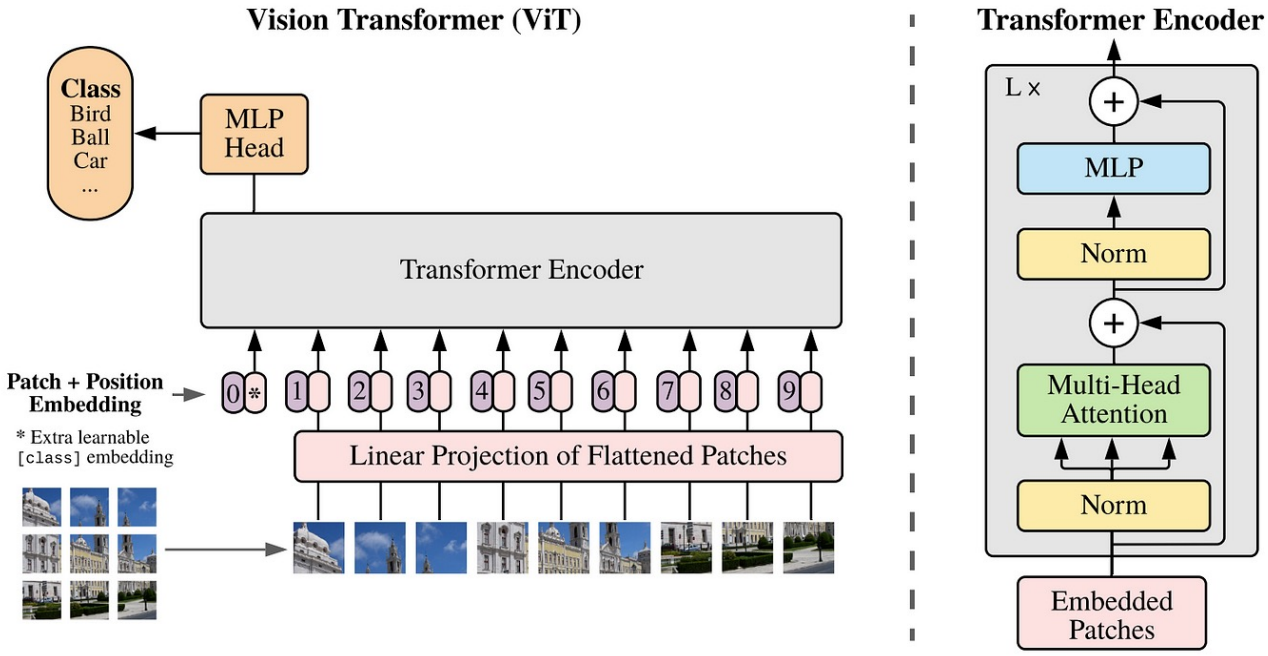
SOP forces the model to learn deep coherence and logical flow, not just keyword matching.

ALBERT's GLUE Results

Models	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
Single-task single models on dev										
BERT-large	86.6	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-	-
XLNet-large	89.8	93.9	91.8	83.8	95.6	89.2	63.6	91.8	-	-
RoBERTa-large	90.2	94.7	**92.2**	86.6	96.4	**90.9**	68.0	92.4	-	-
ALBERT (1M)	90.4	95.2	92.0	88.1	96.8	90.2	68.7	92.7	-	-
ALBERT (1.5M)	**90.8**	**95.3**	**92.2**	**89.2**	**96.9**	**90.9**	**71.4**	**93.0**	-	-
Ensembles on test (from leaderboard as of Sept. 16, 2019)										
ALICE	88.2	95.7	**90.7**	83.5	95.2	92.6	**69.2**	91.1	80.8	87.0
MT-DNN	87.9	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0	87.6
XLNet	90.2	98.6	90.3	86.3	96.8	93.0	67.8	91.6	90.4	88.4
RoBERTa	90.8	98.9	90.2	88.2	96.7	92.3	67.8	92.2	89.0	88.5
Adv-RoBERTa	91.1	98.8	90.3	88.7	96.8	93.1	68.0	92.4	89.0	88.8
ALBERT	**91.3**	**99.2**	90.5	**89.2**	**97.1**	**93.4**	69.1	**92.5**	**91.8**	**89.4**

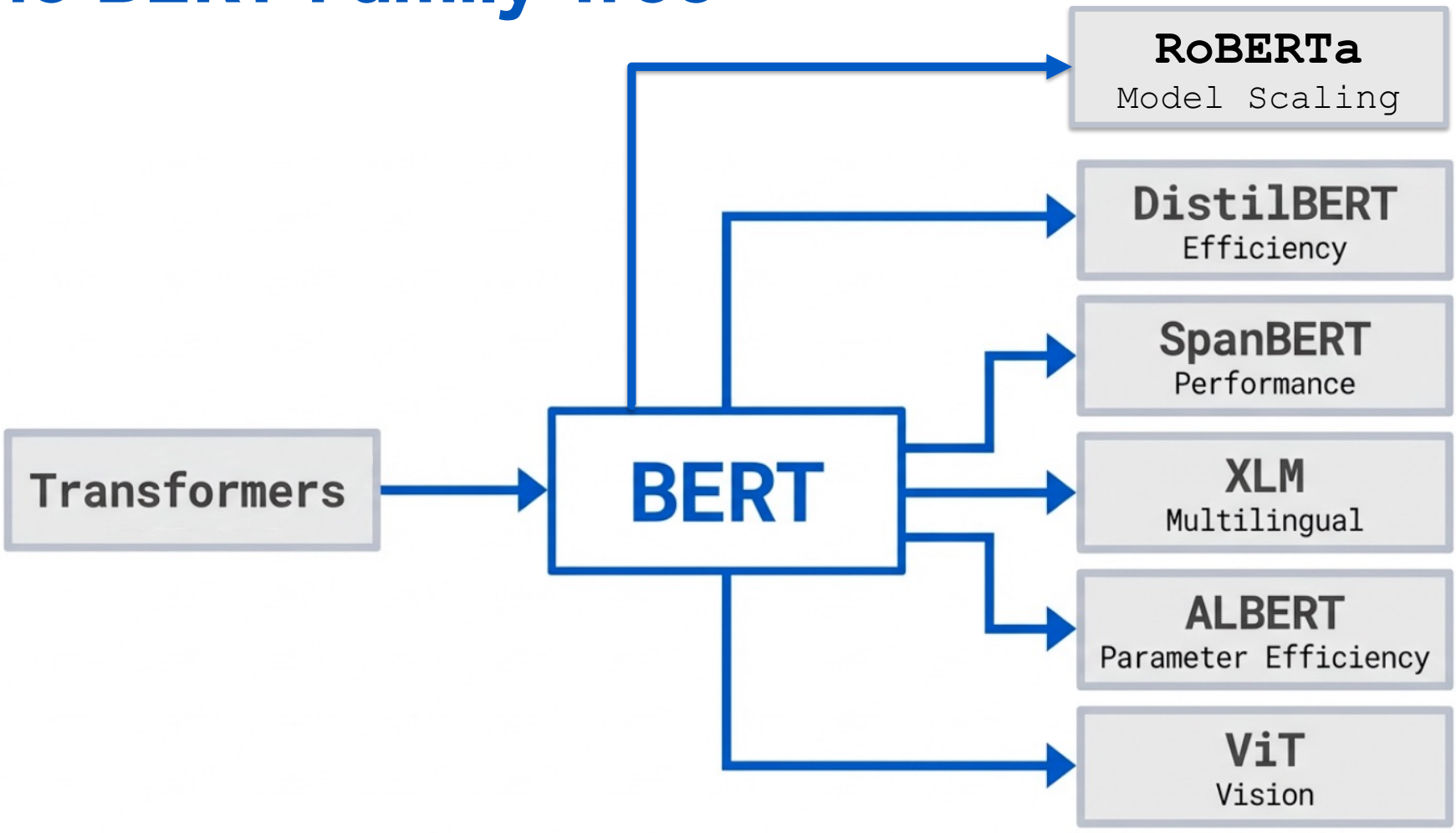
Vision Transformer (ViT) (2020-10)

BERT's architecture is domain-agnostic. It processes sequences of information, whether they are text tokens or image patches.



Dosovitskiy, et al. (2020), "An image is worth 16x16 words: Transformers for image recognition at scale."

The BERT Family Tree



Why BERT Changed Everything?

- **Context Solved**
  - Shifted NLP from static word vectors to dynamic contextual embeddings.
- **Bidirectionality**
  - Enabled models to see the full context of a sentence simultaneously.
- **The Encoder Standard**
  - Established the architectural blueprint for all modern classification and understanding tasks.

What BERT Can (And Cannot) Do

Strengths (Encoder-Only)

- **Sentiment Analysis** (Review Classification)
- **Named Entity Recognition** (Extraction)
- **Extractive Question Answering** (Highlighting answers)
- **Sentence Classification** (Spam detection)

Limitations

- Generative Storytelling (Creative writing)
- Open-ended Chat (Conversational AI)
- Text Completion (Predicting the future)

BERT is a 'Reader' , not a 'Writer'.