BERT

Bidirectional Encoder Representations from Transformers

(A Technique for Natural Language Processing)

Gene Olafsen

BEGINNING

- BERT is an open-sourced NLP pre-training model developed by researchers at Google in 2018.
- The best part about BERT is that it can be download and used for free.
- BERT is a direct descendant to GPT (Generalized Language Models)
- BERT has provided top results:
  - Question Answering (SQuAD v1.1)
  - Natural Language Inference (MNLI)

PRE-TRAINING

- It’s built on pre-training contextual representations, which include:
  - Semi-supervised Sequence Learning (by Andrew Dai and Quoc Le)
  - ELMo (by Matthew Peters and researchers from AI2 and UW CSE)
  - ULMFiT (by fast.ai founder Jeremy Howard and Sebastian Ruder)
  - the OpenAI transformer (by OpenAI researchers Radford, Narasimhan, Salimans, and Sutskever)
  - Vaswani et al

UNIQUE

- It is the first deeply bidirectional, unsupervised language representation, pre-trained using only a plain text corpus.
- BERT is pre-trained on a large corpus of unlabeled text, including:
  - the entire content of **Wikipedia** (~2,500 million words)
  - a book corpus (800 million words)

HOW BERT IS DIFFERENT

- Traditional models (like word2vec or GloVe) are context-free-- they generate a single word embedding representation for each word in the vocabulary.
- In a traditional model example: the word **“right”** would have the same context-free representation in “*I’m sure I’m right*” and “*Take a right turn.*”
- Or the word “*bank*” would have the same context-free representation in “*bank account*” and “*bank of the river.*”
- However, BERT would represent based on both previous and next context making it bidirectional. BERT was first on its kind to successfully pre-train bidirectionality into a deep neural network.

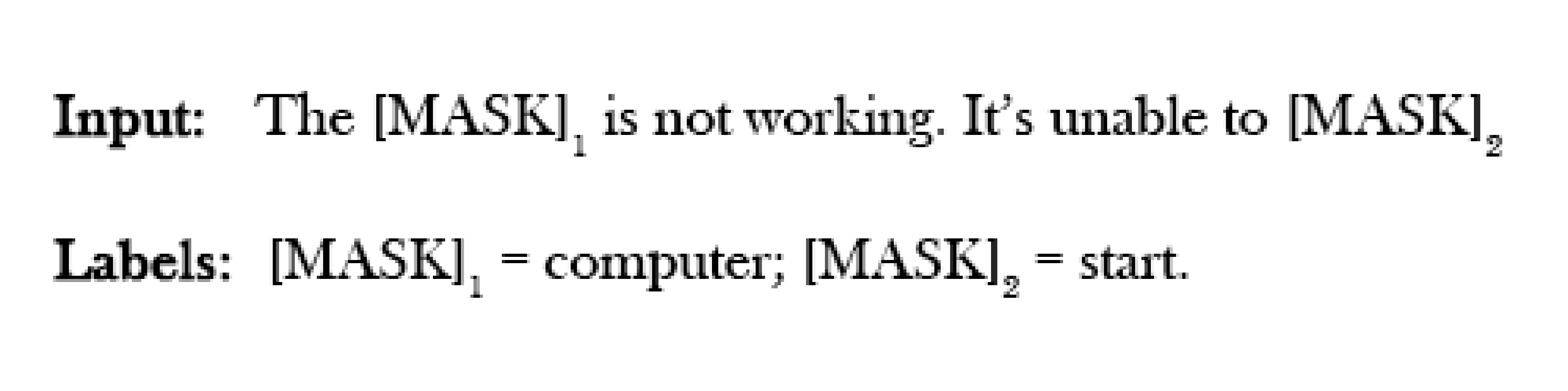
TWO STRATEGIES

- Using:
    - Mask Language Model (MLM)
    - Next Sentence Prediction (NSP)

WORD MASKING

- BERT makes use of a novel technique of masking out some of the words in the input and then tries to predict the masked words.
- 15% of the words in each sequence are replaced with a [MASK] token. The model then attempts to predict the original value of the masked words, based on the context provided by the other, non-masked, words in the sequence.
- The model looks in **both** directions and it uses the full context of the sentence, both left and right surroundings, in order to predict the masked word

MASK SAMPLE

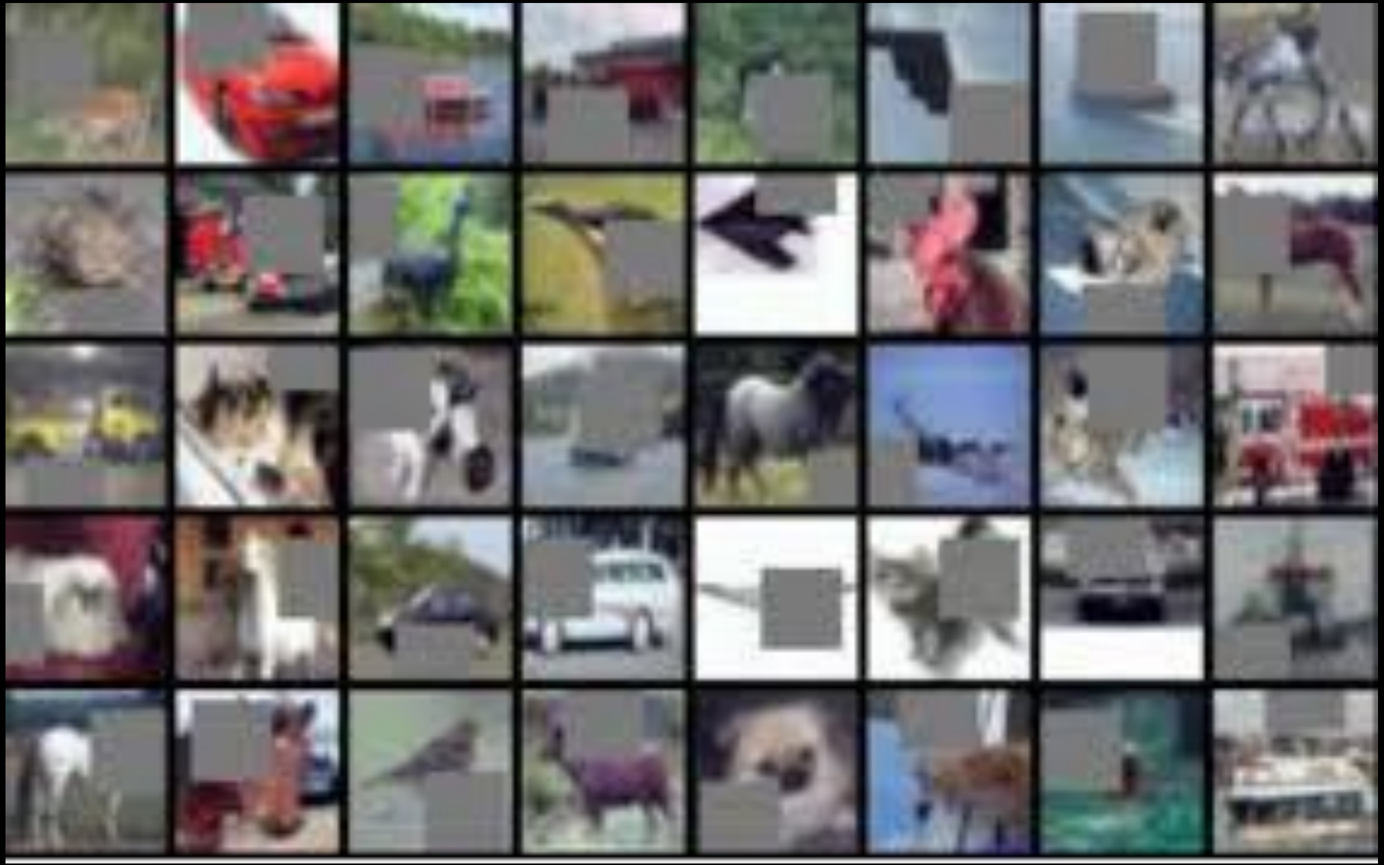


Input: The $[\text{MASK}]_1$ is not working. It’s unable to $[\text{MASK}]_2$

Labels: $[\text{MASK}]_1$ = computer; $[\text{MASK}]_2$ = start.

IMAGE CUTOUT

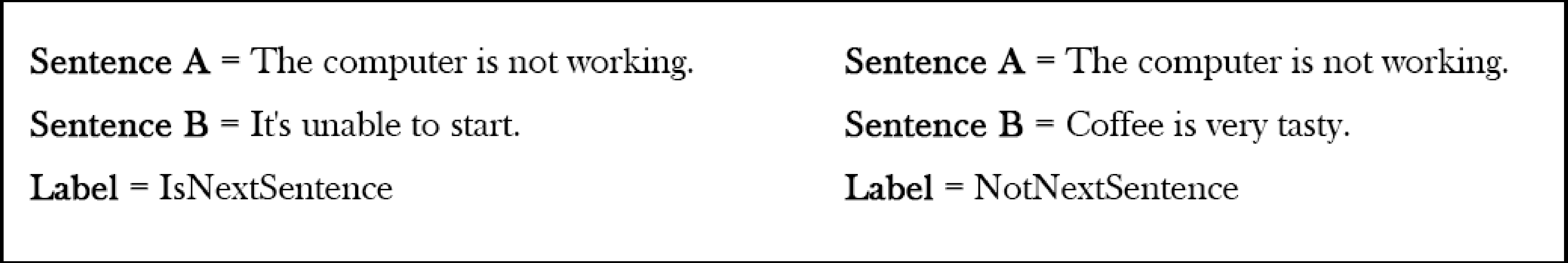
- Sounds similar to image dropout in CNN model training...



NEXT SENTENCE PREDICTION

- The second technique is **Next Sentence Prediction (NSP)**
- BERT learns to model relationships between sentences.
- During training, the model receives pairs of sentences as input and learns to predict if the second sentence in the pair is the sentence which follows in the original document.

NSP SAMPLE



Sentence A = The computer is not working.

Sentence B = It's unable to start.

Label = IsNextSentence

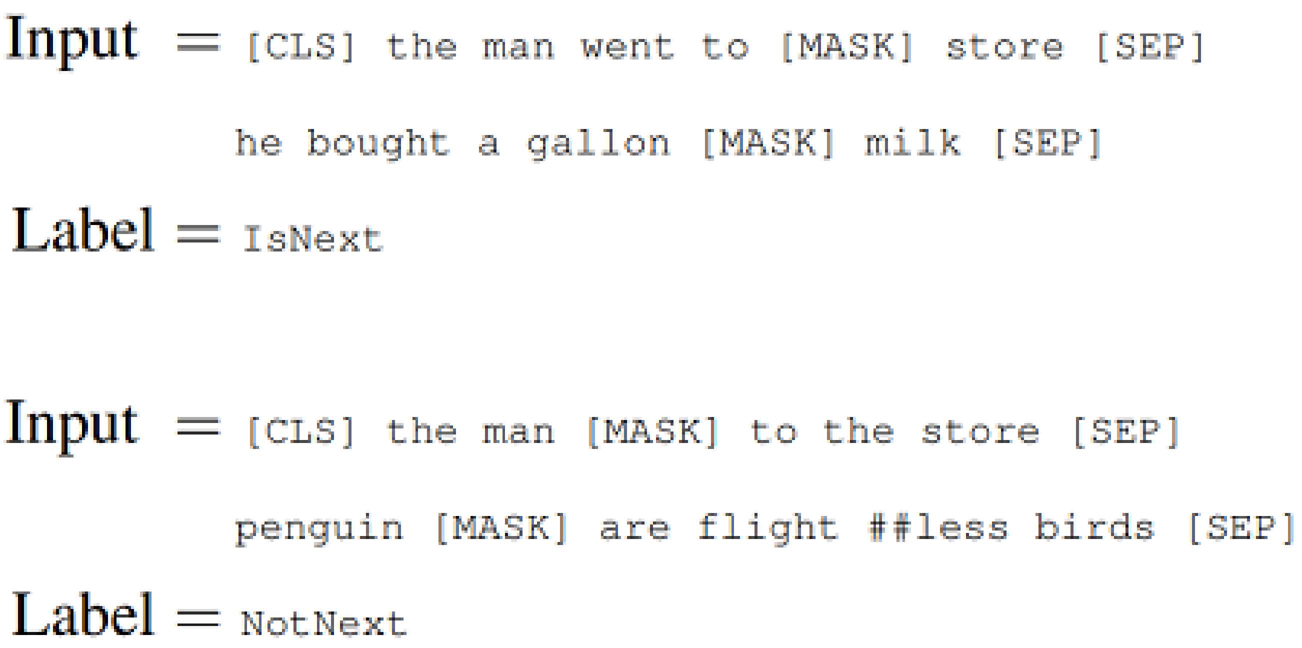
Sentence A = The computer is not working.

Sentence B = Coffee is very tasty.

Label = NotNextSentence

TOKENIZATION OF SENTENCES

- To predict if the second sentence is connected to the first one or not, basically the complete input sequence goes through the Transformer based model, the output of the [CLS] token is transformed into a 2×1 shaped vector using a simple classification layer, and the IsNext-Label is assigned using softmax.



Input = [CLS] the man went to [MASK] store [SEP]

he bought a gallon [MASK] milk [SEP]

Label = IsNext

Input = [CLS] the man [MASK] to the store [SEP]

penguin [MASK] are flight ##less birds [SEP]

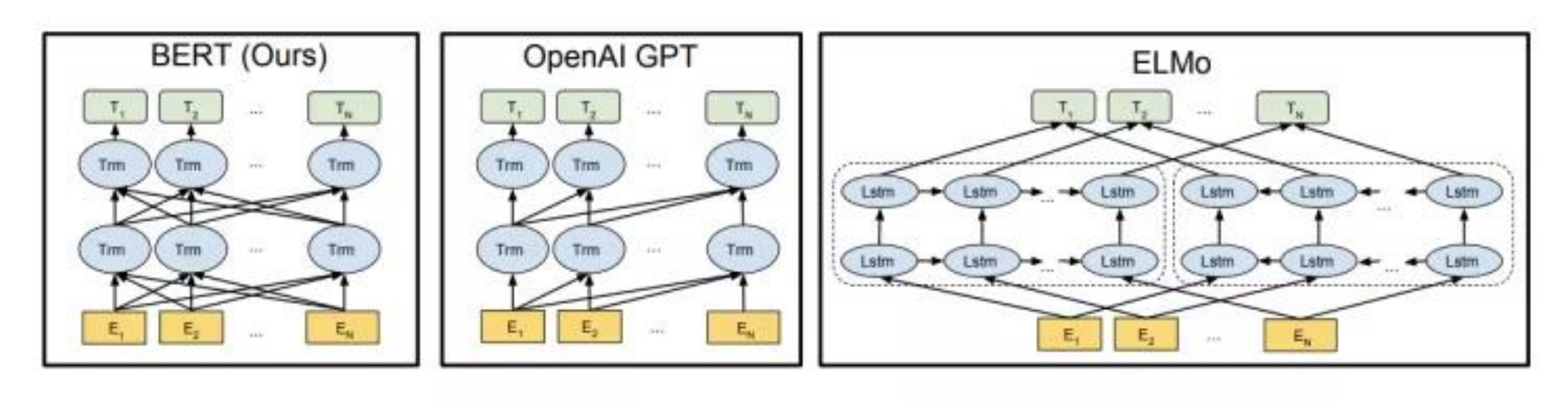
Label = NotNext

NSP USAGE

- A pre-trained model with this kind of understanding is relevant for tasks like question answering.

ARCHITECTURE

- A visualization of BERT’s neural network architecture compared to previous state-of-the-art contextual pre-training methods.The arrows indicate the information flow from one layer to the next. The green boxes at the top indicate the final contextualized representation of each input word.



NOT LSTM

- BERT is based on the Transformer model architecture, instead of LSTMs.
- Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video).
- A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

TRANSFORMER

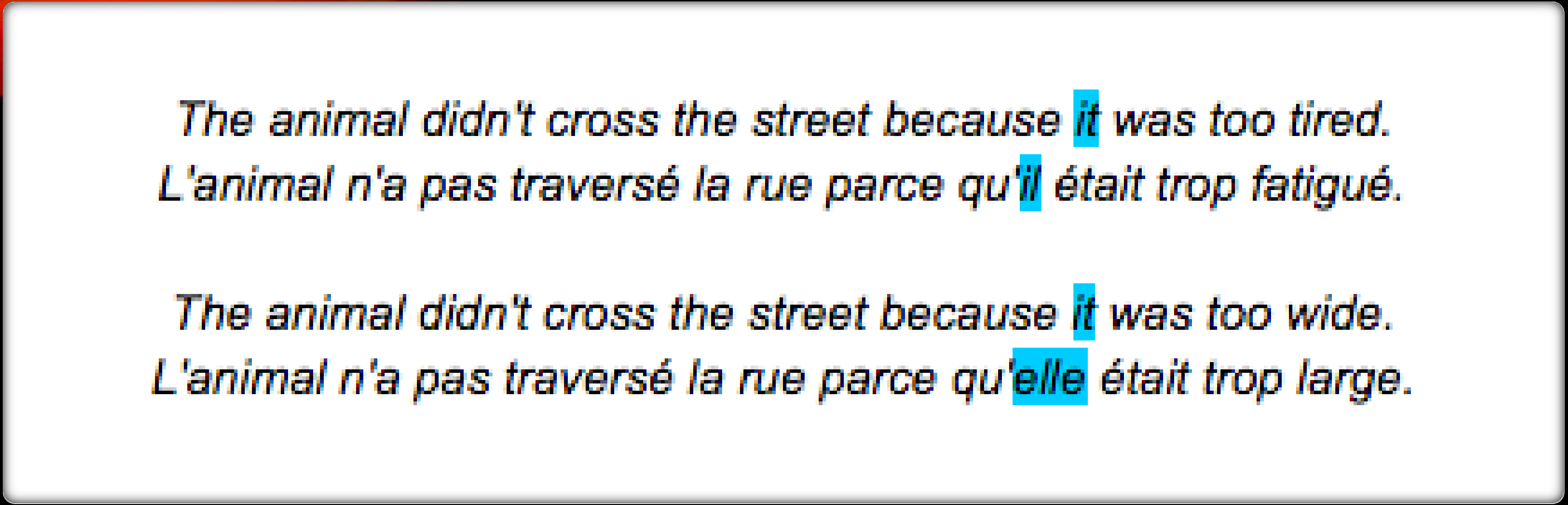
- A Transformer works by performing a small, constant number of steps.
- In each step, it applies an attention mechanism to understand relationships between all words in a sentence, regardless of their respective position.
- For example, given the sentence, "I arrived at the bank after crossing the river", to determine that the word "bank" refers to the shore of a river and not a financial institution, the Transformer can learn to immediately pay attention to the word "river" and make this decision in just one step.

UNDERSTANDING TRANFORMER

- To compute the next representation for a given word - “bank” for example - the Transformer compares it to every other word in the sentence.
- The result of these comparisons is an attention score for every other word in the sentence. These attention scores determine how much each of the other words should contribute to the next representation of “bank”.
- In the example, the disambiguating “river” could receive a high attention score when computing a new representation for “bank”. The attention scores are then used as weights for a weighted average of all words’ representations which is fed into a fully-connected network to generate a new representation for “bank”, reflecting that the sentence is talking about a river bank.

COREFERENCE RESOLUTION

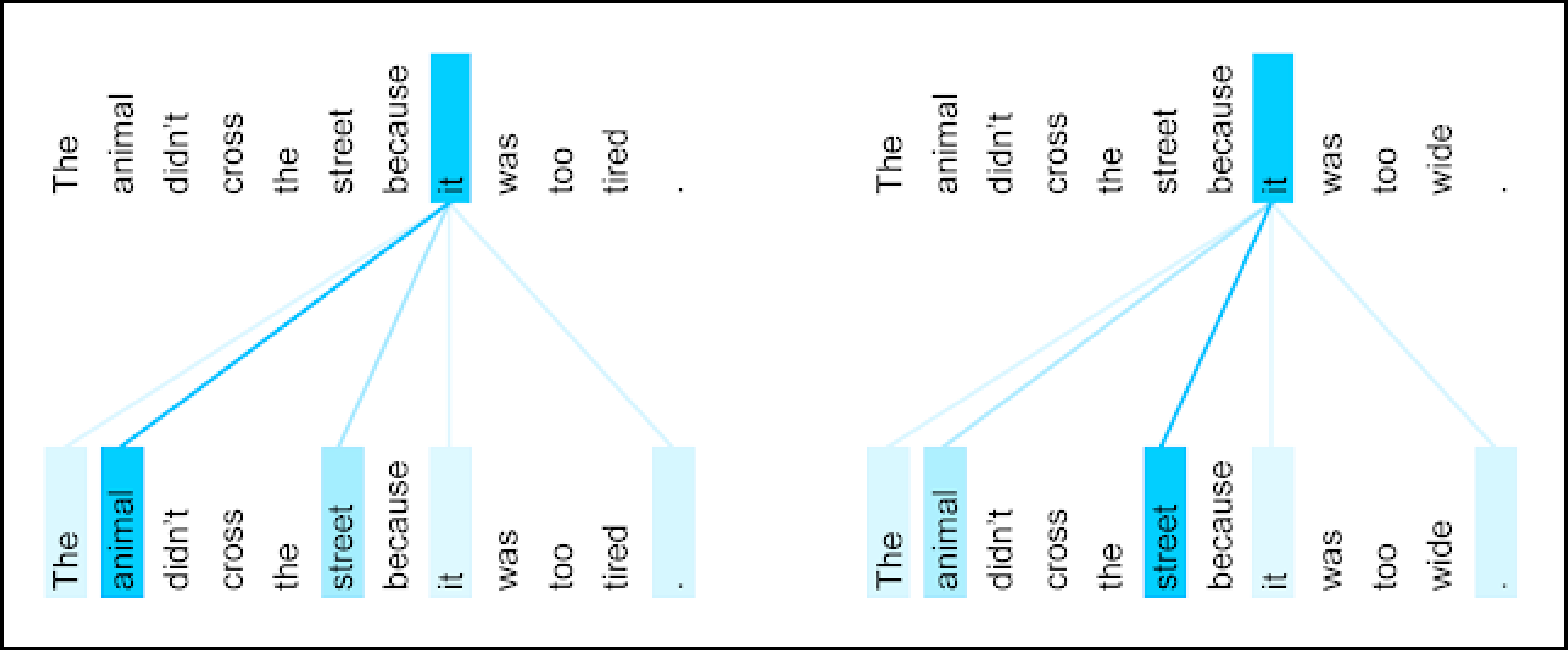
Consider the following sentences and their French translations:



The animal didn't cross the street because it was too tired.
L'animal n'a pas traversé la rue parce qu'il était trop fatigué.

The animal didn't cross the street because it was too wide.
L'animal n'a pas traversé la rue parce qu'elle était trop large.

COREFERENCE CONTEXT



- The encoder self-attention distribution for the word “it” from the 5th to the 6th layer of a Transformer trained on English to French translation (one of eight attention heads).

https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

COREFERENCE IN TRANSLATION

- It is obvious to most that in the first sentence pair “it” refers to the animal, and in the second to the street.
- When translating these sentences to French or German, the translation for “it” depends on the gender of the noun it refers to - and in French “animal” and “street” have different genders. In contrast to the current Google Translate model, the Transformer translates both of these sentences to French correctly.

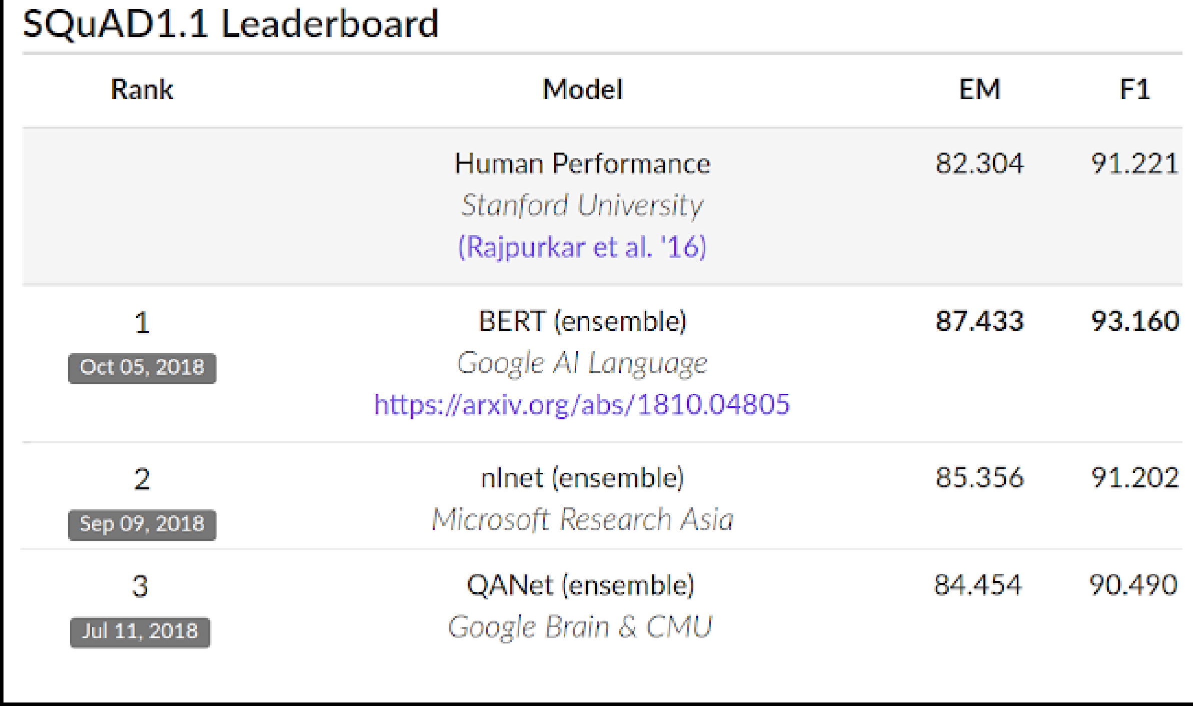
TWO BERTS

- There are four variants of BERT available (Each 'cased' or 'uncased':
  - · BERT Base: 12 layers (transformer blocks), 12 attention heads, and 110 million parameters
  - · BERT Large: 24 layers (transformer blocks), 16 attention heads and, 340 million parameters

BERT-Base was trained on 4 cloud TPUs for 4 days and BERT-Large was trained on 16 TPUs for 4 days.

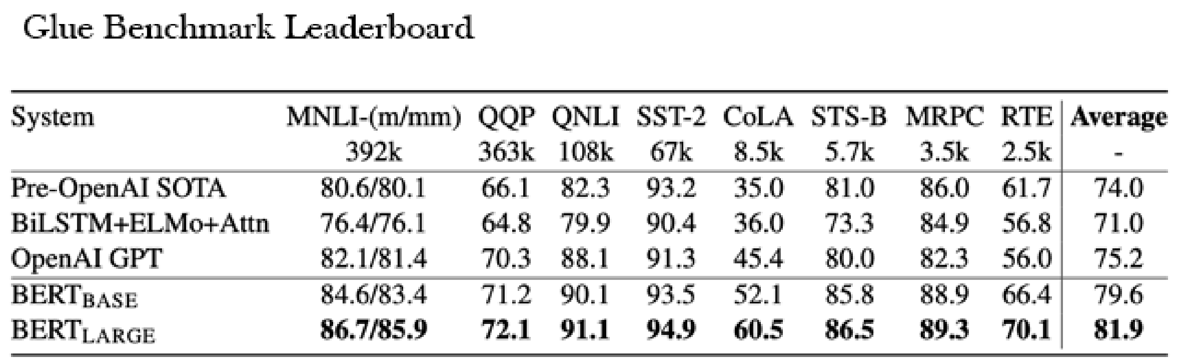
BERT IN USE

- On SQuAD v1.1 BERT achieves 93.2% F1 score (a measure of accuracy), surpassing the previous state-of-the-art score of 91.6% and human-level score of 91.2%: BERT also improves the state-of-the-art by 7.6% absolute on the very challenging GLUE benchmark, a set of 9 diverse Natural Language Understanding (NLU) tasks.



SQuAD1.1 Leaderboard

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar et al. '16)	82.304	91.221
1 Oct 05, 2018	BERT (ensemble) *Google AI Language* https://arxiv.org/abs/1810.04805	**87.433**	**93.160**
2 Sep 09, 2018	nlnet (ensemble) *Microsoft Research Asia*	85.356	91.202
3 Jul 11, 2018	QANet (ensemble) *Google Brain & CMU*	84.454	90.490



Glue Benchmark Leaderboard

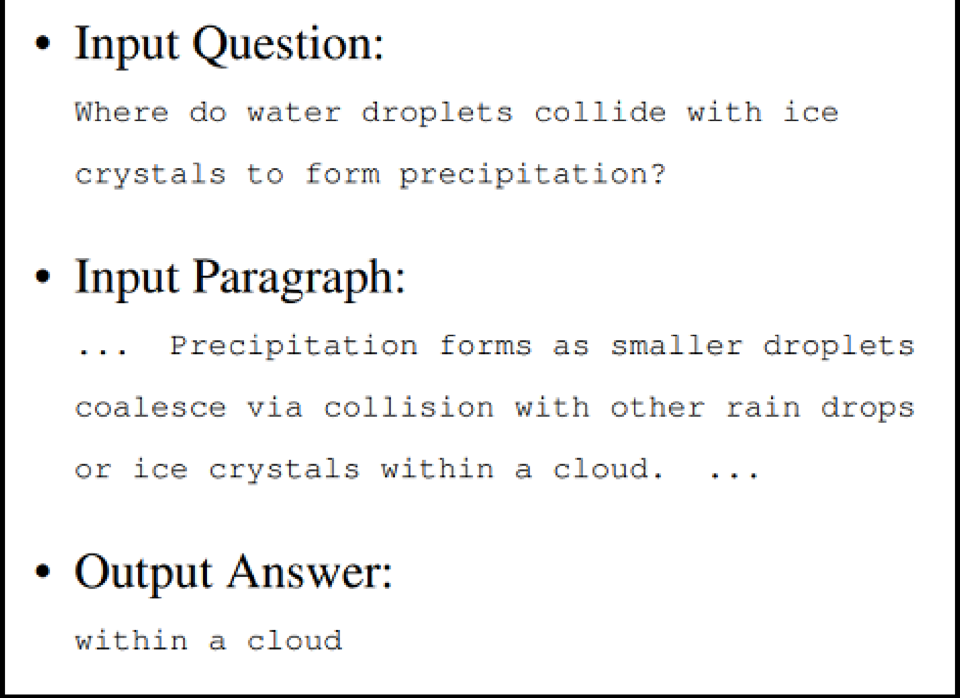
System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
BERT$_{BASE}$	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
BERT$_{LARGE}$	**86.7/85.9**	**72.1**	**91.1**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**81.9**

FINE TUNING BERT

- To fine-tune the original model based on a specific dataset, just add a single layer on top of the core model.
- To create a question and answer system:
  - This is a prediction task — on receiving a question as input, the goal of the application is to identify the right answer from some corpus.
  - Given a question and a context paragraph, the model predicts a start and an end token from the paragraph that most likely answers the question.
  - This means that using BERT a model for our application can be trained by learning two extra vectors that mark the beginning and the end of the answer.

QUESTION-ANSWER

- Just like sentence pair tasks, the question becomes the first sentence and paragraph the second sentence in the input sequence. However, this time there are two new parameters learned during fine-tuning: a **start vector** and an **end vector.**
- In the fine-tuning training, most hyper-parameters stay the same as in BERT training; the paper gives specific guidance on the hyper-parameters that require tuning.



- Input Question:

  `Where do water droplets collide with ice crystals to form precipitation?`

- Input Paragraph:

  `...  Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud.  ...`

- Output Answer:

  `within a cloud`

NOTE

- In case we want to do fine-tuning, we need to transform our input into the specific format that was used for pre-training the core BERT models, e.g., we would need to add special tokens to mark the beginning ([CLS]) and separation/end of sentences ([SEP]) and segment IDs used to distinguish different sentences — convert the data into features that BERT uses.