June 30 2022

2022 SNU AI 전문가 과정

최신 자연어 처리 기법

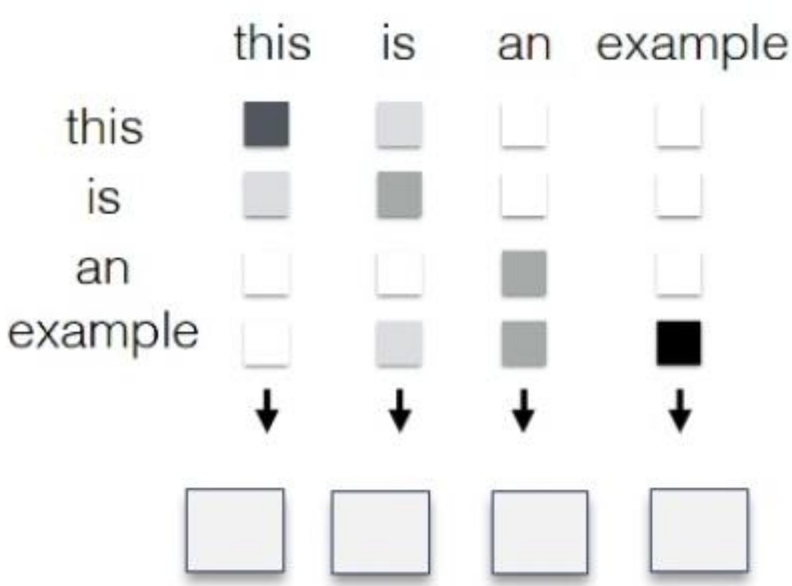
Kyomin Jung

Department of Electrical and Computer Engineering
Seoul National University

Table of Contents

Remind: Self-Attention

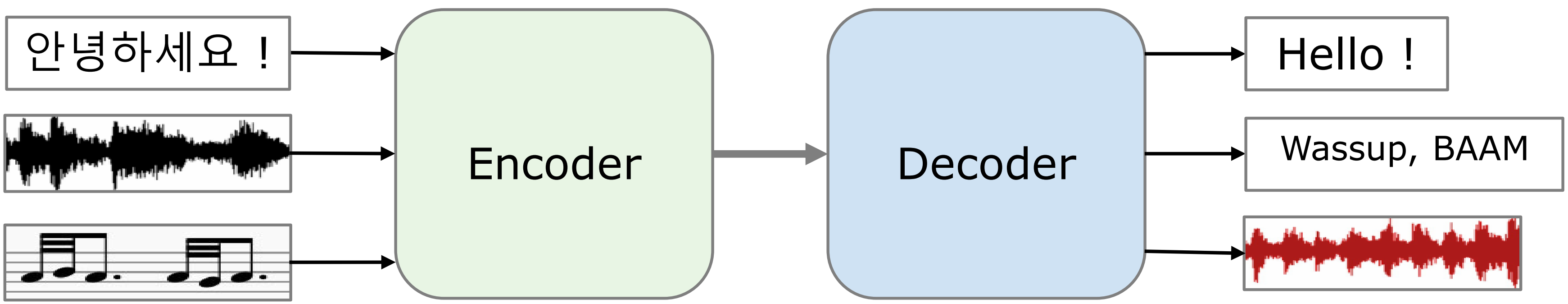
- Self-attention (intra-attention) relates different positions of a single sequence.
  - Each element in the sentence attends to other elements from the same sentence -> context-sensitive encodings
  - Self-attention enhances the automatic understanding of text



Transformer

A. Vaswani et al., "Attention Is All You Need", NIPS 2017

- The transformer has an encoder-decoder structure, and it highly utilizes attention mechanisms.
- Applied to the sequence transduction tasks
  - Machine translation, Dialogue system
  - Speech recognition, Text-to-speech transformation
  - Music generation



RNN vs. Transformer

Better
Faster
Stronger

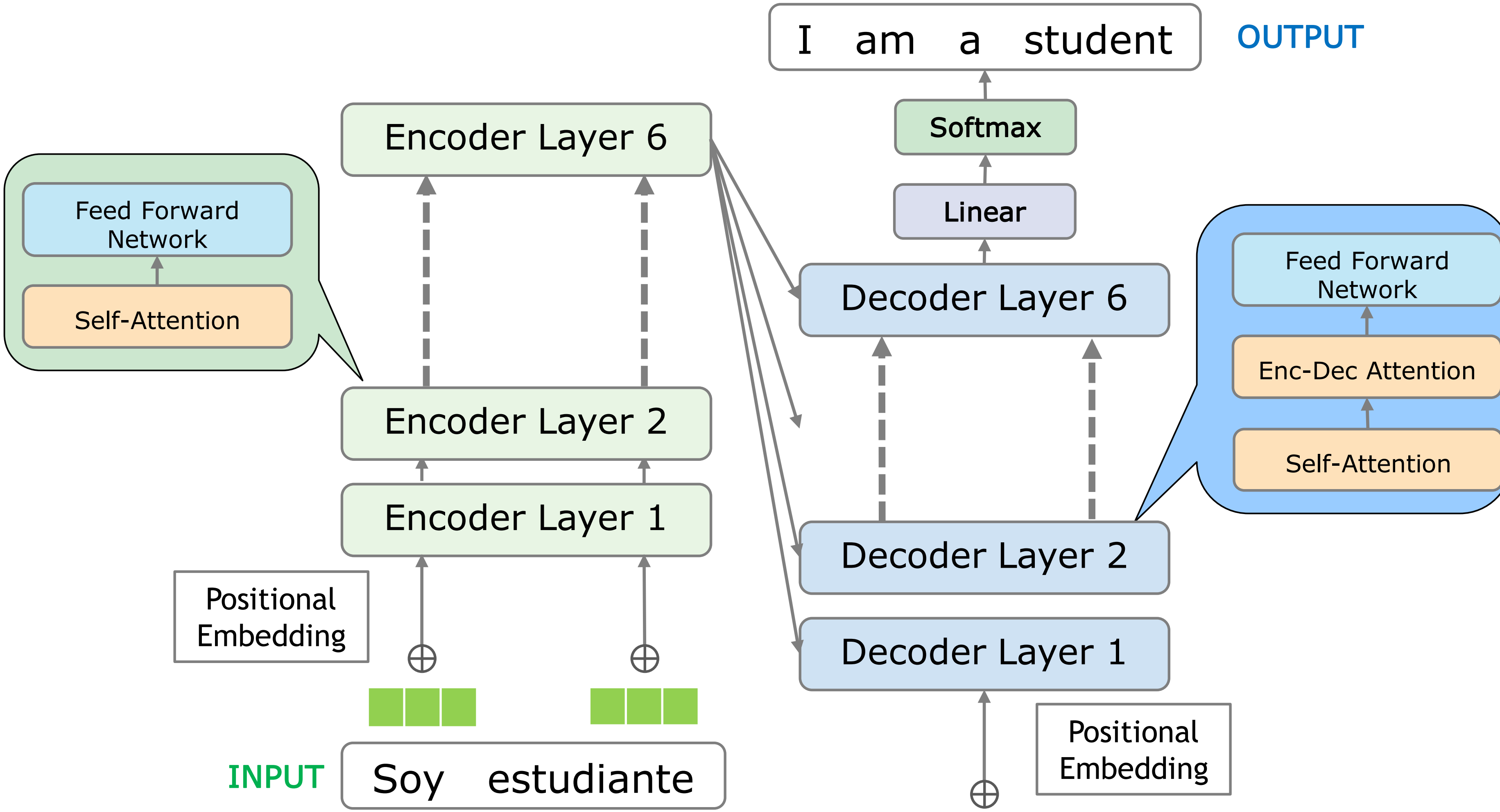
- **RNN**
  - **Recurrent module** : Suitable for processing **variable length** representations
- **Transformer**
  - **Self-attention mechanism** : Consists of **fixed-size** matrix multiplications

RNN	Transformer
Hard to parallelize efficiently (sequential nature)	Easy to parallelize → Fast
No explicit modeling of long-range dependencies	Complete connections between consecutive layers
Maximum path length : $O(n)$	Maximum path length : $O(1)$

n : sequence length

distance between any pair of input and output positions in networks

Transformer : Model Overview

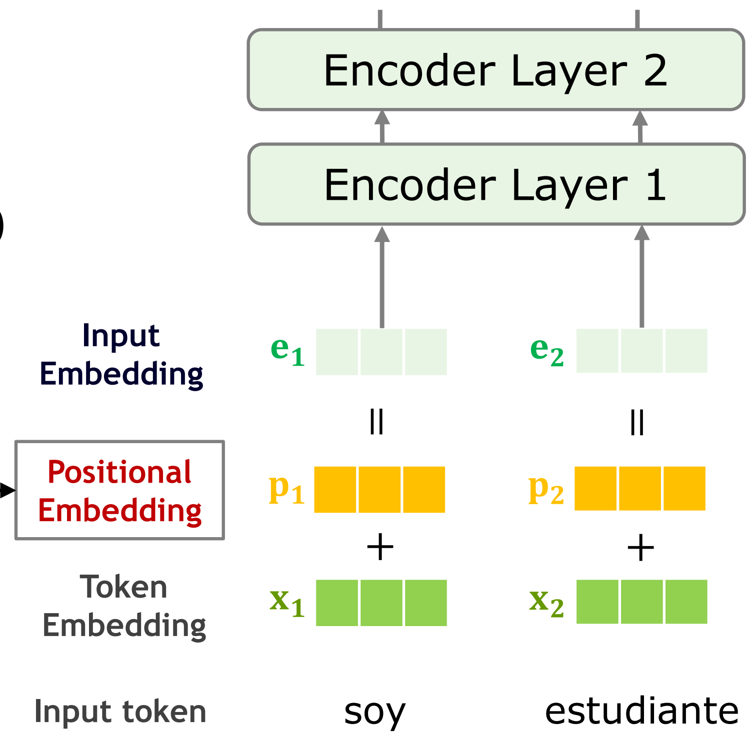


Transformer : Positional Embedding

- **Positional Embedding**
  - Information about the relative or absolute position of the tokens in the sequence (to utilize the order of the sequence)

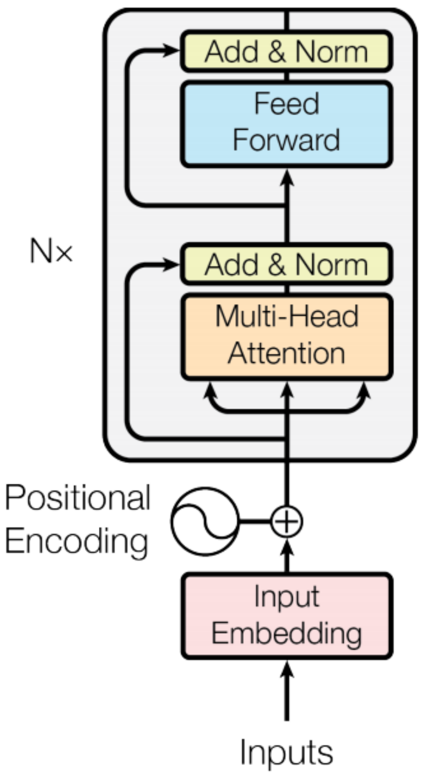
$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

(pos = position, i = dimension)



Transformer : Encoder

- Encoder



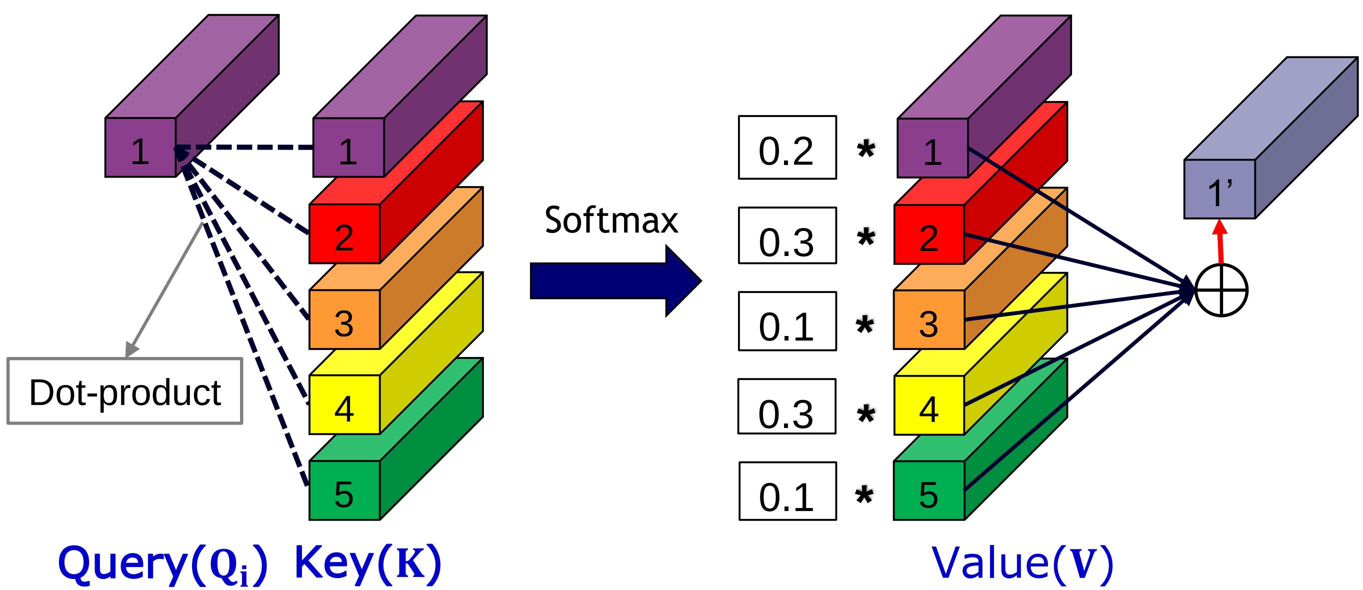
- A stack of N=6 identical layers
- Each layer has two sub-layers
  - (1) Multi-Head Self-attention
  - (2) Position-wise Feed Forward Network
- The output of each sub-layer
  = LayerNorm (x + Sublayer(x))
  residual connection
- All of the queries, keys, and values come from the output of the previous layer (Q=K=V)

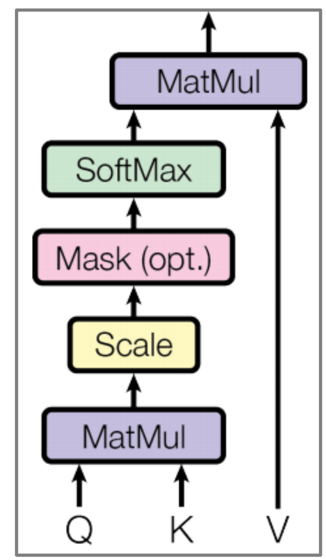
Transformer : Self-Attention

- **Self-attention (Scaled Dot-product Attention)**
  - **Computing a representation of a sequence considering the relationship between different positions**

The animal didn't cross the …
[1] [2] [3] [4] [5]

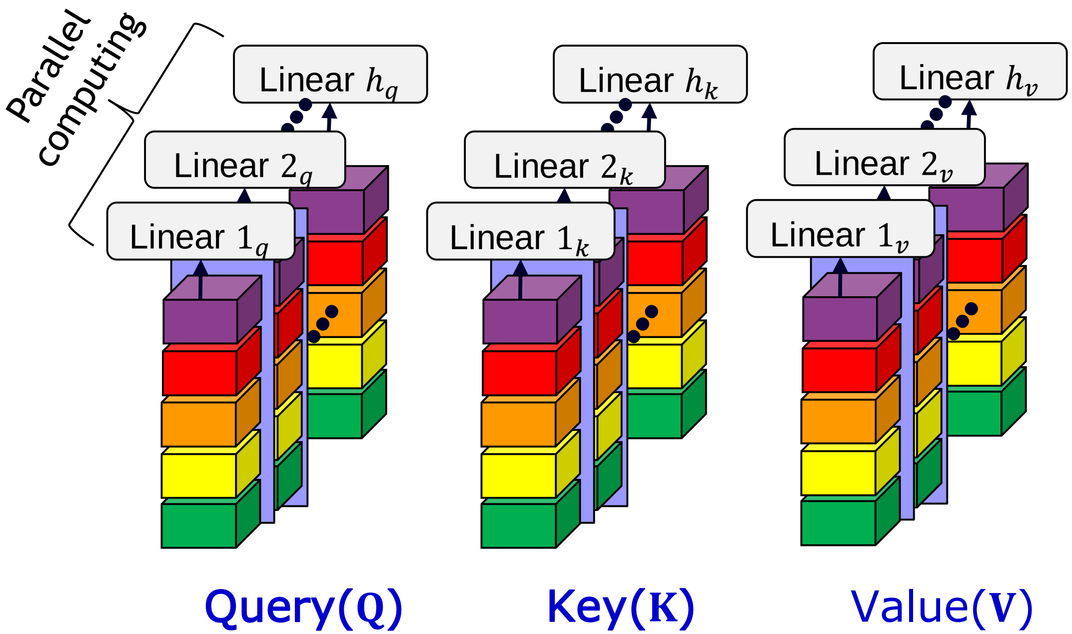
$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



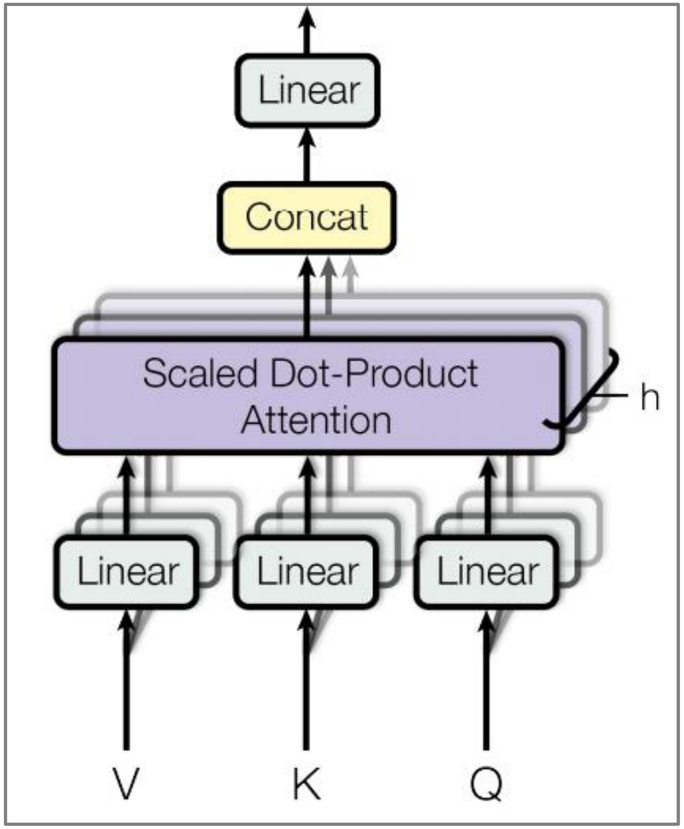


Transformer : Self-Attention

- **Multi-Head Attention**
  - Apply the "Scaled Dot-Product Attention" several times with different linear projections

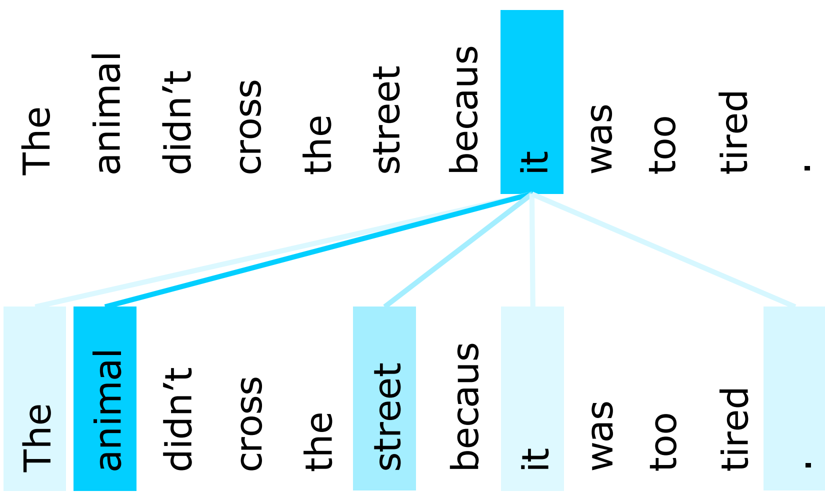


$$\mathrm{MultiHead}(Q,K,V) = \mathrm{Concat}(\mathrm{head_1},\ldots,\mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

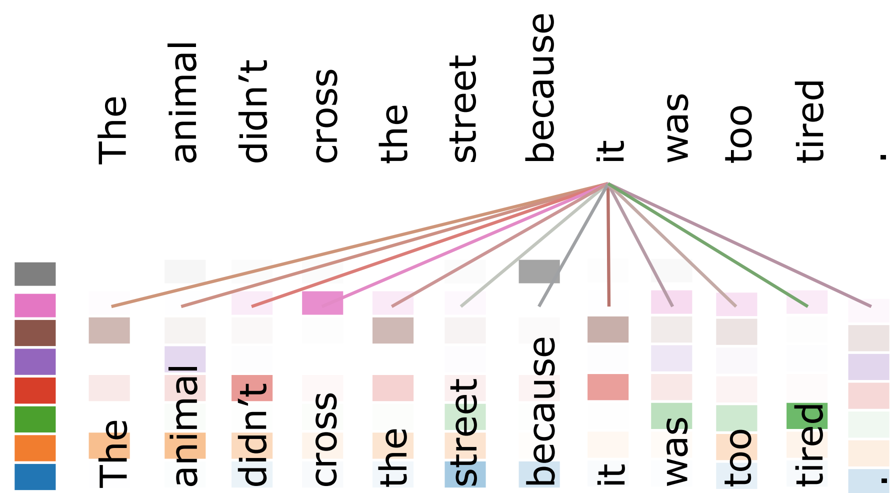


Transformer : Self-Attention

- **Self-Attention Visualization**



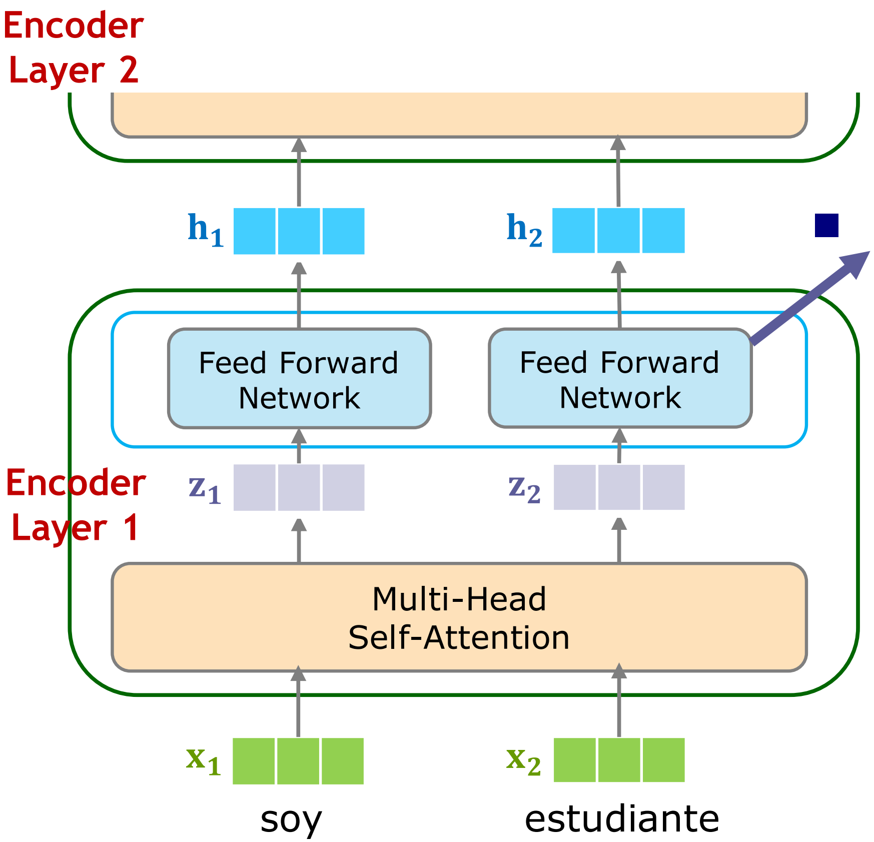
Single-Head Self-Attention



Multi-Head(8) Self-Attention

Transformer : Encoder

- Encoder

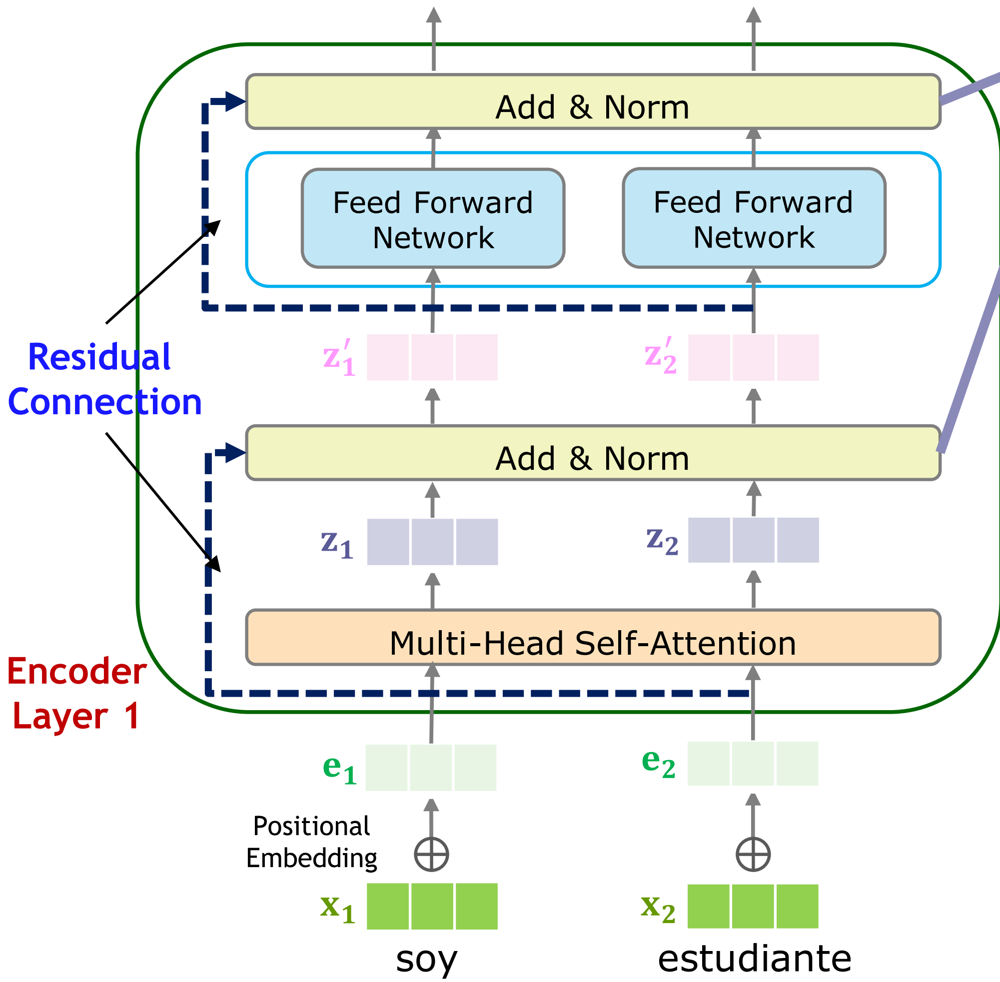


- Point-wise Feed Forward Network
  - Applied to each position separately and identically (different parameters from layer to layer)
  - Two linear transformations with ReLU activation in between

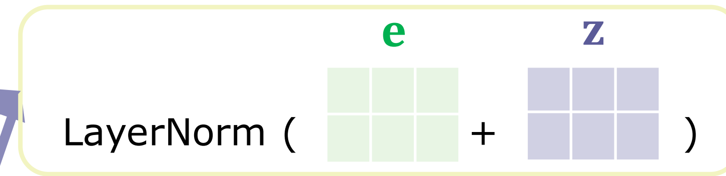
$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

Transformer : Residual Connection

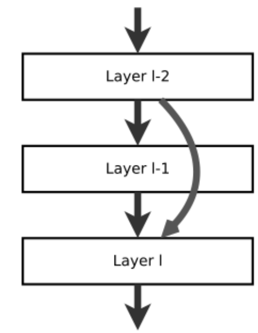
■ Residual Connection



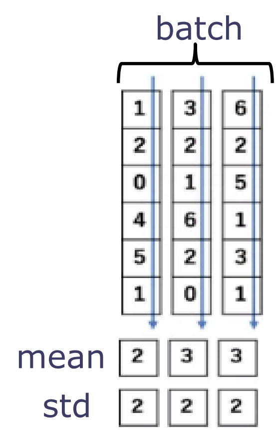
LayerNorm (x + Sublayer(x))



* Residual Connection
- Skip connection
- Short-cuts to jump over some layers



* Layer Normalization
- Normalizes the inputs across the features



batch-dim

feature-dim

$$\mu_i = \frac{1}{m}\sum_j^m x_{ij}$$

$$\sigma_i = \frac{1}{m}\sum_j^m (x_{ij} - \mu_i)^2$$

$$\widehat{x_{ij}} = \frac{x_{ij} - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}}$$

Transformer : Decoder

- **Decoder** (almost the same as encoder)
  - A stack of N=6 identical layers
  - Each layer has three sub-layers

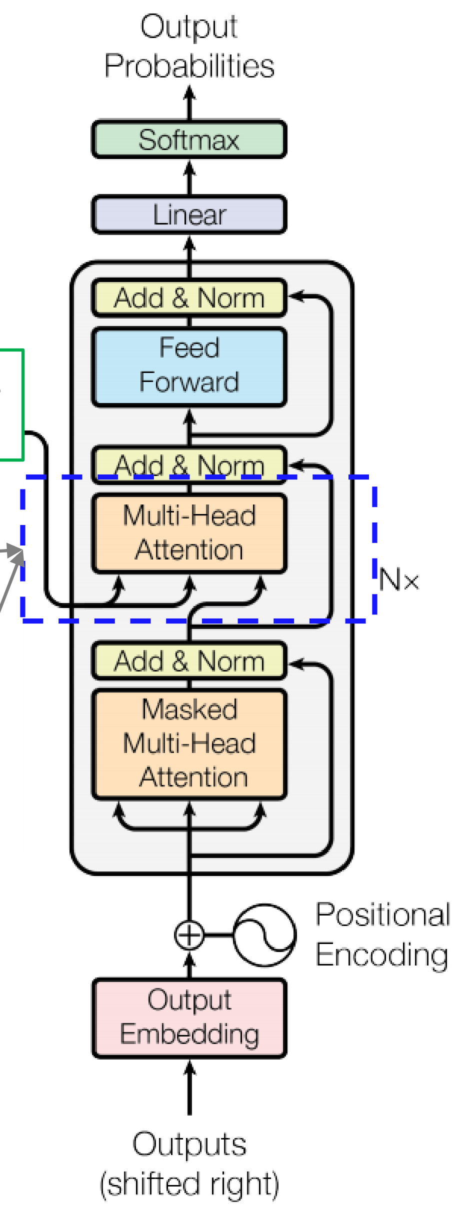
    (1) Multi-Head Self-attention

    (2) Multi-Head Attention over the output of the encoder stack (enc-dec attention)

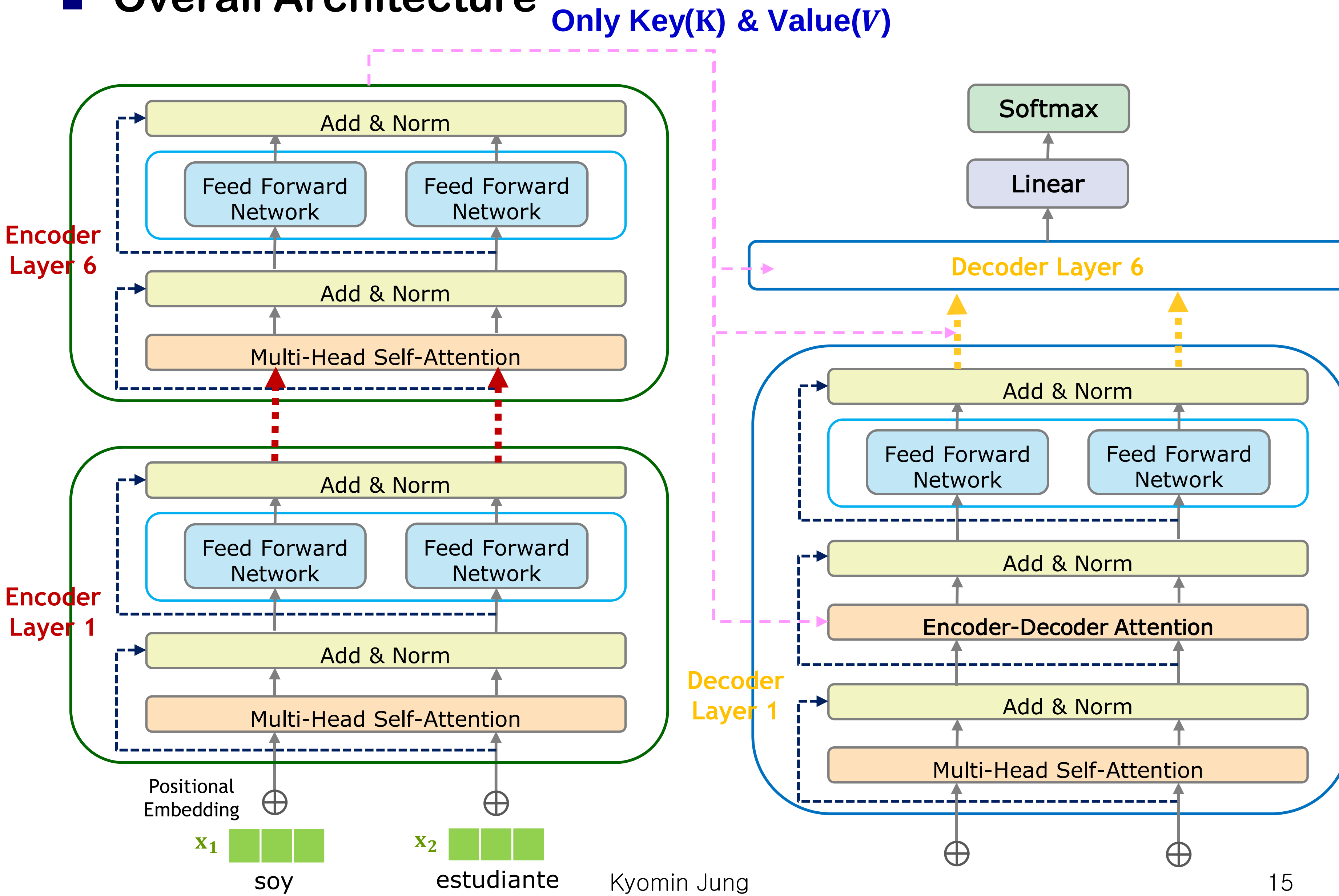
    (3) Position-wise Feed Forward Network

Encoder's output

The queries Q come from the previous decoder layer, and the key K and value V come from the output of the encoder



■ Overall Architecture



Transformer : Training

- Dataset
  - WMT 2014 English-German dataset
    - 4.5M sentence pairs
    - 37k word-piece tokens
  - WMT 2014 English-French dataset
    - 36M sentence pairs (Larger!)
    - 32k word-piece tokens

- Training details
  - Byte-pair encoding
  - ADAM optimizer with learning rate warmup
  - Batching: approximately 25000 source tokens, 25000 target tokens
  - Regularization: Dropout, LayerNorm
  - Parameter settings
    - base: 512 hidden size, 8 multi-heads, 0.1 dropout probability, 100k steps
    - big: 1024 hidden size, 16 multi-heads, 0.3 dropout probability, 300k steps
    - big model size $\sim 2 \times$ base model size

Transformer: Experiments

- **Machine Translation: WMT-2014**
  - Transformer (big) achieved state-of-the-art (SOTA) performance
  - Transformer (base) achieved similar performance as RNN-based model (GNMT) with 3x faster training time

Model	EN-DE	EN-FR	FLOPs ($\times 10^{18}$)
GNMT (RNN-based)	24.6	39.9	23.0
ConvSeq2Seq (CNN-based)	25.2	40.5	9.6
Transformer (base)	27.3	38.1	3.3
Transformer (big)	28.4	41.0	23.0

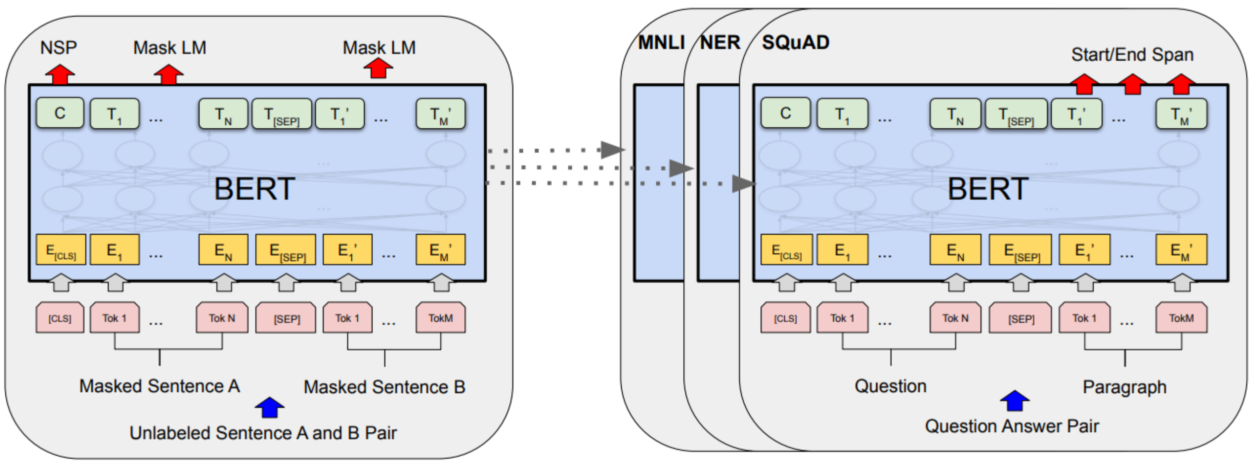
  - Evaluation metric: BLEU Score (higher is better)
  - Computational cost: floating point operations per second (FLOPs)

BERT: Bidirectional Encoder Representations from Transformers

J. Devlin et al., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”

- BERT
  - Language model designed for bidirectional representation
  - Pre-training and Fine-tuning approaches
  - Achieves the state of the art for eleven NLP tasks

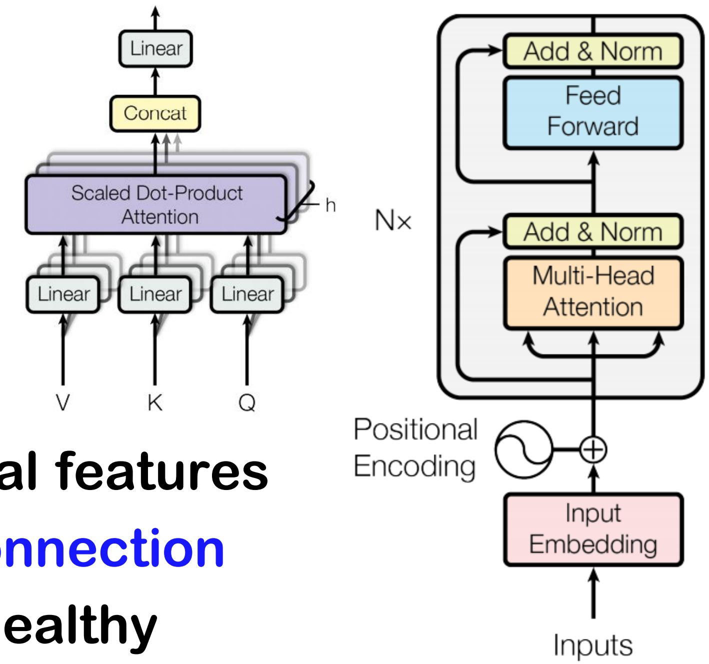
    - QA, language inference, sentiment classification, named entity recognition…



Pre-training   Fine-Tuning

BERT: Model Architecture

- BERT's model architecture is
  a multi-layer Transformer encoder



- Multi-Head Self-attention
  * Models bidirectional context
- Point-wise Feed-forward layers
  * Computes non-linear hierarchical features
- Layer Normalization & Residual Connection
  * Makes training deep networks healthy
- Positional embedding
  * Allows model to learn relative positioning

BERT: Language Modeling Process

(1) Pre-training

Training on large amounts of data (Language modeling)

*Model

BERT

*Dataset

BooksCorpus (800M words)

WIKIPEDIA
Die freie Enzyklopädie

Wikipedia (2,500M words)

*Objective

(1) Predict the masked word
(2) Next sentence prediction

(2) Fine-tuning (supervised)

Training on a specific downstream task with a labeled dataset

*Model

Classifier

75% Spam
25% Not Spam

Just one additional output layer

BERT (Pre-trained)

*Dataset

Email content	Label
Buy one, get one free	Spam
Dear Harry, Hi this is..	Not Spam

Pre-training Tasks (1) : Masked LM

- Masked Language Model (MLM)
  - Task that enables the BERT model to learn bidirectional context information and grammars of the language
  - Mask out 15% of the input words, and then predict the masked word
  - True labels can be easily obtained from the training data text corpus

the man went to the [MASK] to buy a [MASK] of milk

Predict Predict

“store” “gallon”

Pre-training Tasks (1) : Masked LM

- Mismatch between pre-training and fine-tuning
  - [MASK] token never seen at fine-tuning

Solution?

- When masking out the input words

80% of the time : replace with [MASK]	10% of the time : replace with random word	10% of the time : keep same
went to the store → went to the [MASK]	went to the store → went to the running	went to the store → went to the store

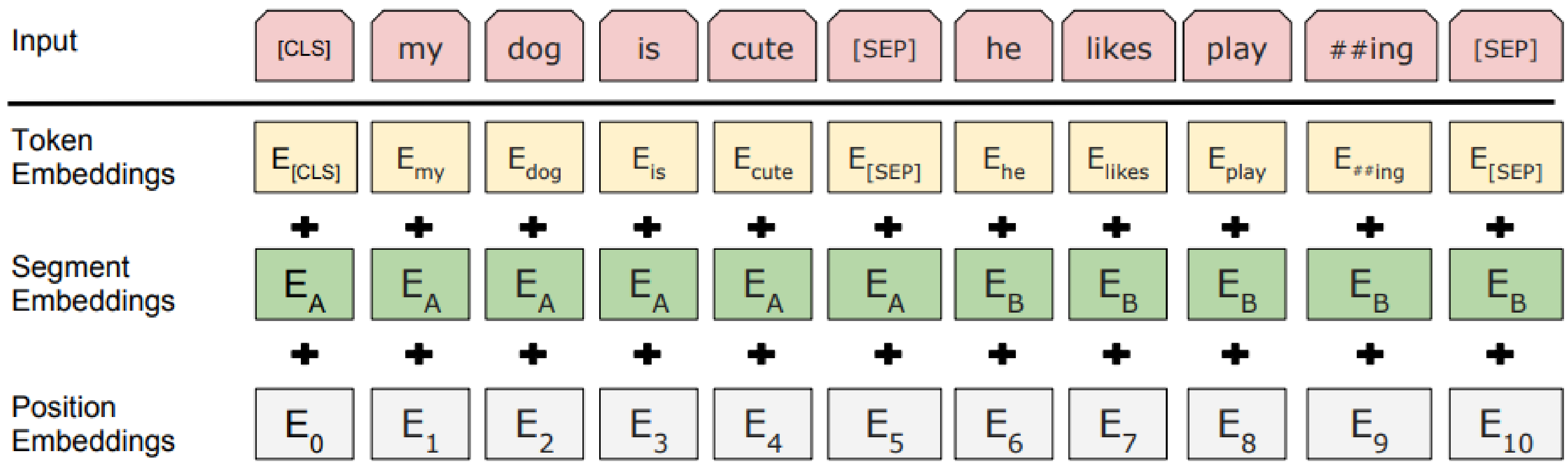
Pre-training Tasks (2) : Next Sentence Prediction

- Next Sentence Prediction
  - To learn relationships between sentences
  - Binary prediction task that predicts whether Sentence B is the actual next sentence that follows Sentence A
  - True labels can be easily obtained from the training data text corpus

Sentence A = The man went to the store
Sentence B = He bought a gallon of milk
Label = IsNextSentence

Sentence A = The man went to the store
Sentence B = Penguins are flightless
Label = NotNextSentence

BERT: Input Representation



- **WordPiece embeddings with a 30,000 token vocabulary**
- **Every sequence starts with [CLS] token**
  - *** [CLS]'s final state is used as the aggregate sequence representation for classification**
- **Input embedding =**
  Elementwise-Sum(Position + Segment + Token) embedding

BERT: Fine-tuning to specific task

- After pre-training,
  BERT is fine-tuned to specific NLP tasks

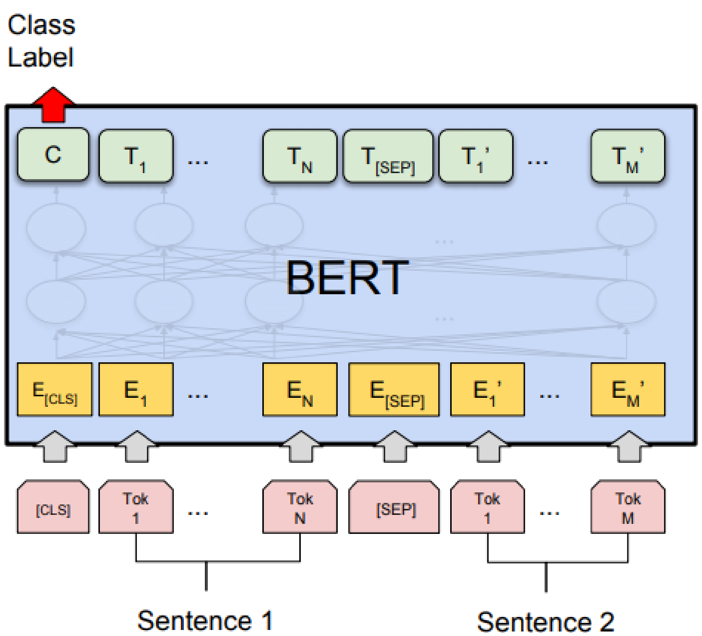
- Fine-tuning method : Incorporate BERT
  with one additional output layer for prediction
- The pre-trained BERT model parameters remain the same
  - Minimum architecture change:
    minimize the number of parameters needed to learn in the fine-tuning

BERT: Fine-tuning to specific task

* Sequence-level classification tasks

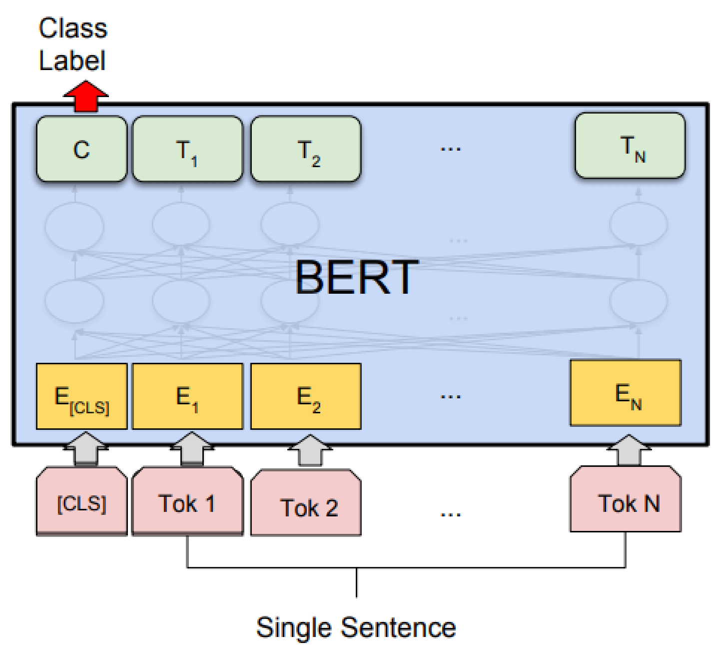
- **Sentence-pair classification**

- Use [SEP] token to separate sentence pair
- Use final hidden state C for representation



- **Singe Sentence classification**

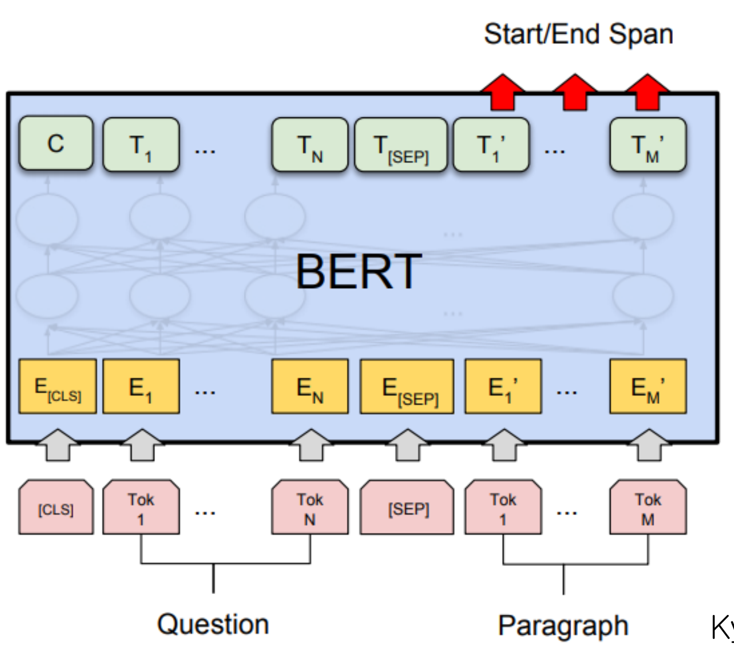
- Use final hidden state C for representation



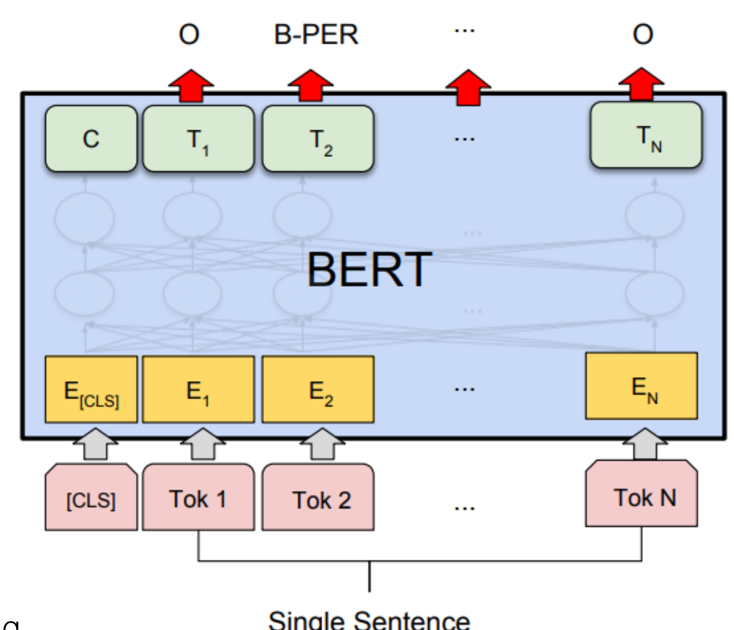
BERT: Fine-tuning to specific task

* Token-level classification tasks

- **Question Answering**
  - Use [SEP] token to separate question and paragraph
  - Use final hidden states of paragraph $T_1, \ldots, T_m$ for start/end span



- **Singe Sentence Tagging**
  - Use final hidden states of paragraph $T_1, \ldots, T_m$ for token classification



BERT: Experiments

- **Task: SQuAD Question Answering**
  - **Given a paragraph and question, locate the span in the paragraph that contains the answer**

System	Test EM	Test F1
Human	82.3	91.2
nlnet (Ensemble, *Leaderboard #1)	25.2	40.5
QANet (Ensemble, *Leaderboard #2)	27.3	38.1
$\text{BERT}_{\text{Large}}$ (Ensemble + **data aug.)	87.4	93.2

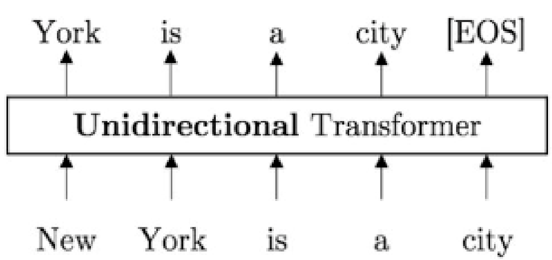
  - **BERT Outperforms previous SOTA and reaches Human performance!**

* Leaderboard from SQuAD official website on Oct 8th, 2018
** Data augmentation by jointly training using TriviaQA dataset

XLNet : Motivation

- Two notable objectives for language pretraining

Auto-regressive Language Modeling



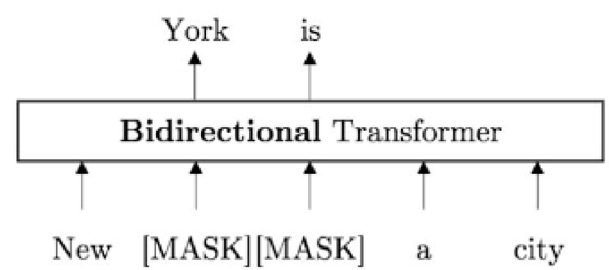
- Maximize likelihood by auto-regressive factorization

$$p(x) = \prod_{t=1}^{T} p(x_t | \mathbf{x}_{<t})$$

- No artificial symbols

! Unidirectional context

Denoising Auto-encoding (e.g. BERT)



! Independent prediction of masked tokens

$$p(x) = \prod p(x_{mask} | \hat{\mathbf{x}})$$

! Artificial symbol [MASK]

- Bidirectional context

XLNet : Methodology

- Proposed objective: Permutation Language Modeling
  - Combine the pros and remove the cons of the two objectives
  - Maximize the following likelihood of a sequence x:

$$p(x) = \mathbb{E}_{\mathbf{z}\sim Z_t}\left[\prod_{t=1}^{T} p\left(x_{z_t} \middle| \mathbf{x}_{\mathbf{z}_{<t}}\right)\right]$$

Z_T: set of all possible permutations

$\mathbf{z}$: factorization order sampled from Z_t

New york is a city
1 2 3 4 5
New york is city a
1 2 3 5 4
New york a is city
1 2 4 3 5
…
city a is york New
5 4 3 2 1

$|Z_T| = T!$

- **Maximize likelihood by factorization**
- **No artificial symbols**
- **Bidirectional context**

XLNet : Implementation

- **Model architecture: multi-layer Transformer encoder (same as BERT)**
  - **predict tokens by factorization order**

Contextual representation from Transformer encoder

$$p_\theta(X_i = x | \mathbf{x}_{\mathbf{z}_{<t}}) = \frac{\exp\left(e(x)^T h_\theta(\mathbf{x}_{\mathbf{z}_{<t}})\right)}{\sum_{x'} \exp\left(e(x')^T h_\theta(\mathbf{x}_{\mathbf{z}_{<t}})\right)}$$

word embedding

Sentence $\mathbf{x}$ = "New York is a city"

1 2 3 4 5

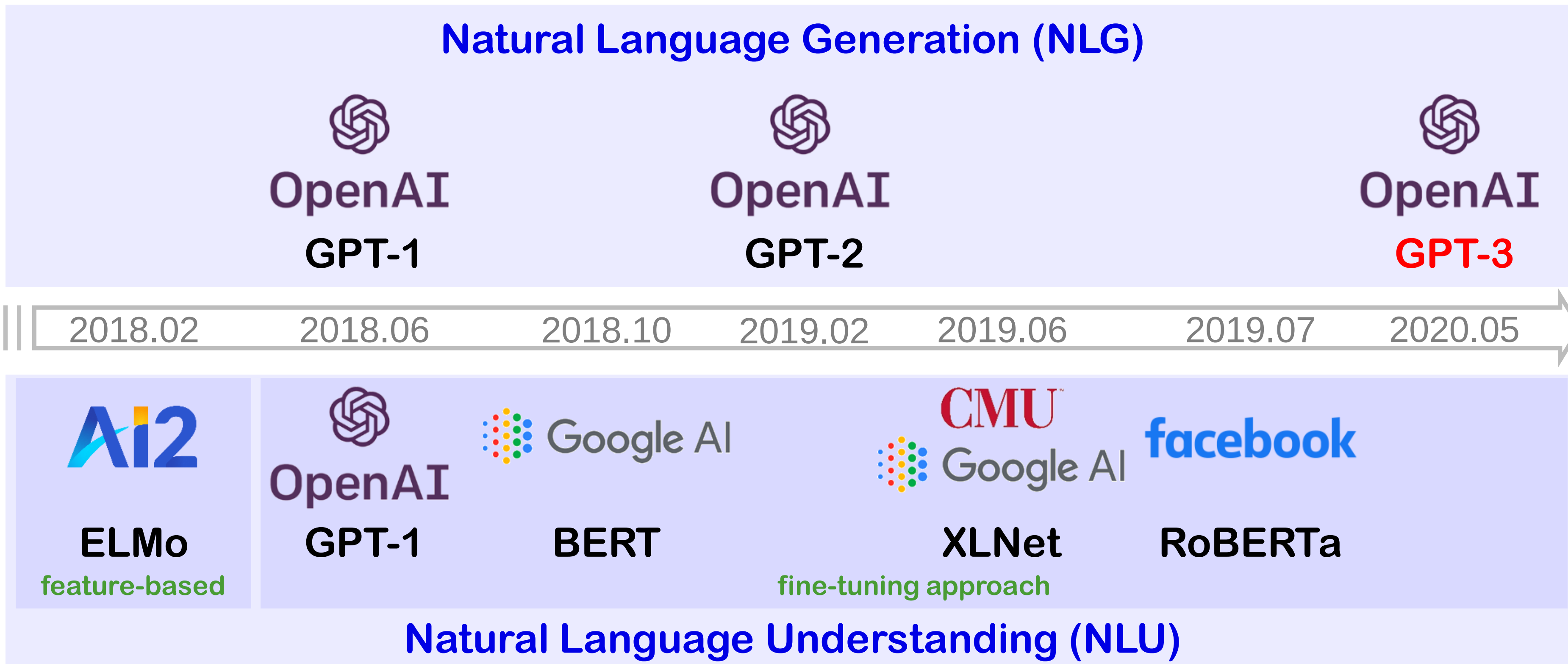
Factorization order $\mathbf{z} = (3, 2, 4, 1, 5)$

⬇

P(is) P(York | is) P(a | is, York) P(New | is, York, a) P(city | is, York, a, New)

GPT: What is Different from Others

- **Generation VS Understanding**
  - OpenAI's GPT is an unidirectional Language Model(LM)
    - GPT is good for text generation tasks because of the auto-regressive LM
  - On the other hand, BERT and XLNet are bidirectional LMs
    - They are good for natural language understanding (NLU) tasks



GPT: Generative Pre-Training

A. Radford et al., "Improving Language Understanding by Generative Pre-Training"

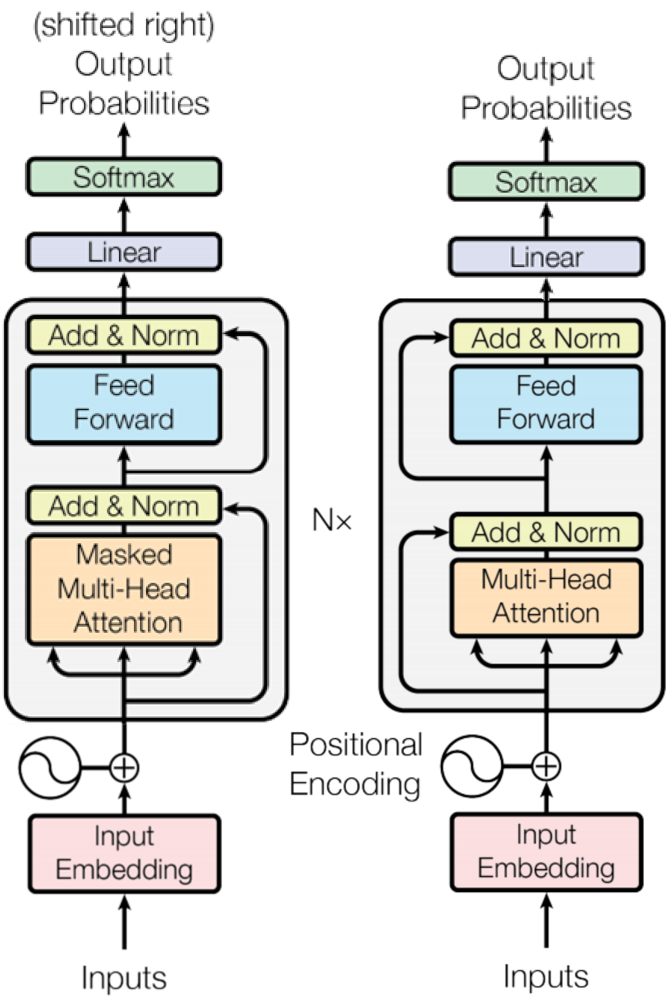
- Significance of the first GPT
  (known as GPT-1 now)
  - The first successful model of the pre-training and (then) fine-tuning approach using large model with large corpus
  - GPT-1 outperforms the previous state-of-the-arts on 9 out of 12 tasks

- Two phase of training GPT-1
  (similar to BERT)
  - Pre-training:
    LM is trained to predict the next word using the previous context, which is an auto-regressive (generative) language modeling
  - Fine-tuning:
    Almost all layers of pre-trained LM is transferred into any downstream task with minimal task-specific modification

GPT: Comparison with BERT

- **Pre-training objective**
  - GPT: "Next Word Prediction"
    auto-regressive language modeling
  - BERT: "Masked Word Prediction"
    masked language modeling
    + "Next Sentence Prediction"

- **Performances on NLU tasks**
  - ELMo < GPT-1:
    - The betterment of fine-tuning approach than feature-based approach
  - GPT-1 < BERT:
    - In natural language understanding tasks, there is a fundamental limitation of auto-regressive language modeling:
    - GPT-1 uses only unidirectional context, while BERT uses full contextual information.



GPT model BERT model

GPT: After BERT Beats GPT-1

- Different goal of GPT-3 (and GPT-2)
  - They have focused on enhancing language model
    - Using large and various corpus and model sizes for LM training

	GPT-1	GPT-2	GPT-3
`dataset_size`	1B words (BooksCorpus)	10B words (WebText)	300B (Mixture of corpus)
`max_token_num`	512	**1024**	**2048**
`batch_size`	64	512	0.5 - 3.2M
`model_size`	0.1B params 12 layers)	{0.1 - 1.5}B params {12-48} layers	{0.1 - 175}B params {12-96} layers

  - They have applied GPT to unsupervised learning tasks
    - They have explored the few-shot behaviors
    - GPT-2 and GPT-3 are NOT fine-tuned to any target tasks

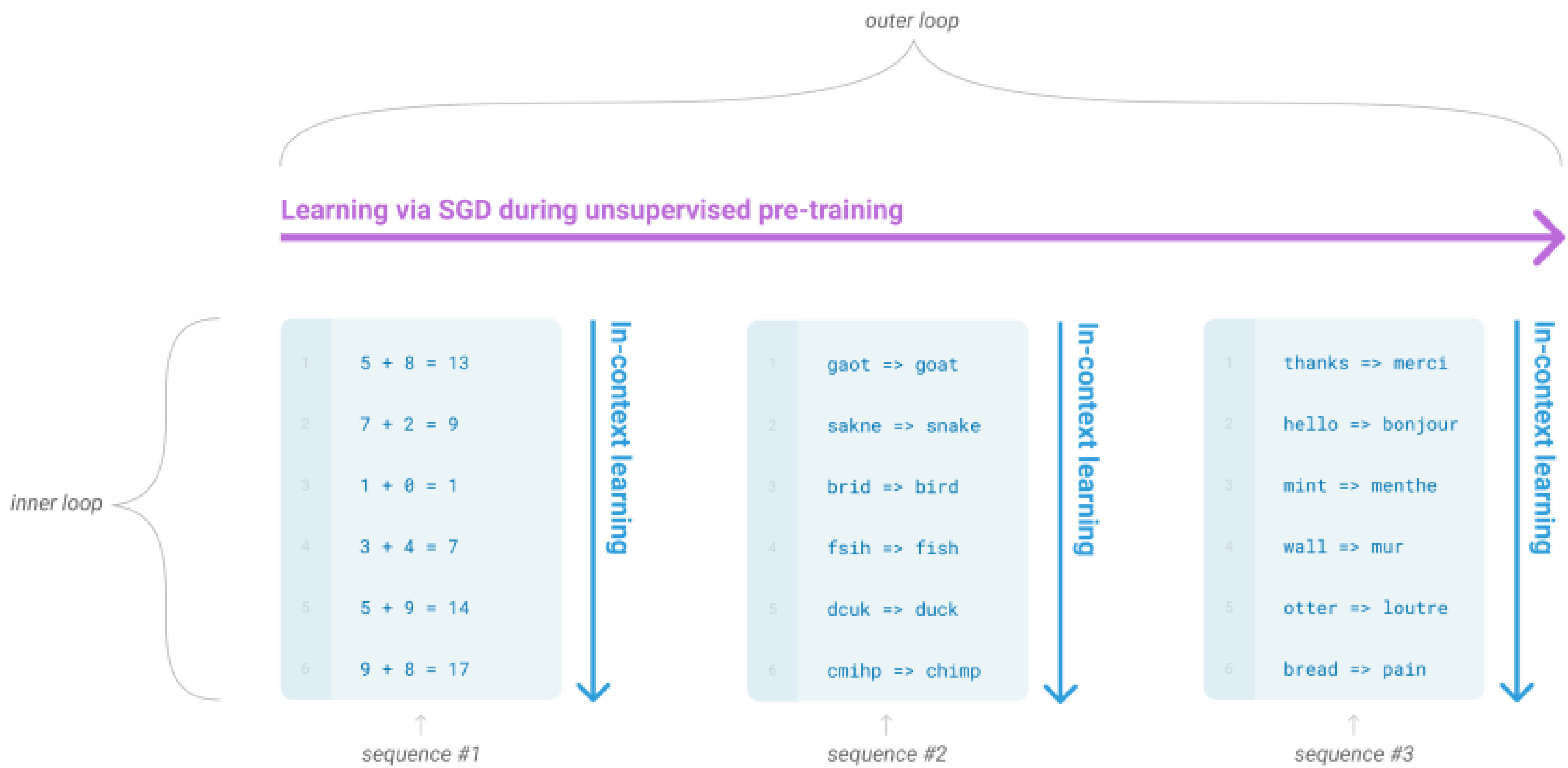
GPT-3: Text Generation Example

- News Article Generation using GPT-3

Title: United Methodists Agree to Historic Split
Subtitle: Those who oppose gay marriage will form their own denomination
Article: After two days of intense debate, the United Methodist Church has agreed to a historic split - one that is expected to end in the creation of a new denomination, one that will be "theologically and socially conservative," according to The Washington Post. The majority of delegates attending the church's annual General Conference in May voted to strengthen a ban on the ordination of LGBTQ clergy and to write new rules that will "discipline" clergy who officiate at same-sex weddings. But those who opposed these measures have a new plan: They say they will form a separate denomination by 2020, calling their church the Christian Methodist denomination.
The Post notes that the denomination, which claims 12.5 million members, was in the early 20th century the "largest Protestant denomination in the U.S.," but that it has been shrinking in recent decades. The new split will be the second in the church's history. The first occurred in 1968, when roughly 10 percent of the denomination left to form the Evangelical United Brethren Church. The Post notes that the proposed split "comes at a critical time for the church, which has been losing members for years," which has been "pushed toward the brink of a schism over the role of LGBTQ people in the church." Gay marriage is not the only issue that has divided the church. In 2016, the denomination was split over ordination of transgender clergy, with the North Pacific regional conference voting to ban them from serving as clergy, and the South Pacific regional conference voting to allow them.

GPT-3: What is Few-Shot Learning

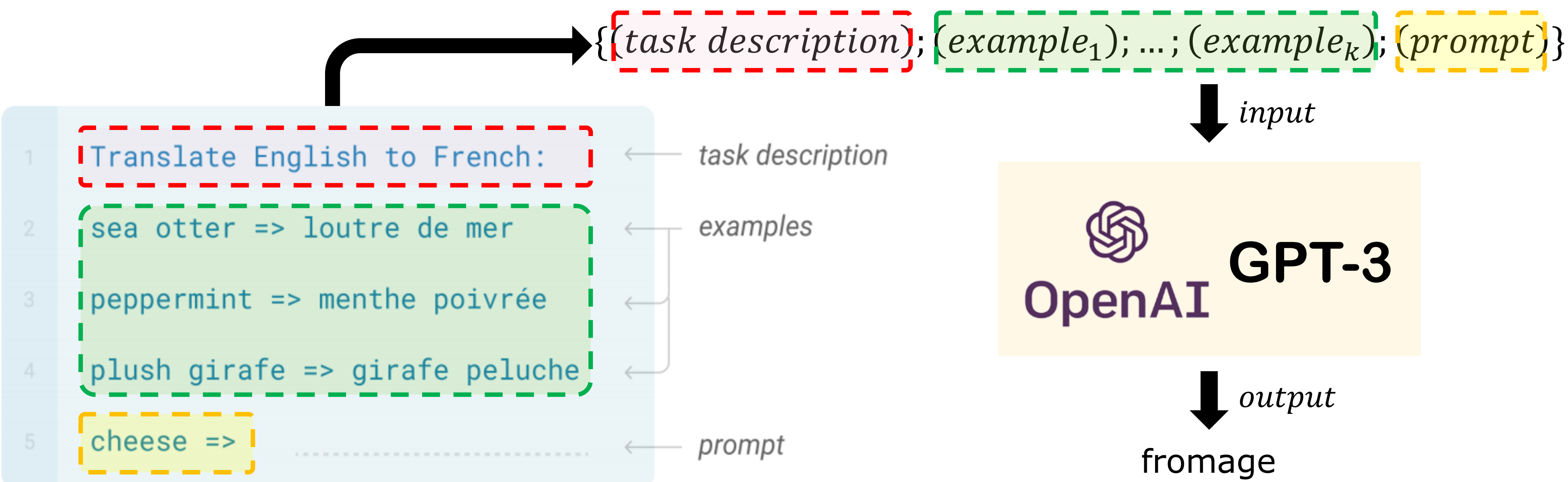
- **Language model meta-learning**
  - During unsupervised pre-training, GPT-3 develops a broad set of skills and pattern recognition abilities.



GPT-3: What is Few-Shot Learning

- In-context learning
  - Contrary to the traditional fine-tuning, GPT-3 does NOT change the model parameters for specific tasks
  - Their few-shot setting uses several paired examples of the task in addition to the task description

- Example of few-shot setting on the machine translation task:
  - The model sees task description and a few examples of the task

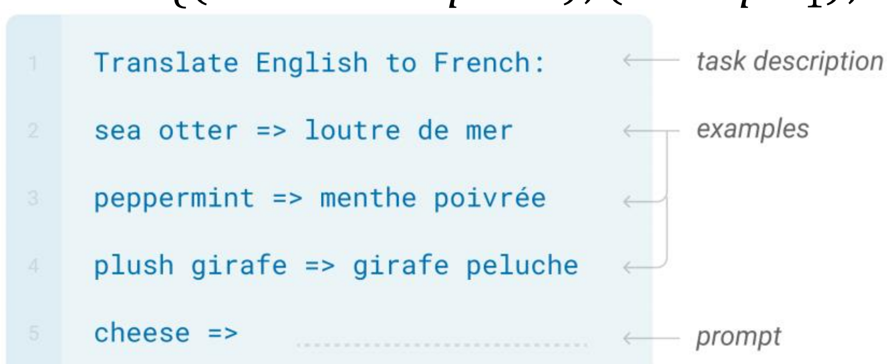


GPT-3: What is Few-Shot Learning

- Three settings for in-context learning
  - **Zero-shot:** $\{(task\ description);(prompt)\}$

  - **One-Shot:** $\{(task\ description);(example_1);(prompt)\}$

  - **Few-Shot:** $\{(task\ description);(example_1);\ldots;(example_k);(prompt)\}$



GPT-3: Experiments

- GPT is good at LM-friendly tasks
  - LAMBADA task
    (Language Modeling Broadened to Account for Discourse Aspects)
    ex)

Alice was friends with Bob. Alice went to visit her friend ______.	→ Bob
George bought some baseball equipment, a ball, a glove, and a ______.	→

  - Accuracy (%):
    - State-of-the-art (SOTA) at 2019.02: 59.2
    - GPT-2: 63.2

    - previous SOTA at 2020.06: 68.0
    - GPT-3 Zero-Shot: 76.2
    - GPT-3 One-Shot: 72.5
    - GPT-3 Few-Shot: 86.4

GPT-3: Qualitative Analysis

- Interestingly, GPT-3 can learn and use novel words
  - Test to learn to use previously nonexistent words in few-shot setting
    - Prompts and all definitions were human-generated
    - **Bolds text** are results
    - GPT-3 appears to be at least proficient at the task of using novel words in a sentence.

```
A "whatpu" is a small, furry animal native to Tanzania.  An example of a sentence that uses
the word whatpu is:
We were traveling in Africa and we saw these very cute whatpus.
```

```
To do a "farduddle" means to jump up and down really fast.  An example of a sentence that uses
the word farduddle is:
```

One day when I was playing tag with my little sister, she got really excited and she started doing these crazy farduddles.

```
A "yalubalu" is a type of vegetable that looks like a big pumpkin.  An example of a sentence
that uses the word yalubalu is:
```

I was on a trip to Africa and I tried this yalubalu vegetable that was grown in a garden there. It was delicious.