DNABERT

2021-2 Advanced Bioinformatics

SNU Graduate School of Public Health, Genome & Health Data Lab

Yeojin Jeong

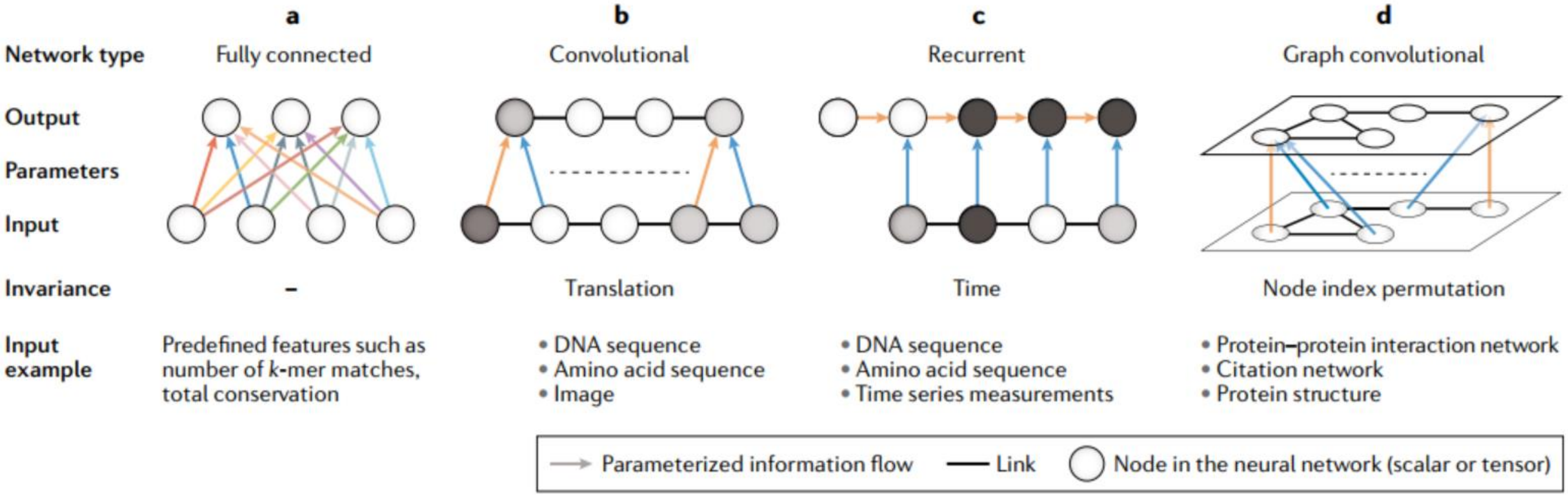
2021 / 12 / 01

Contents

Background

Deep Learning in Genomics

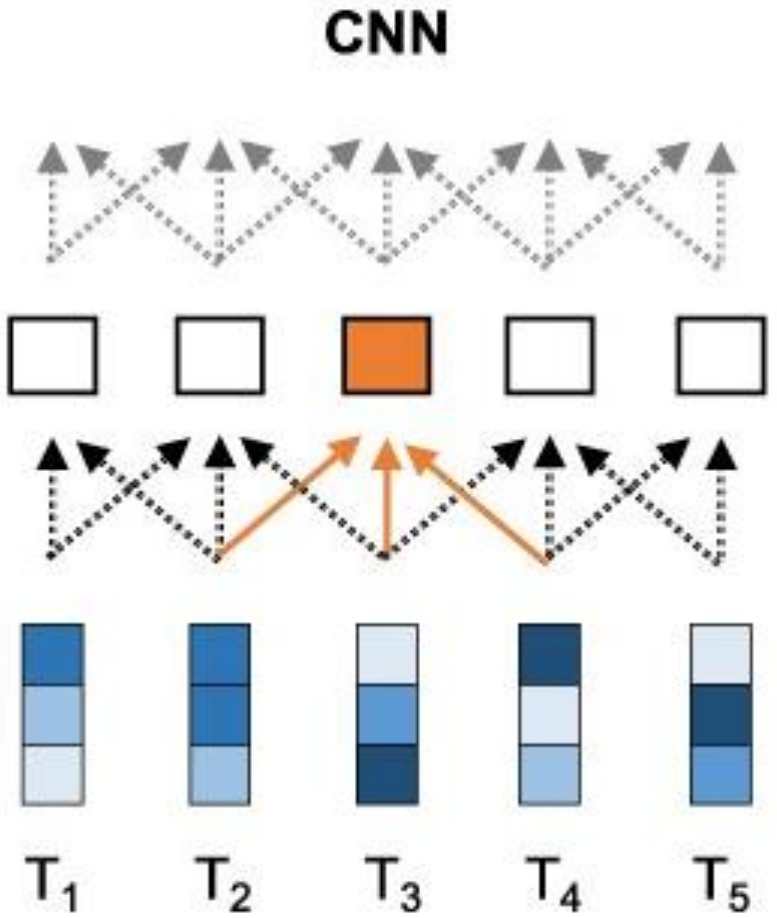
- **Traditional Methods**



Eraslan, G., Avsec, Ž., Gagneur, J. *et al.* Deep learning: new computational modelling techniques for genomics. *Nat Rev Genet* **20,** 389–403 (2019).

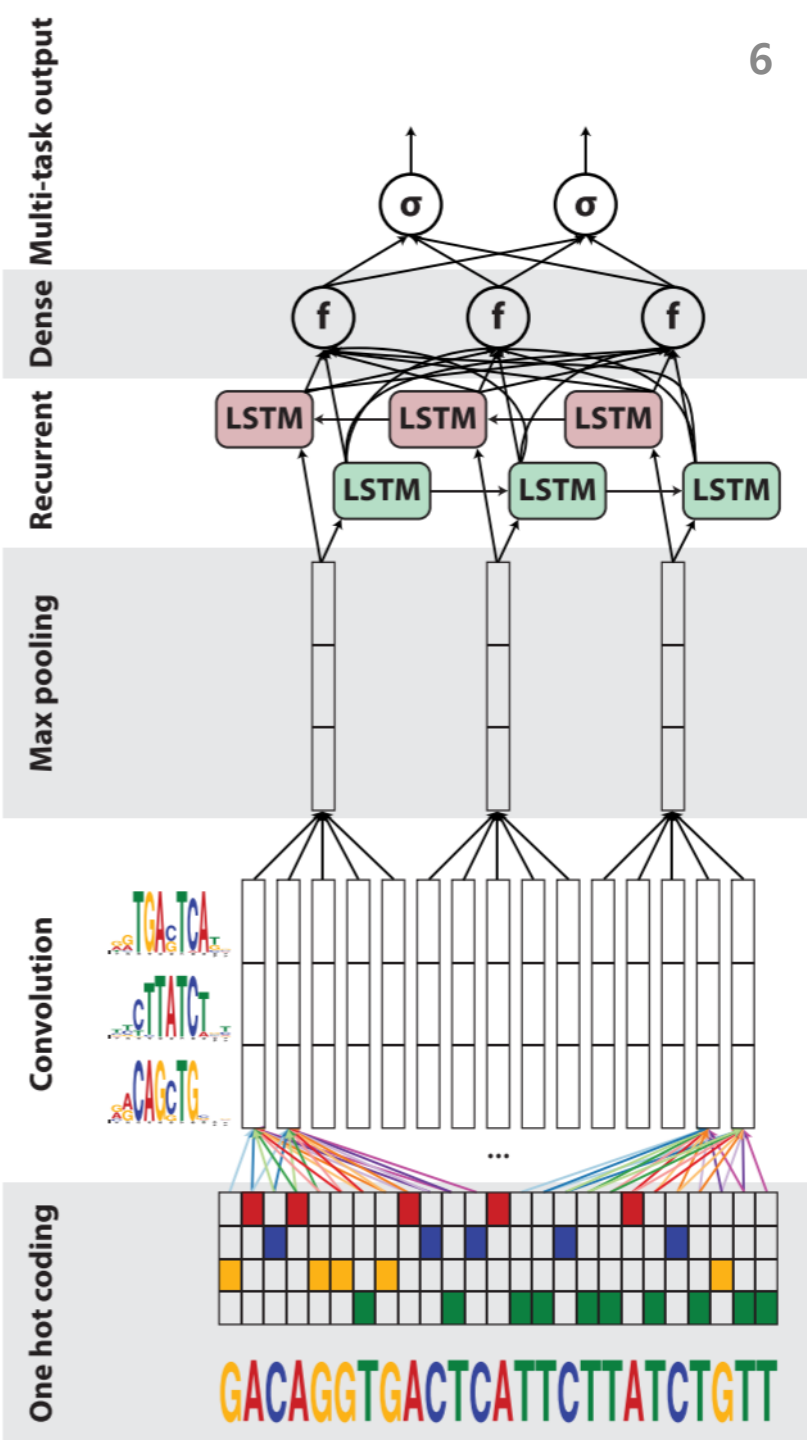
Deep Learning in Genomics

- **Traditional Methods & Limitations**
  - CNN (convolutional Neural Networks)
    - Takes the *local information* in developing each representation
    - Unable to capture the semantic dependency within long-range contexts
    - Its capability to extract local features is limited by the filter size



Deep Learning in Genomics

- **Traditional Methods & Limitations**
  - RNN (Recurrent Neural Networks)
    - Model long-range dependencies in sequences
    - Carry over information through long sequences via memory
    - Propagates information through all hidden states
    - Suffer from vanishing gradient and low-efficiency problem when it sequentially process all past states and compresses contextual information into a bottleneck w/ long input sequences



Quang D, Xie X. DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. Nucleic Acids Res. 2016 Jun 20;44(11):e107.

Representation of Word

- **Sparse Representation**
  - One-hot vector
  - Disadvantage
    - High dimension (curse of dimensionality)
    - Cannot encode the semantic of word

The cat sat on the mat

The: [0 1 0 0 0 0 0]

cat: [0 0 1 0 0 0 0]

sat: [0 0 0 1 0 0 0]

on: [0 0 0 0 1 0 0]

the: [0 0 0 0 0 1 0]

mat: [0 0 0 0 0 0 1]

https://medium.com/analytics-vidhya/one-hot-encoding-of-text-data-in-natural-language-processing-2242fefb2148

Representation of Word

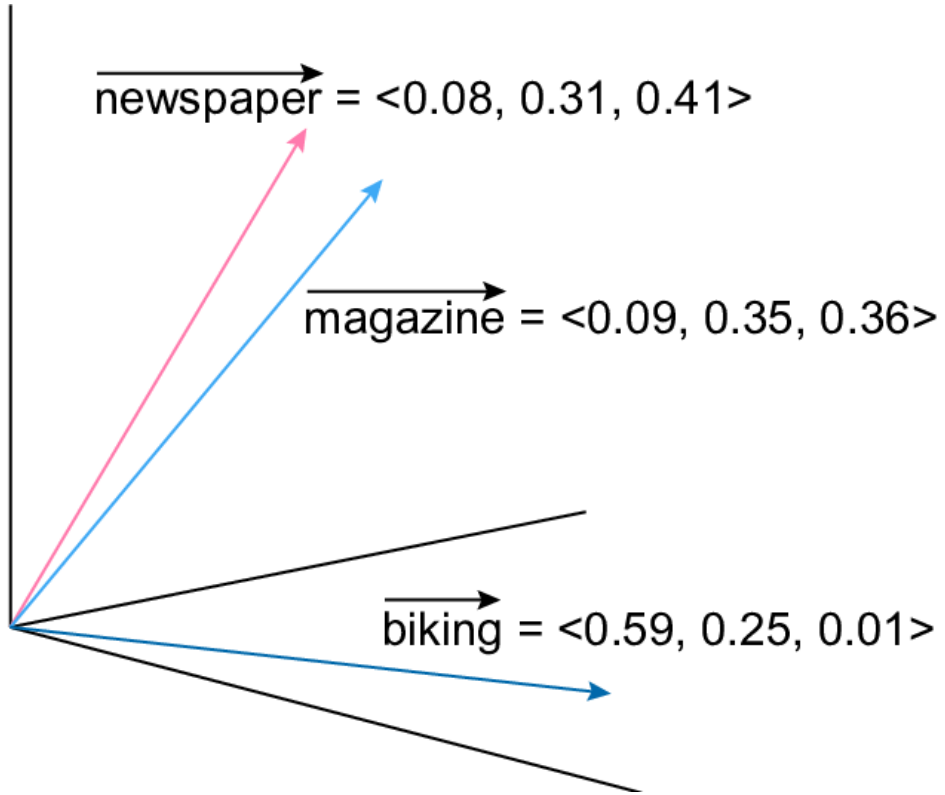
- **Sparse Representation**
  - One-hot vector
    - ex) DNA sequence

	C	G	A	T	A	A	C	C	G	A	T	A	T
A	0	0	1	0	1	1	0	0	0	1	0	1	0
C	1	0	0	0	0	0	1	1	0	0	0	0	0
G	0	1	0	0	0	0	0	0	1	0	0	0	0
T	0	0	0	1	0	0	0	0	0	0	1	0	1

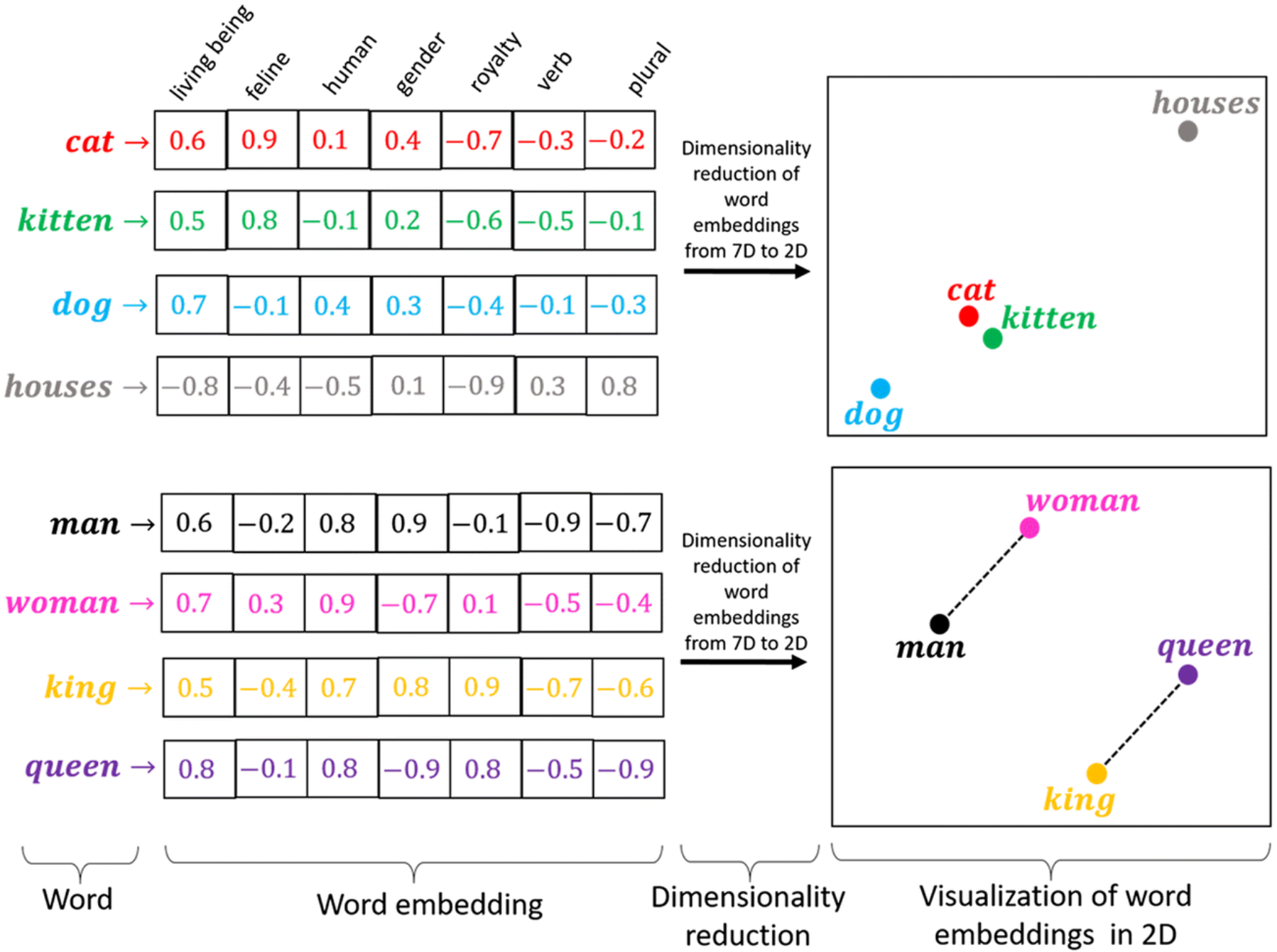
https://elferachid.medium.com/one-hot-encoding-dna-92a1c29ba15a

Representation of Word

- **Dense Representation**
  - **Word embedding (representation, vector)**
    - Dense vector for each word
    - Distributed representation of word
    - Represent a point in some sort of “word” space
    - Sufficient to encode all semantics of the word
    - Each dimension would encode some meaning
  - GloVe, FastText, Word2Vec …



https://www.researchgate.net/publication/325451970_Text_Understanding_for_Computers



https://medium.com/@hari4om/word-embedding-d816f643140

Representation of Word

- **Dense Representation**
  - Limitation of **static (context-free)** word embedding
    - We use pre-defined word embedding

      → Cannot distinguish homonym / polysemy
    - ex) “He got bit by Python.”

      “Python is my favorite programming language.”

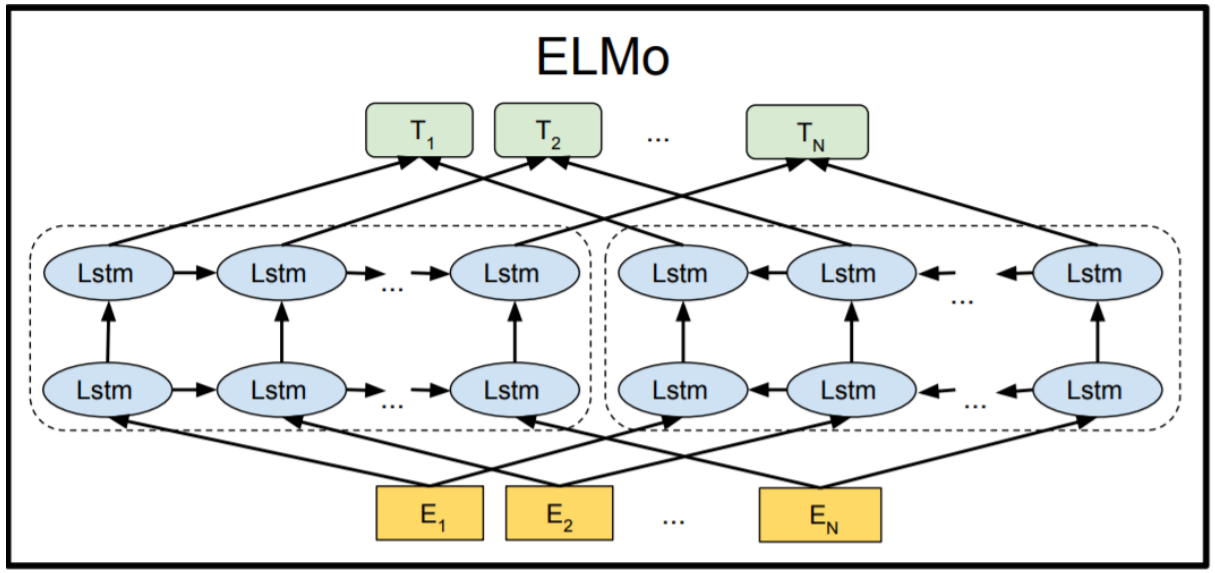
      → Word embedding of “Python” would be same

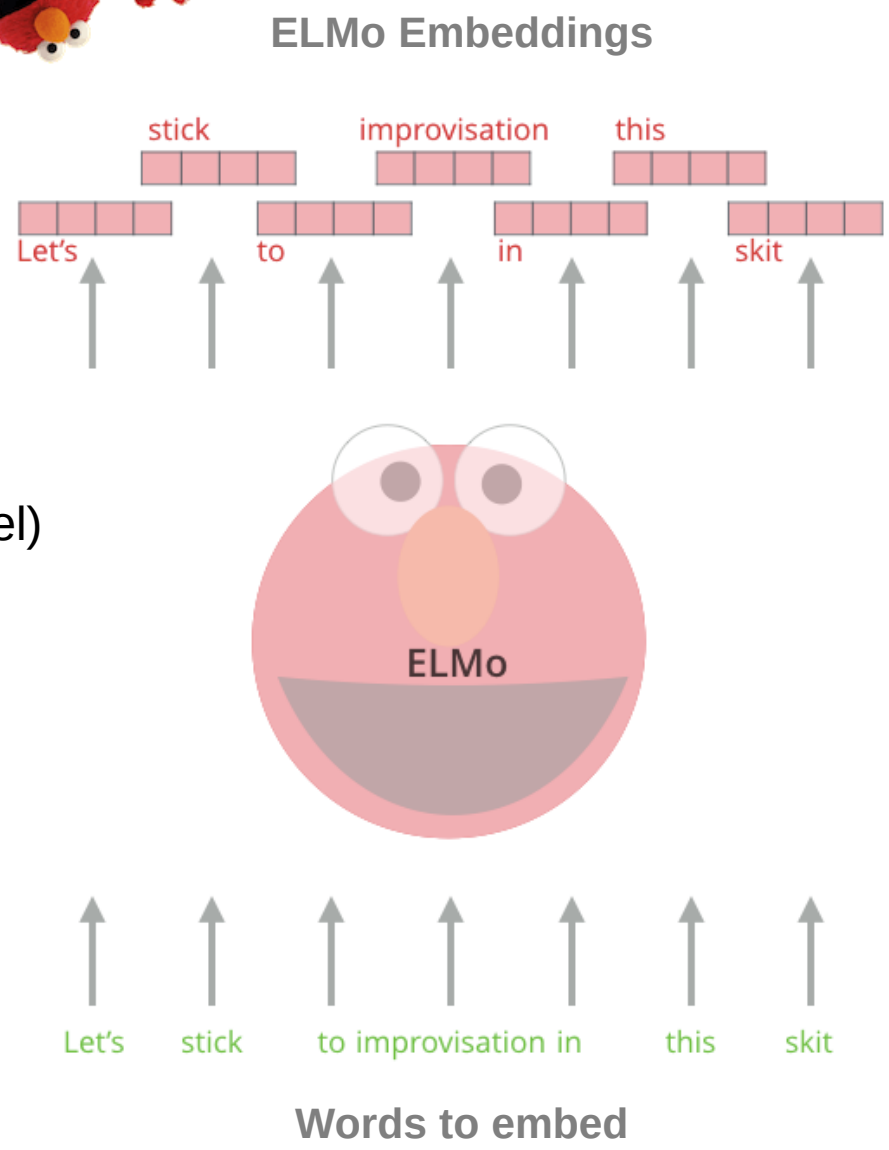
      in both sentences
  - It ignores the context and always give the same embedding irrespective of context

Representation of Word

- **Dense Representation**
  - **Dynamic (Context-based)** Word Embedding
    - ELMo (embeddings from language model)
      - From bidirectional language model

        (forward language model + backward language model)





https://medium.com/saarthi-ai/elmo-for-contextual-word-embedding-for-text-classification-24c9693b0045

Attention, Transformer, and BERT

No More RNN, Only Attention

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

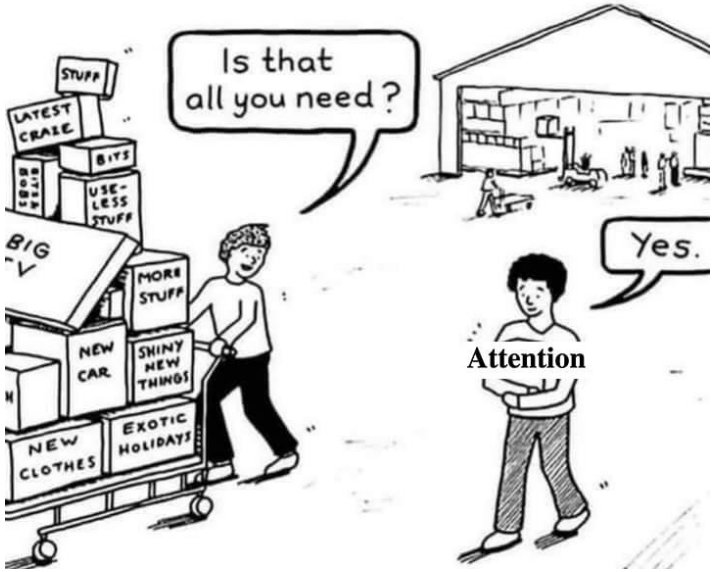
Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

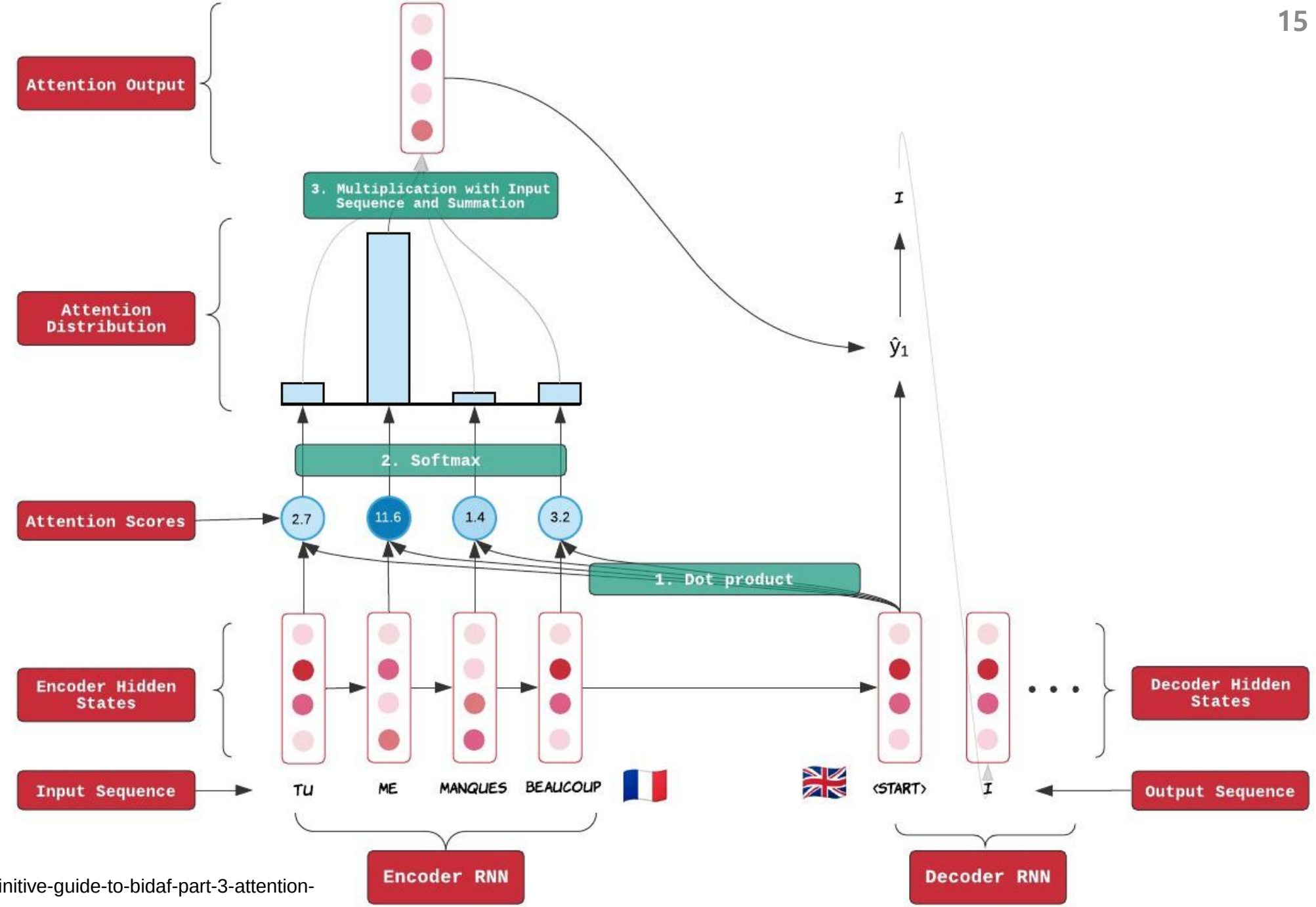
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com



https://knowyourmeme.com/memes/attention-is-all-you-need

Attention

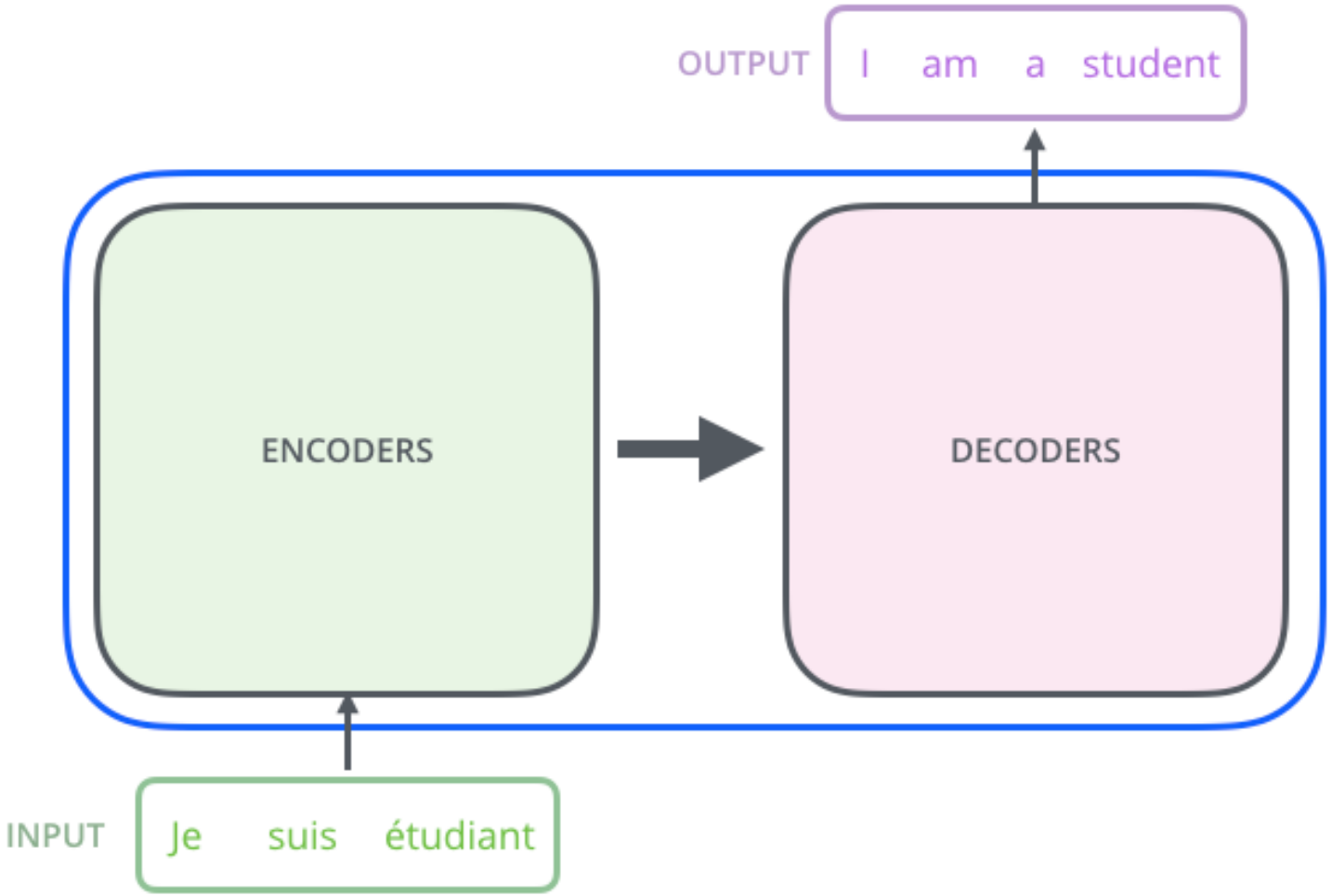


https://towardsdatascience.com/the-definitive-guide-to-bidaf-part-3-attention-92352bbdcb07

Transformer

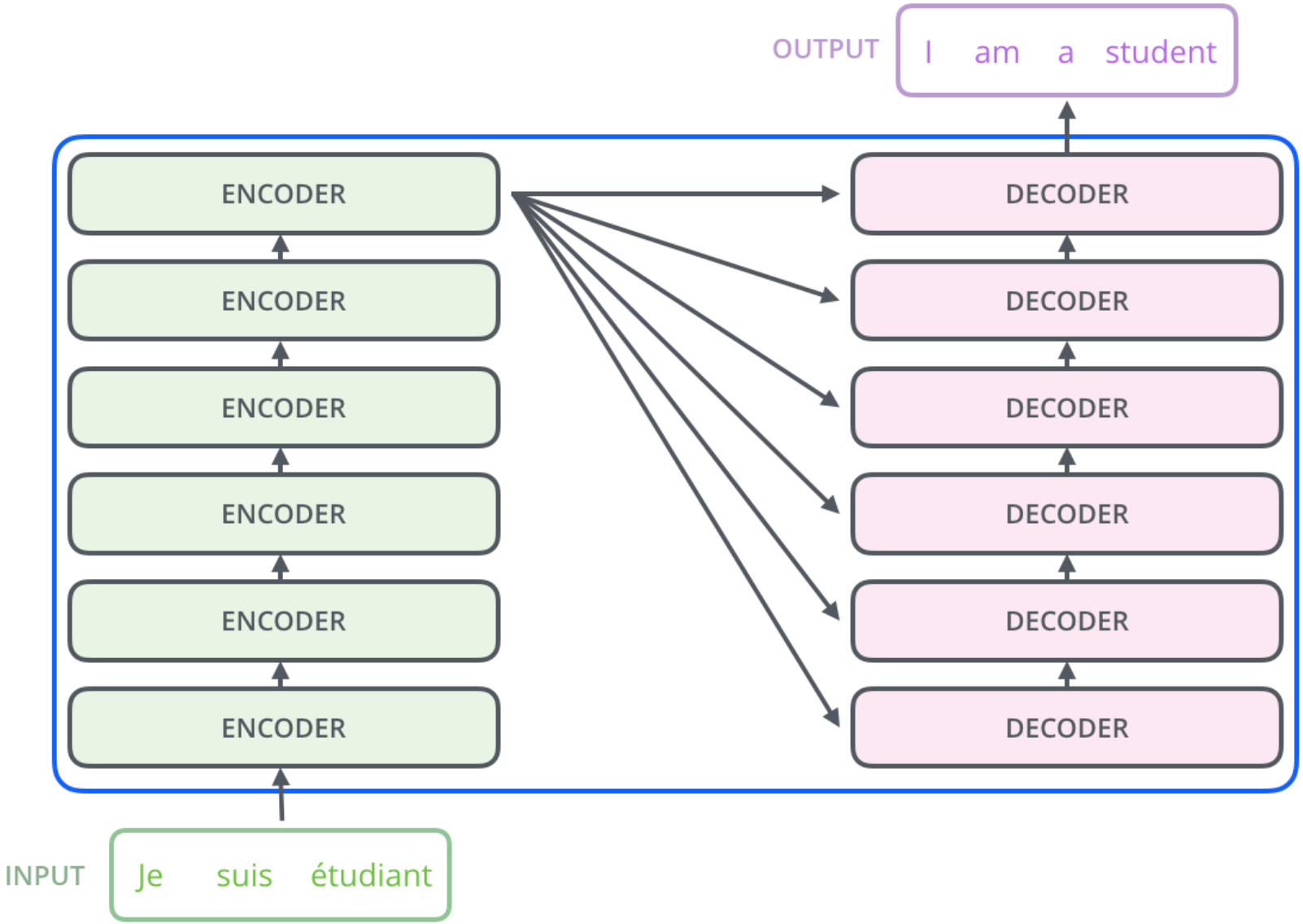
https://jalammar.github.io/illustrated-transformer/

Transformer



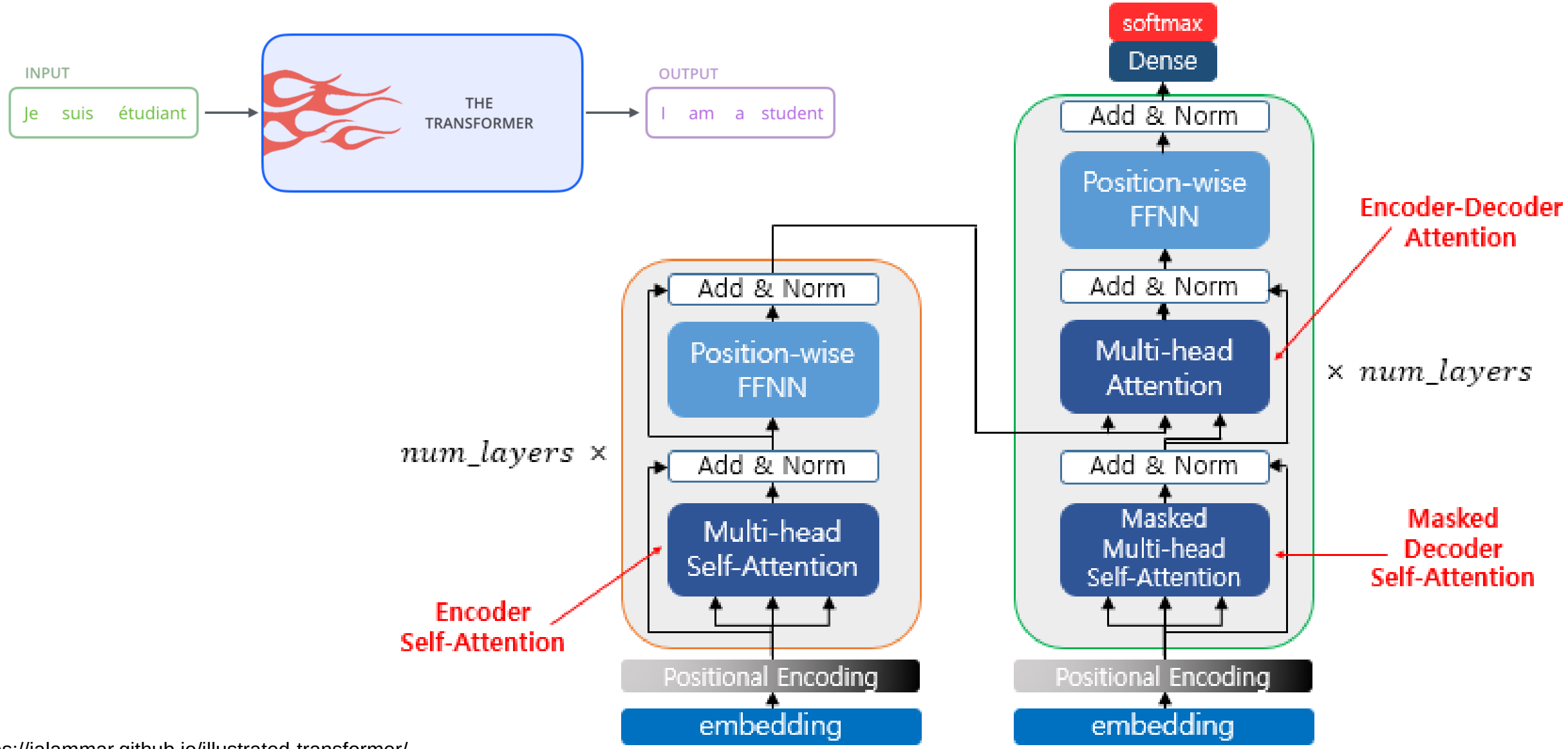
https://jalammar.github.io/illustrated-transformer/

Transformer



https://jalammar.github.io/illustrated-transformer/

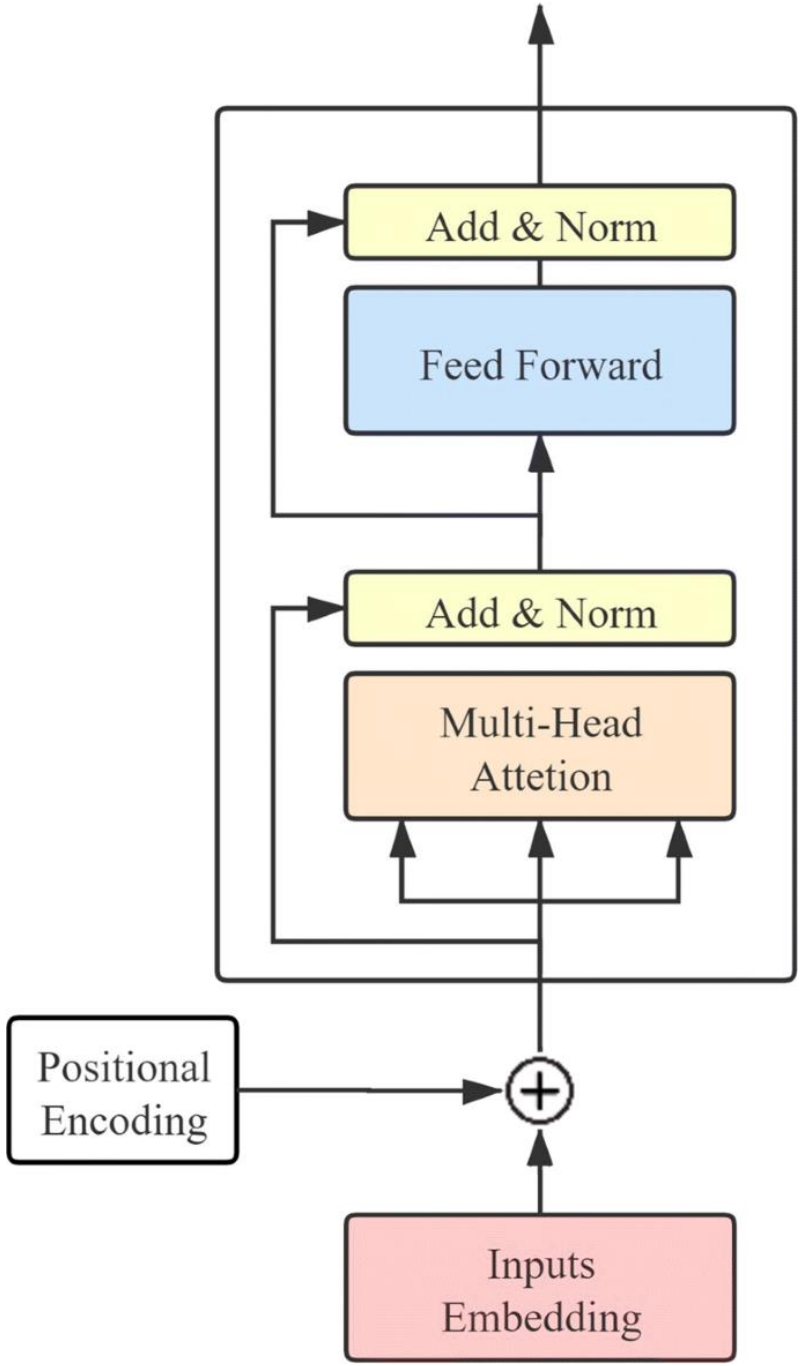
Transformer



https://jalammar.github.io/illustrated-transformer/

Encoder of Transformer

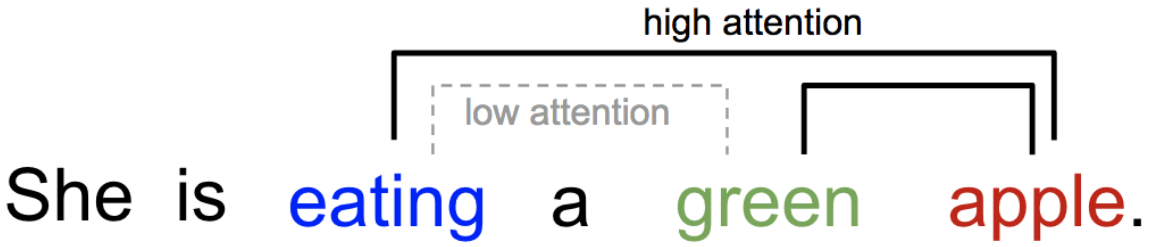
- 1) Multi-head attention
- 2) Feedforward network

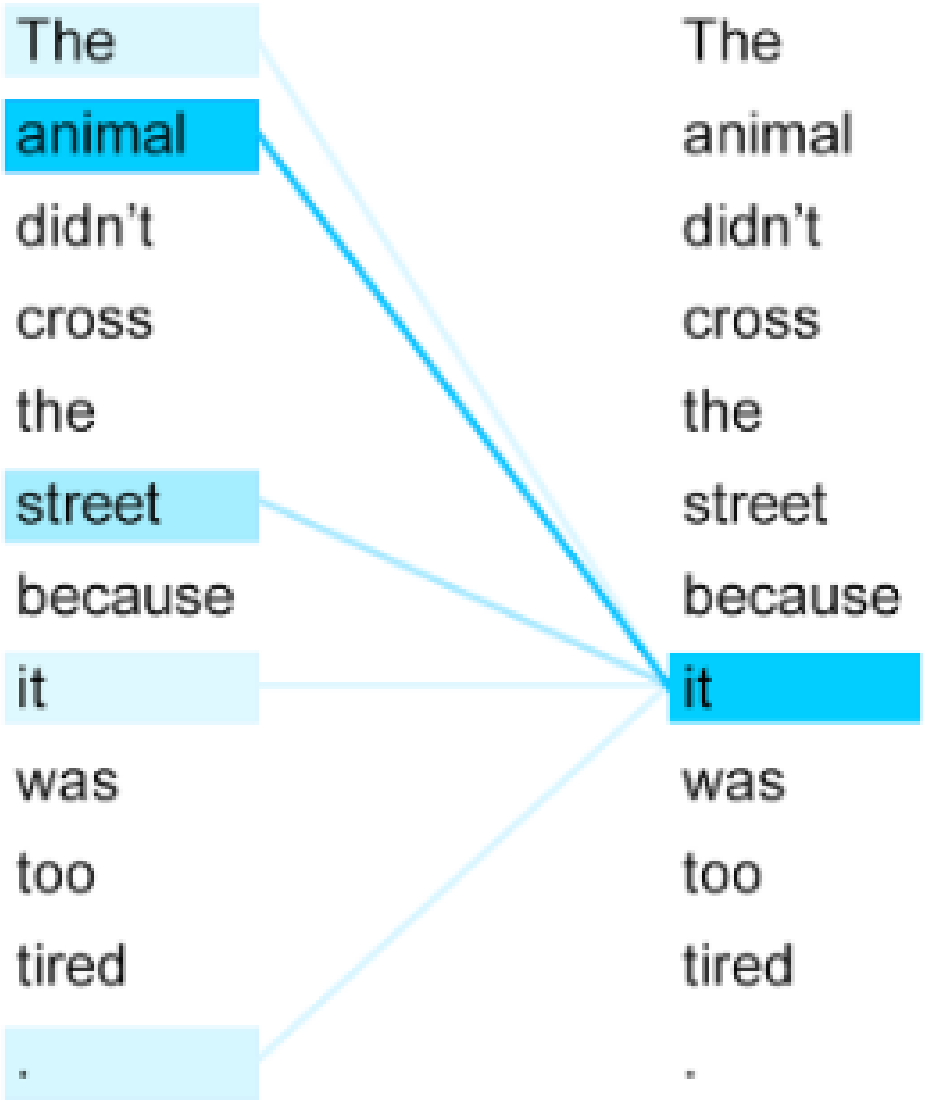


https://link.springer.com/article/10.1007/s10489-021-02863-9

Self-Attention

- Example



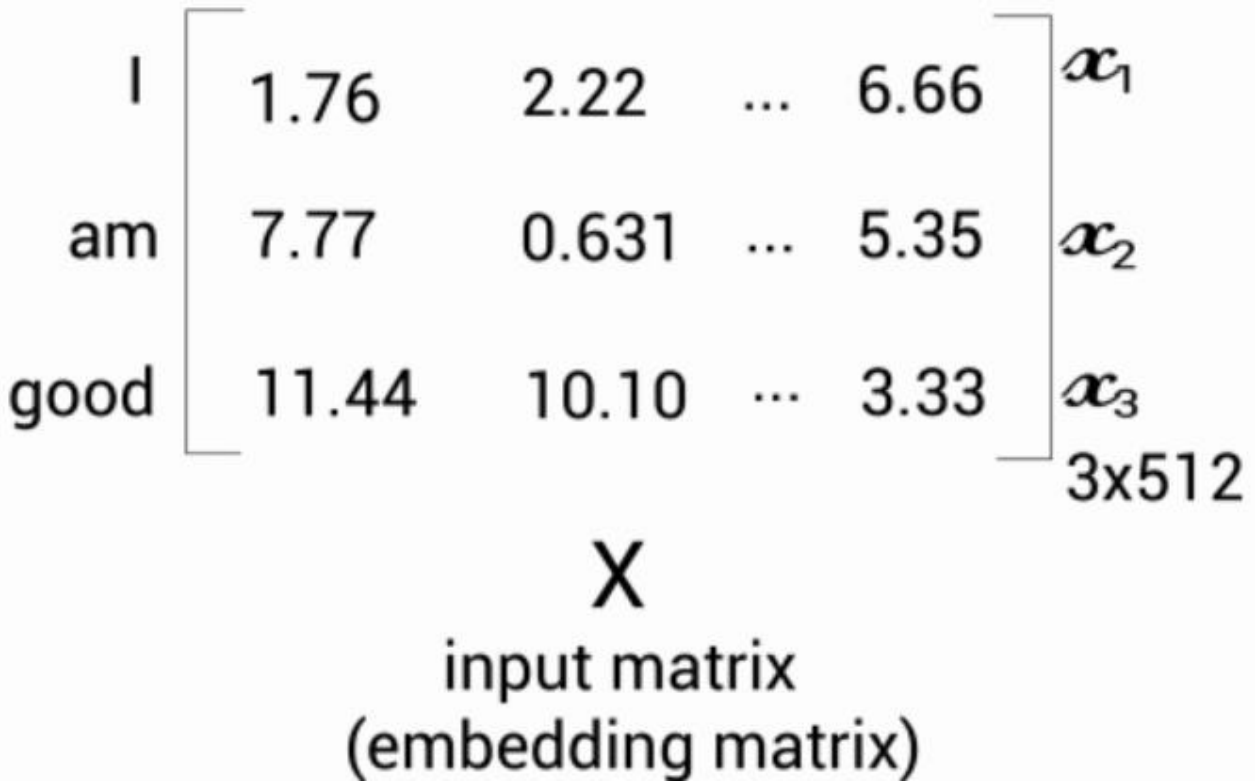


https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html

https://www.researchgate.net/figure/An-example-of-the-self-attention-mechanism-following-long-distance-dependency-in-the_fig1_342301870

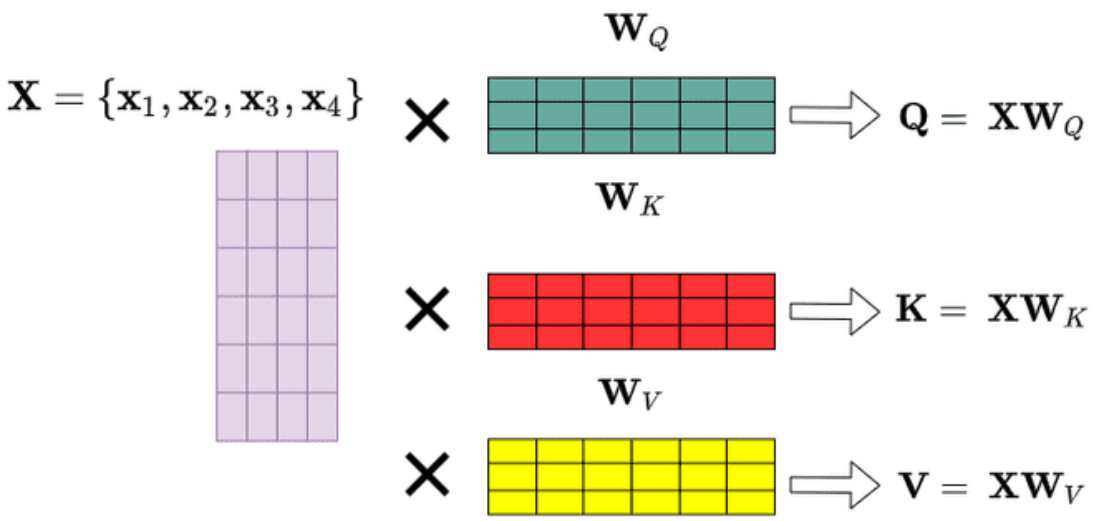
Self-Attention

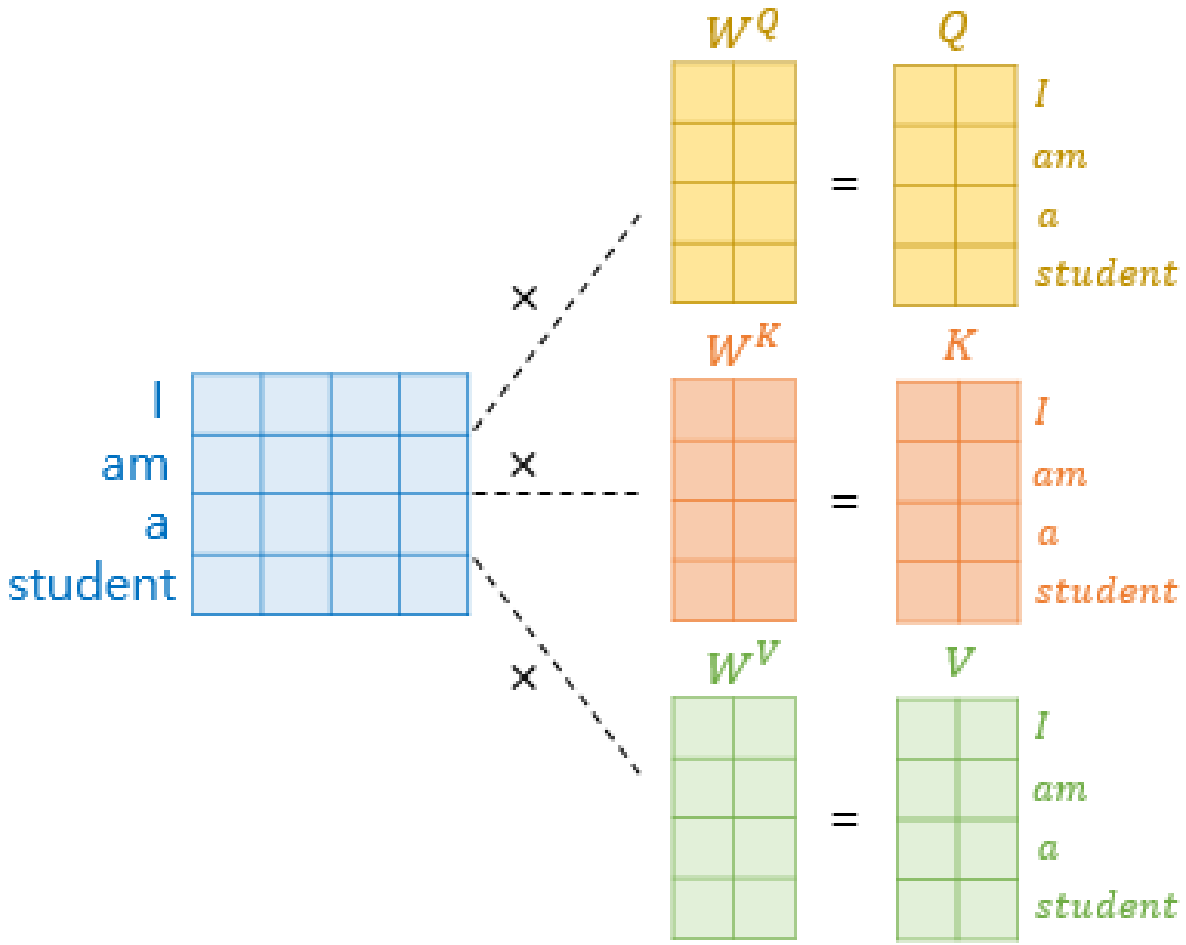
- Input Matrix (embedding matrix / input embedding)
  - ex) “I am good”
    - 1st row : embedding of “I”
    - 2nd row : embedding of “am”
    - 3rd row : embedding of “good”



Self-Attention

- Query (Q), Key (K), Value (V) matrix
  - W^Q, W^K, W^V → Multiplied to input matrix X
    - 1st row : Query, Key, Value of first word
    - 2nd row : Query, Key, Value of second word
  - $Q = XW^Q$, $K = XW^K$, $V = XW^V$

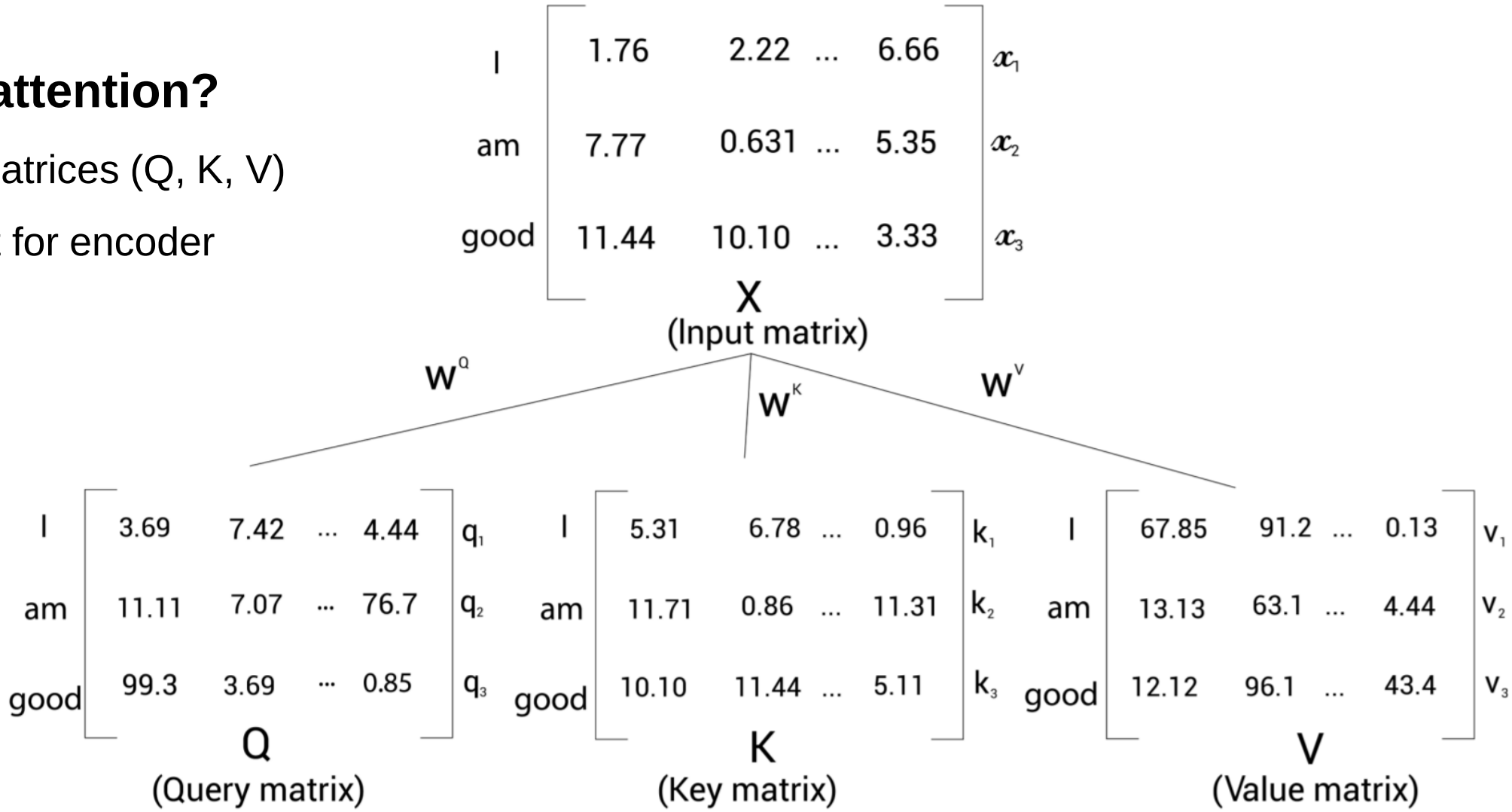




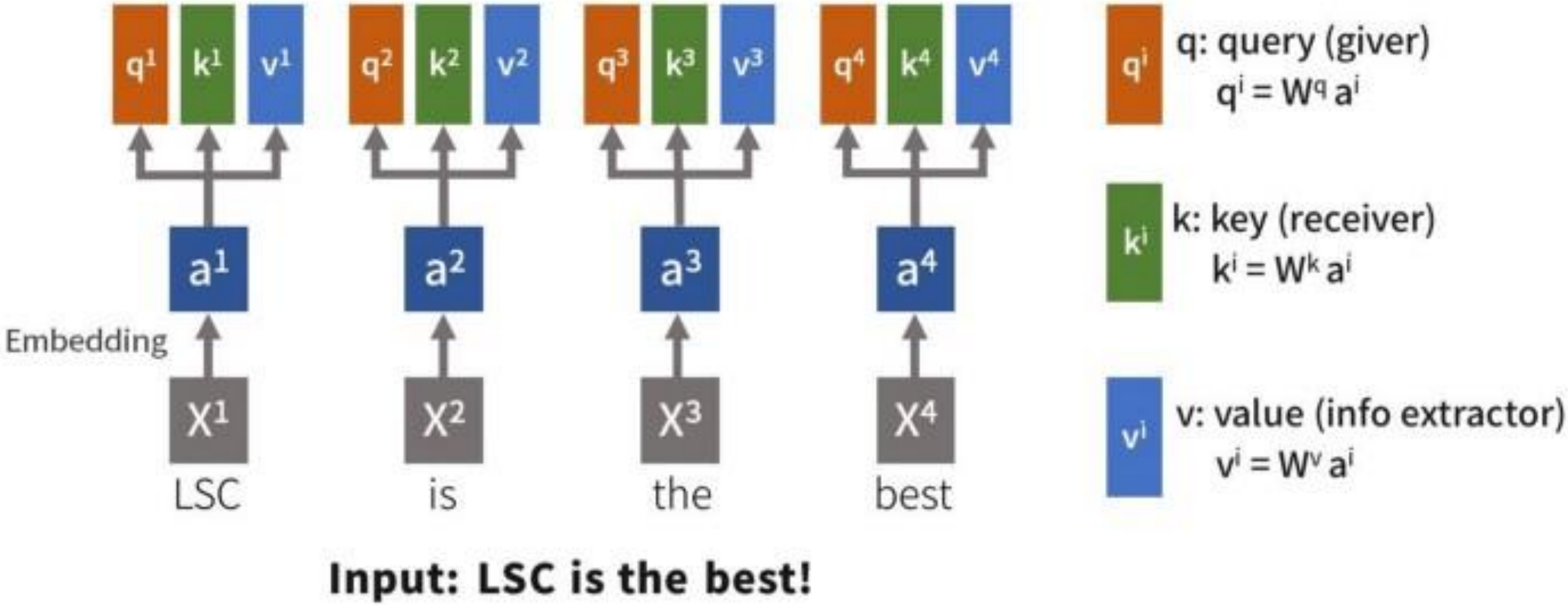
https://wikidocs.net/22893
https://theaisummer.com/transformer/

Self-Attention

- **Why self-attention?**
  - All the matrices (Q, K, V) are input for encoder



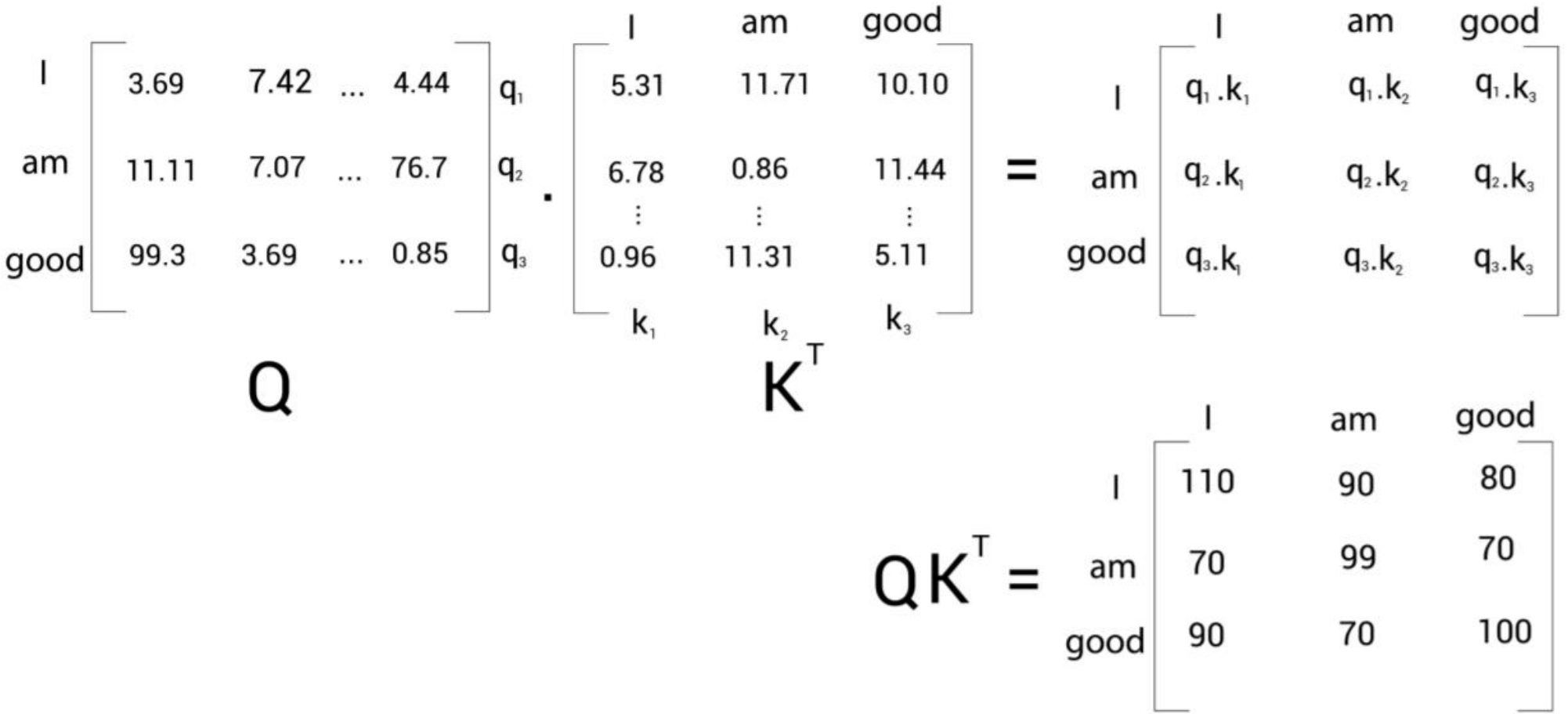
Self-Attention



https://medium.com/lsc-psd/introduction-of-self-attention-layer-in-transformer-fc7bff63f3bc

Self-Attention

- Dot product Query & Key
  - QK^T



Self-Attention

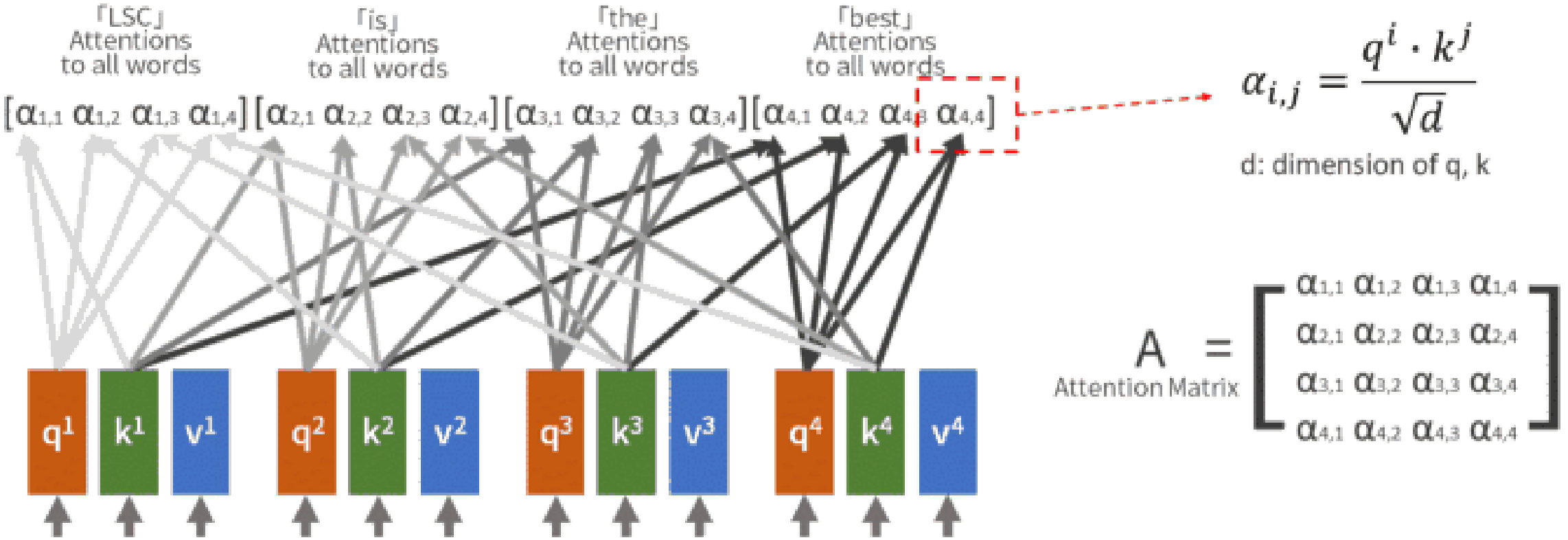
- Dot product Query & Key
  - QK^T
    - 1sr row : Similarity between q_1(I) & k_1(I), k_2(am), k_3(good)

$$QK^T = \begin{array}{c} \\ \text{I} \\ \text{am} \\ \text{good} \end{array} \begin{array}{c} \begin{array}{ccc} \text{I} & \text{am} & \text{good} \end{array} \\ \begin{bmatrix} 110 & 90 & 80 \\ 70 & 99 & 70 \\ 90 & 70 & 100 \end{bmatrix} \end{array}$$

$q_1.k_1$ $q_1.k_2$ $q_1.k_3$

Self-Attention

- Dot product Query & Key
  - QK^T



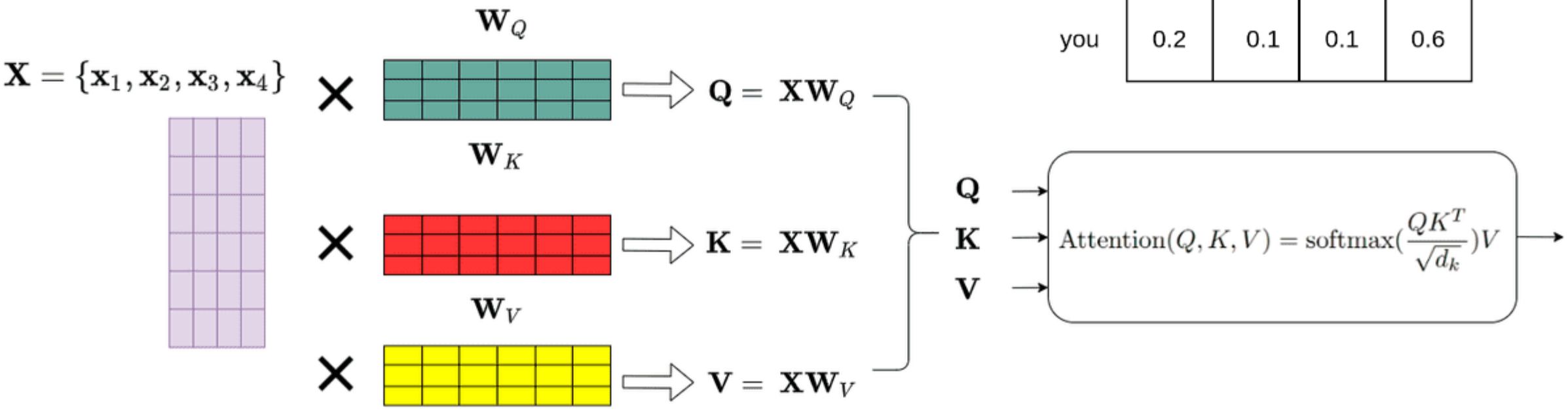
https://medium.com/lsc-psd/introduction-of-self-attention-layer-in-transformer-fc7bff63f3bc

Self-Attention

- Score matrix = $softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)$

Self-attention
Probability score matrix

	Hello	I	love	you
Hello	0.8	0.1	0.05	0.05
I	0.1	0.6	0.2	0.1
love	0.05	0.2	0.65	0.1
you	0.2	0.1	0.1	0.6



https://theaisummer.com/transformer/

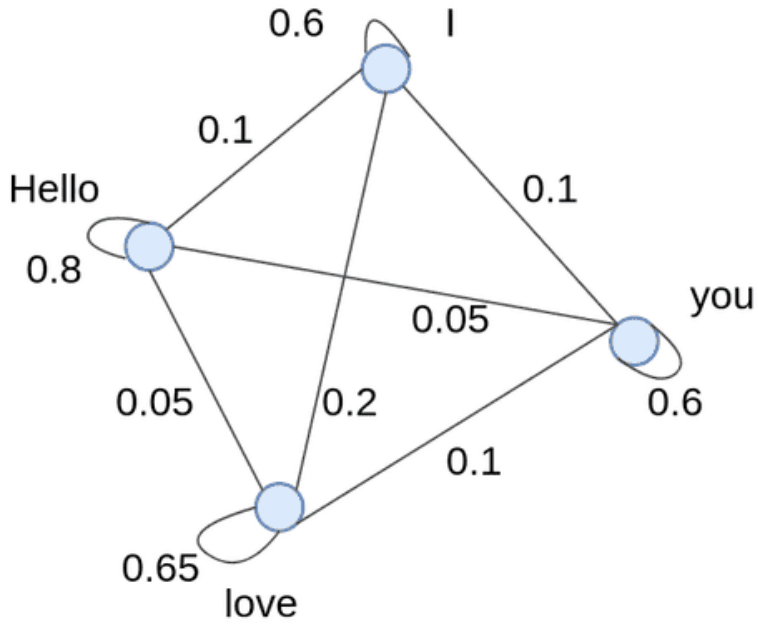
Self-Attention

- Score matrix = $softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)$
  - **Self-attention as an undirected weighted graph**

Self-attention
Probability score matrix

	Hello	I	love	you
Hello	0.8	0.1	0.05	0.05
I	0.1	0.6	0.2	0.1
love	0.05	0.2	0.65	0.1
you	0.2	0.1	0.1	0.6

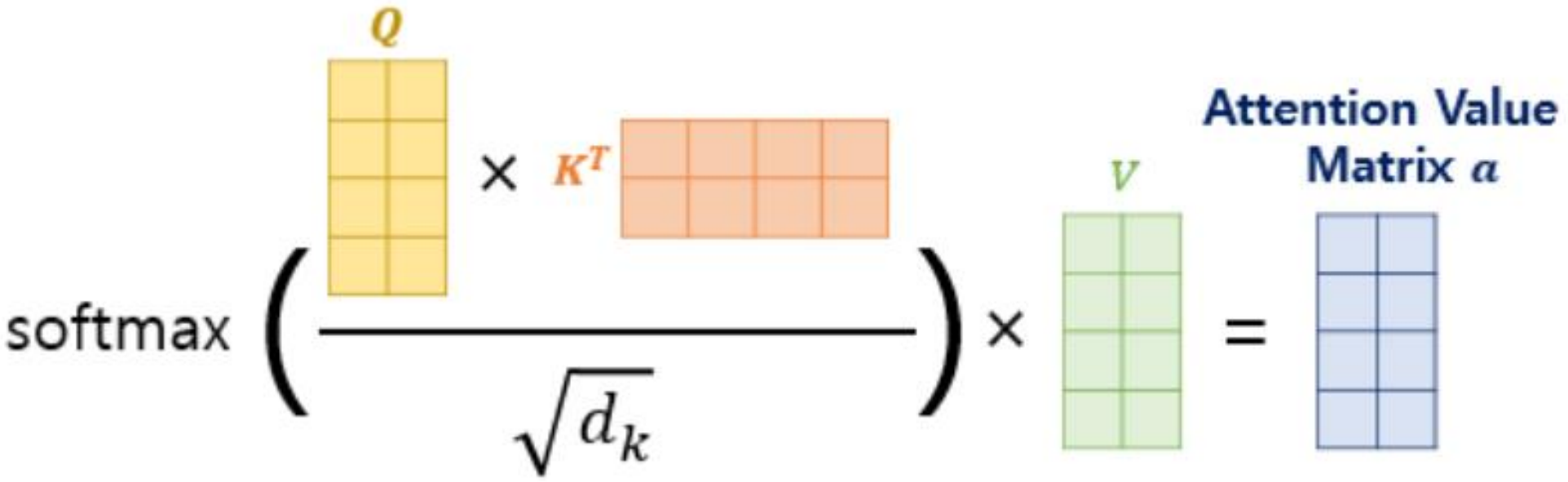
Self-attention as a
undirected weighted graph



https://theaisummer.com/attention/

Self-Attention

- **Attention Matrix**
  - = Score Matrix x **V (value)**

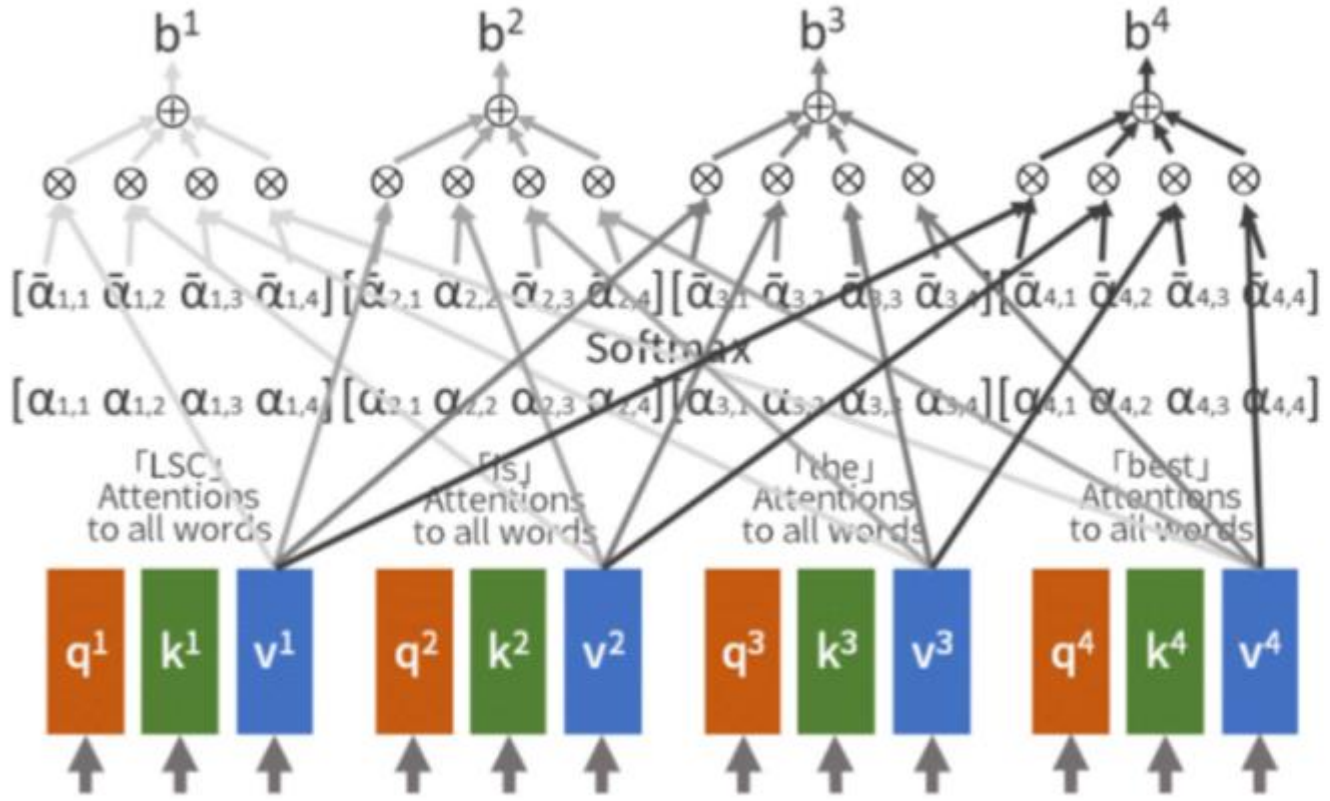


- $Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$

https://wikidocs.net/22893

Self-Attention

- **Attention Matrix *Z***
  - = Score Matrix x **V (value)**

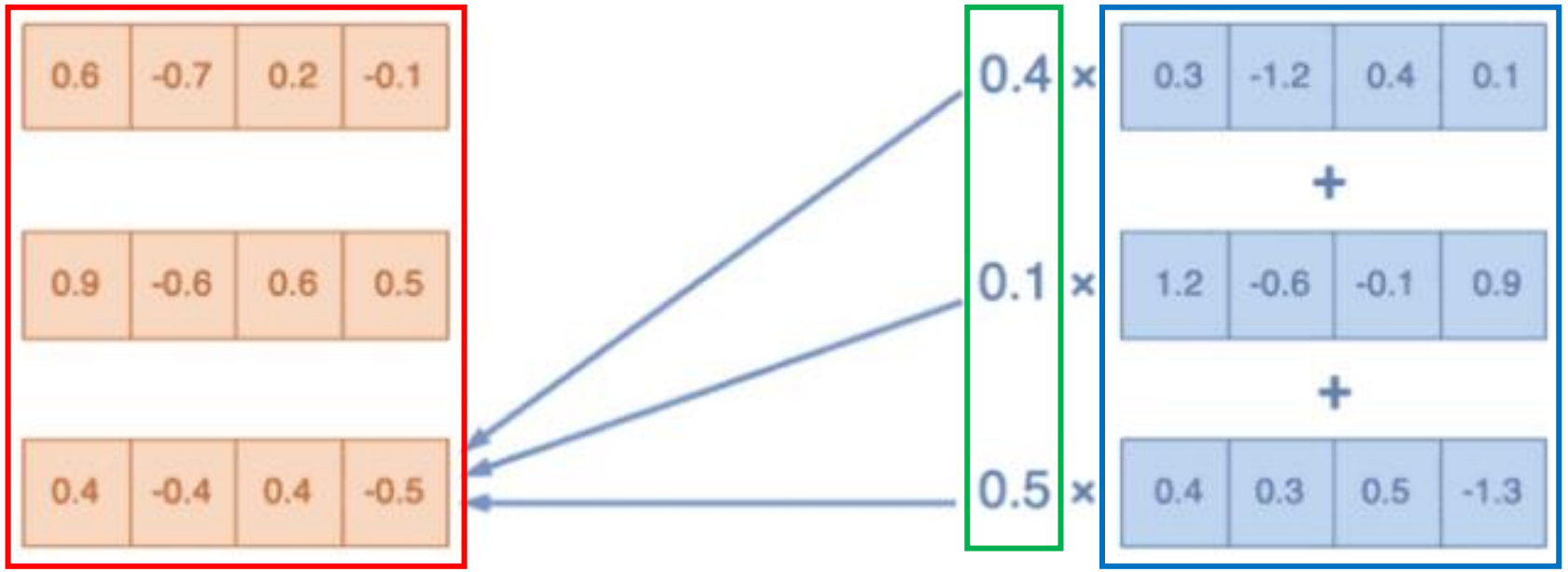


$$b^i = \sum_j \bar{\alpha}_{i,j} v^j$$

https://medium.com/lsc-psd/introduction-of-self-attention-layer-in-transformer-fc7bff63f3bc

Self-Attention

- Attention is just a fancy name for weighted average
  - **Each row of attention matrix (Z) = Weighted average of rows of Value matrix (V)**



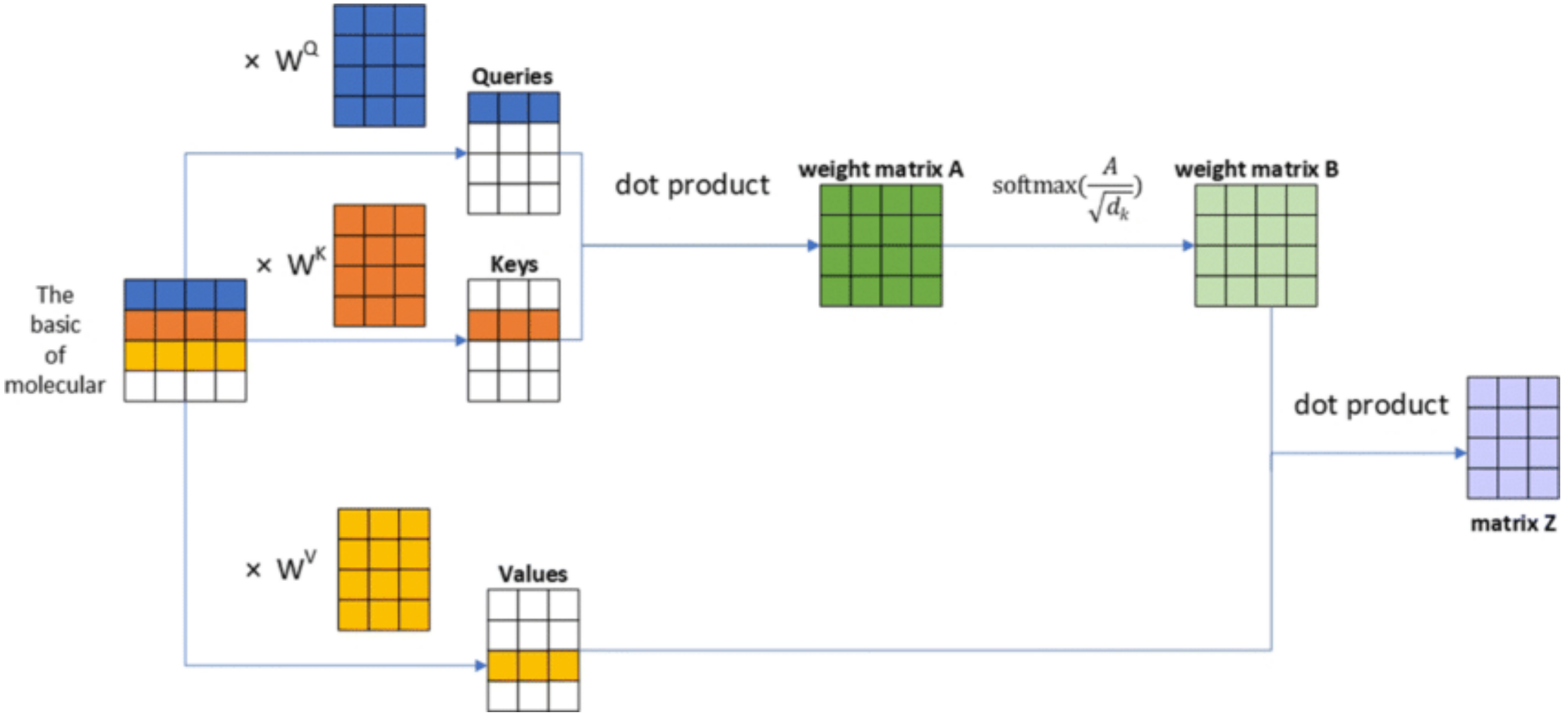
Attention Matrix (Z)

from Score Matrix $(softmax\left(\frac{QK^T}{\sqrt{d_k}}\right))$

Value Matrix (V)

https://towardsdatascience.com/deconstructing-bert-part-2-visualizing-the-inner-workings-of-attention-60a16d86b5c1

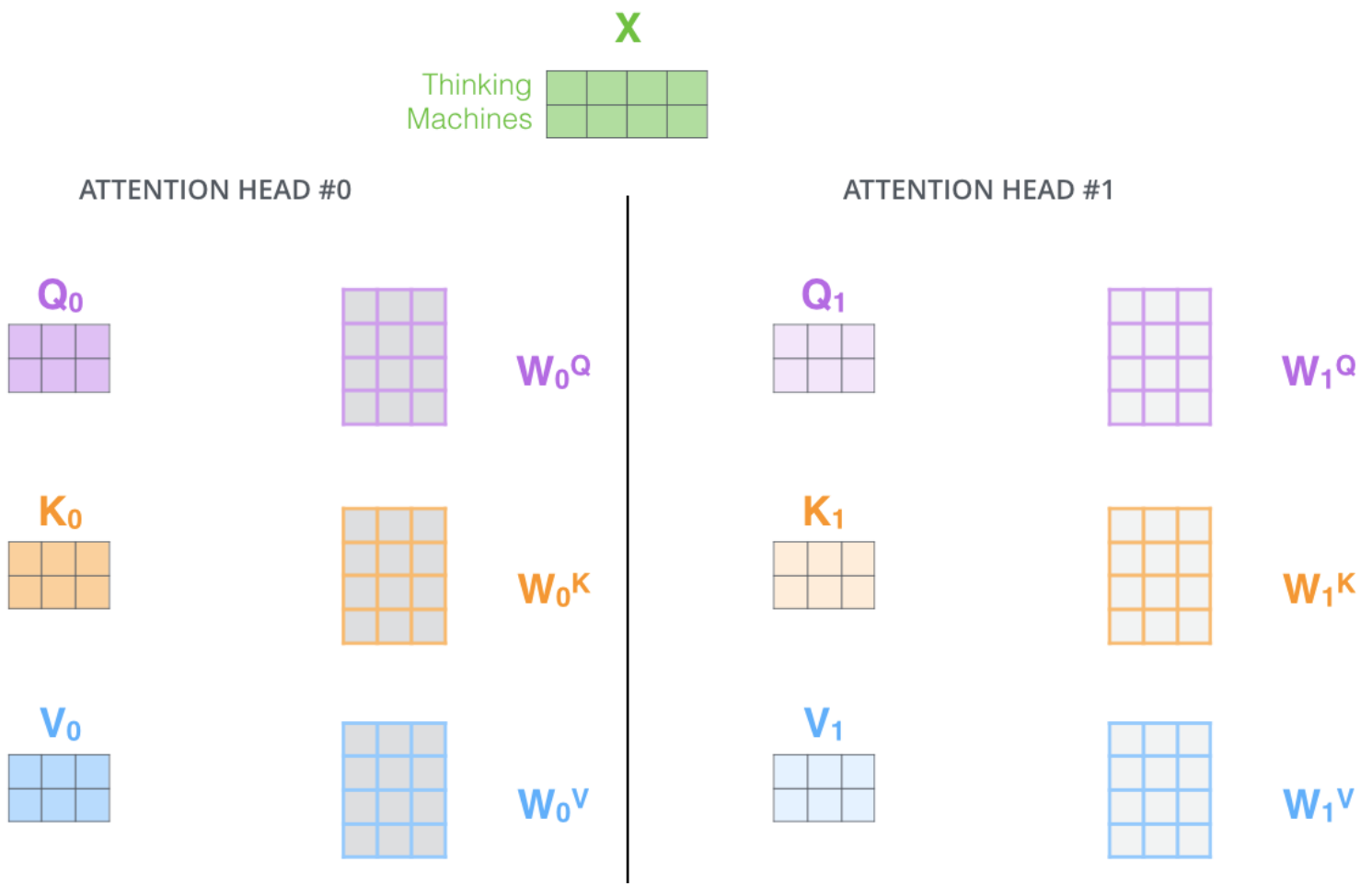
Self-Attention



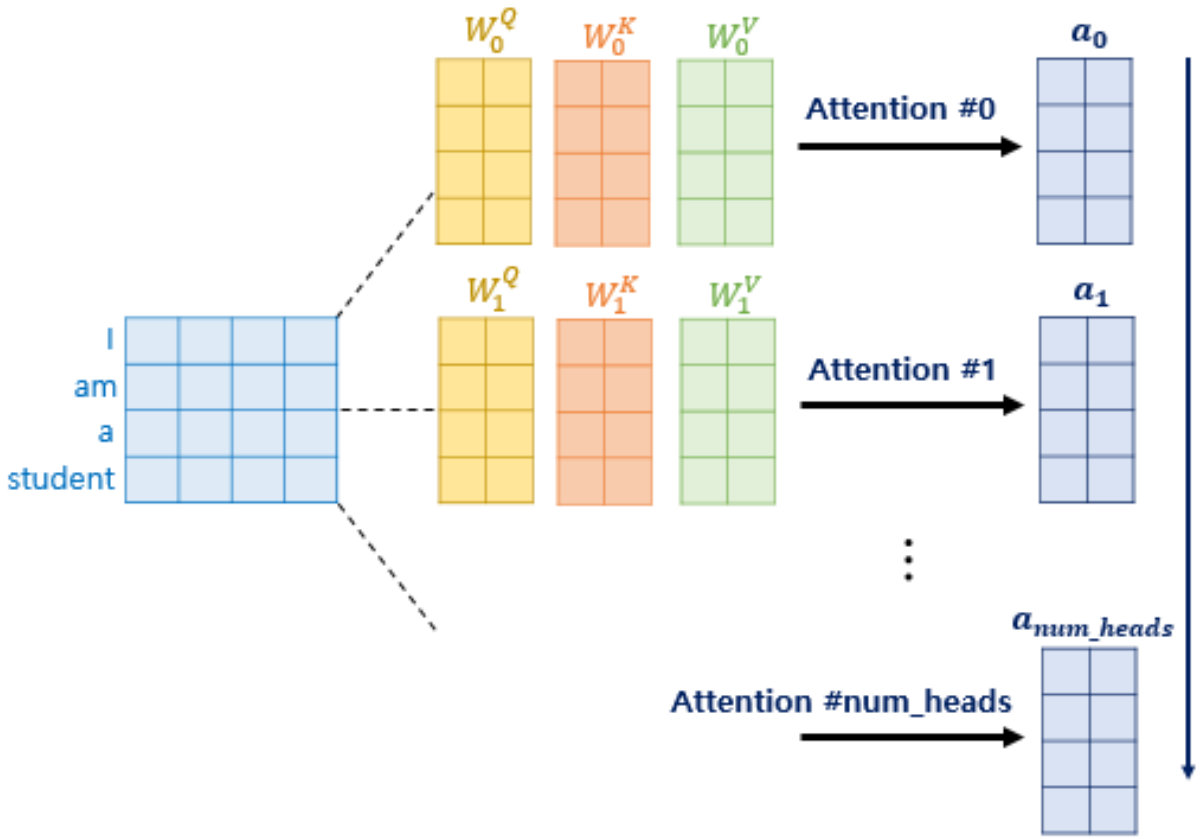
https://www.researchgate.net/figure/The-self-attention-mechanism-calculation-process-We-can-get-three-vectors-that-are-a_fig4_344301413

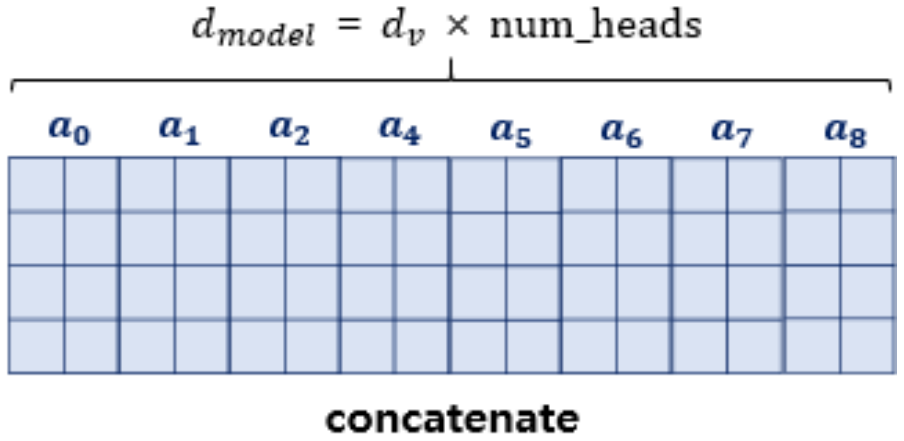
Multi-head Self-Attention

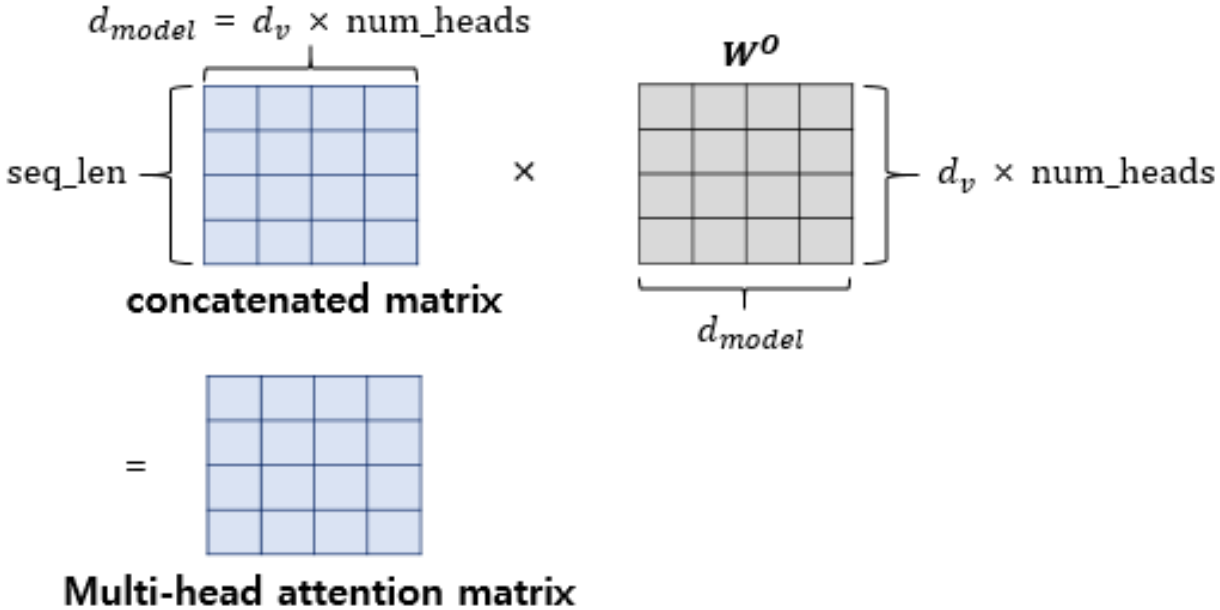
- Use multiple attention heads
  - Improve accuracy



https://jalammar.github.io/illustrated-transformer/

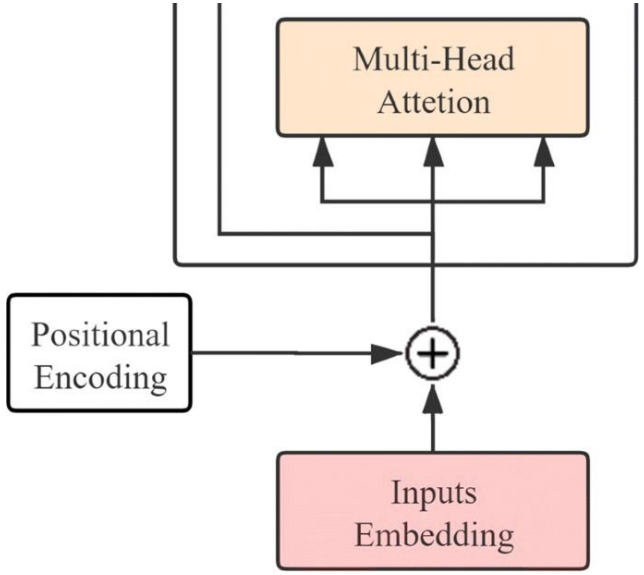

I
am
a
student
W_0^Q
W_0^K
W_0^V
W_1^Q
W_1^K
W_1^V
Attention #0
Attention #1
Attention #num_heads
a_0
a_1
a_{num_heads}

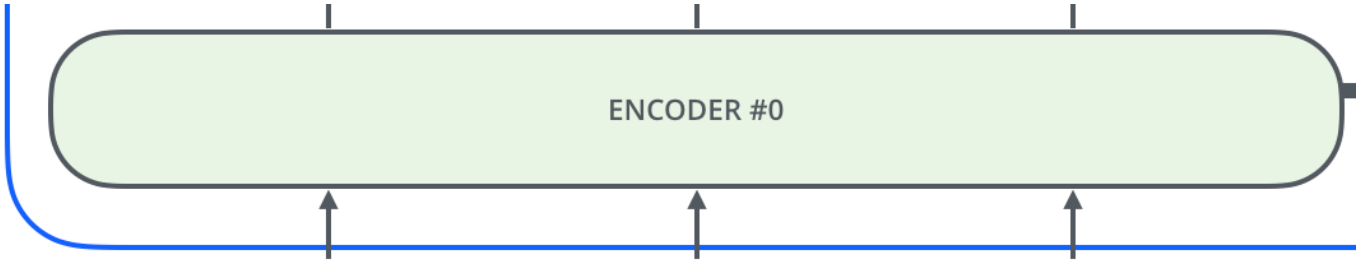

$d_{model} = d_v \times$ num_heads
a_0 a_1 a_2 a_4 a_5 a_6 a_7 a_8
concatenate

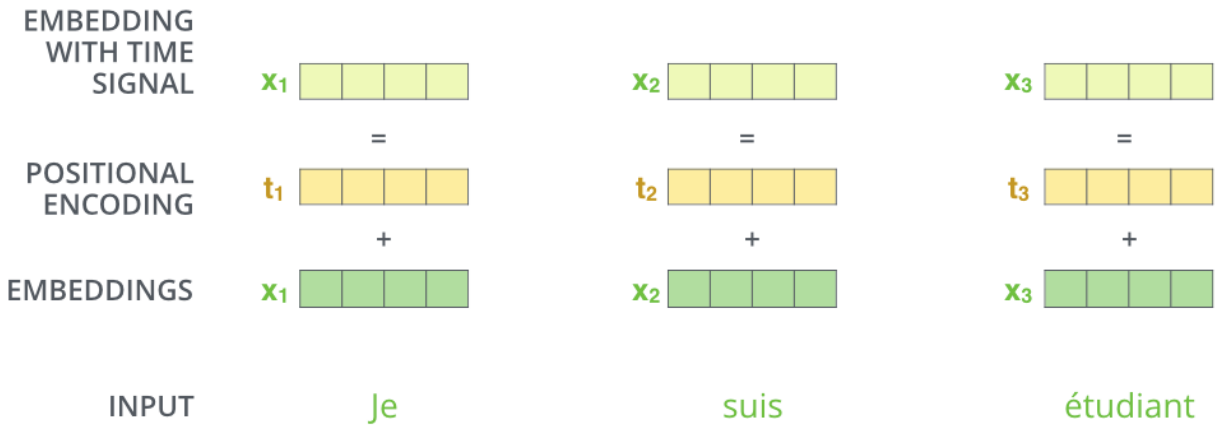

$d_{model} = d_v \times$ num_heads
seq_len
concatenated matrix
×
W^O
$d_v \times$ num_heads
d_{model}
=
Multi-head attention matrix

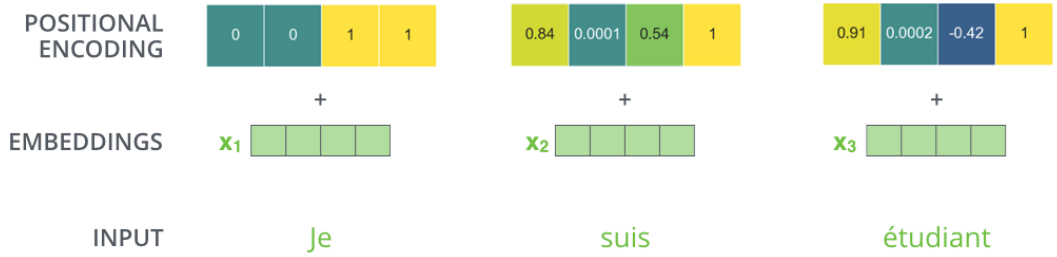
Positional Encoding

- Positional encoding
  - Way to account for the order of the words in the input sequence
  - Determine the position of each word





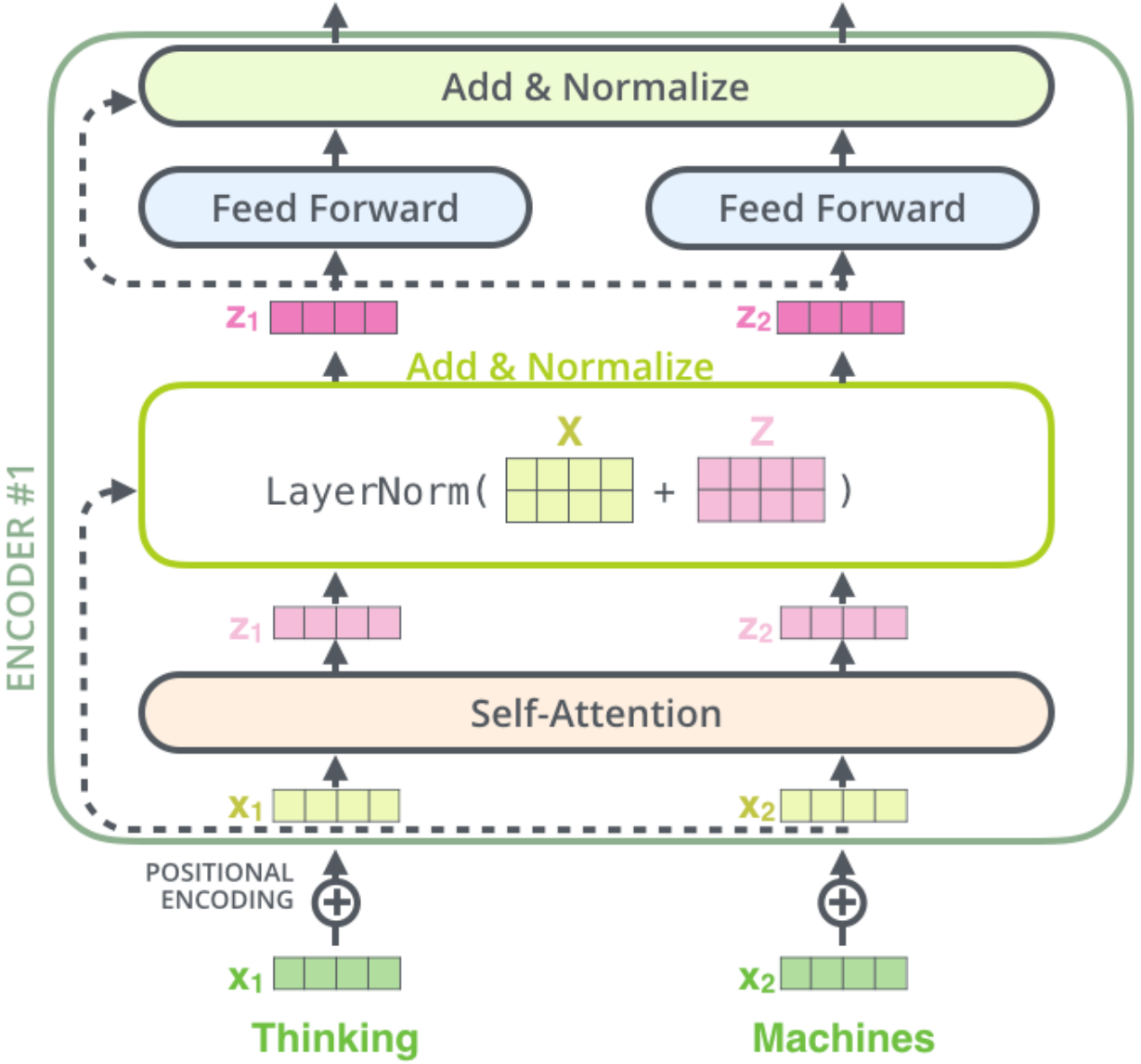




https://jalammar.github.io/illustrated-transformer/

Feedforward Network

- Typical fully connected layer
  - 2 Dense layer
    - Parameters are the same over the different positions of the sentence and different over the encoder blocks
  - ReLU (activation function)



https://jalammar.github.io/illustrated-transformer/

BERT

- **Bidirectional Encoder Representation from Transformer**
  - *BERT is a method of **pre-training language representations,***

    *meaning that we train a **general-purpose "language understanding"** model*

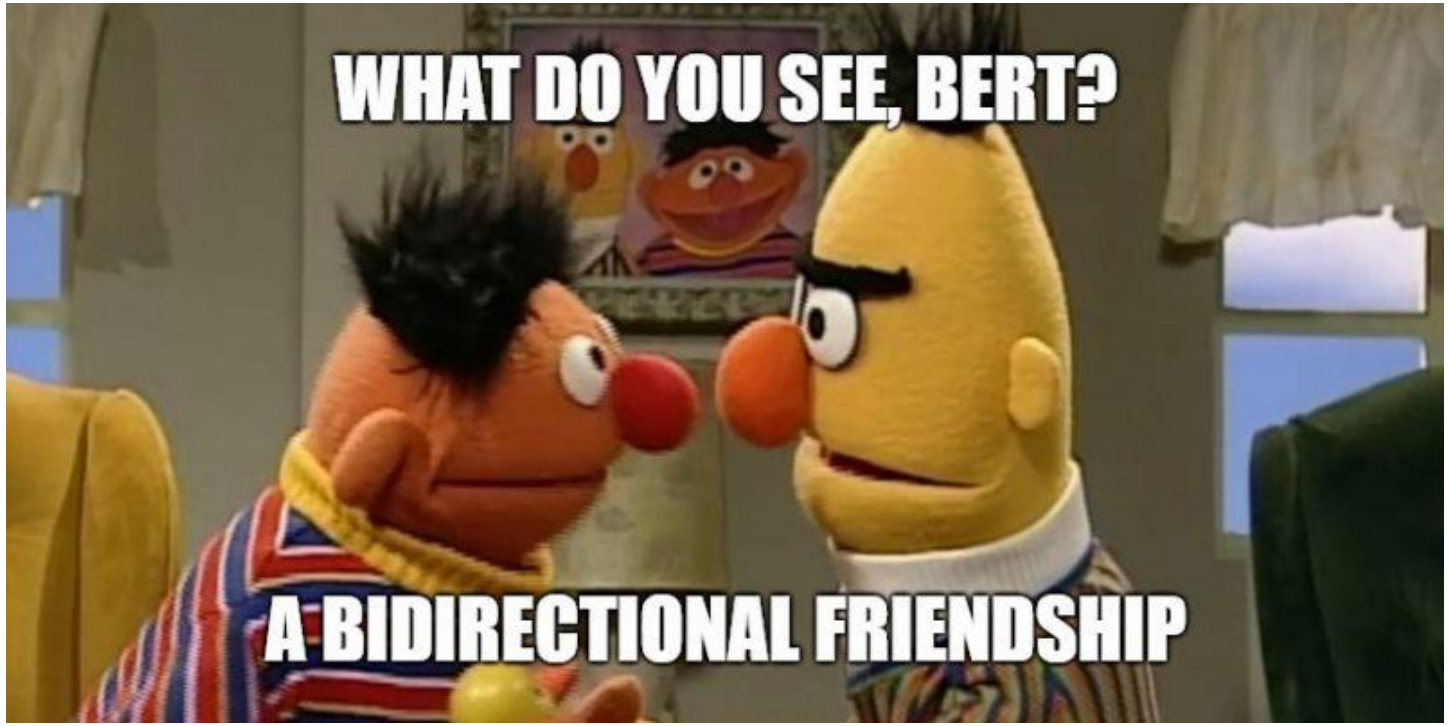
    *on a **large text corpus** (BooksCorpus and Wikipedia),*

    *and then use that model for **downstream NLP tasks (fine-tuning)** that we care about*

https://towardsdatascience.com/build-a-bert-sci-kit-transformer-59d60ddd54a5

BERT

- **Bidirectional Encoder Representation from Transformer**
  - **Transformer (like GPT)**
  - **Bi-directional (like ELMo)**
  - **Use only Encoder**

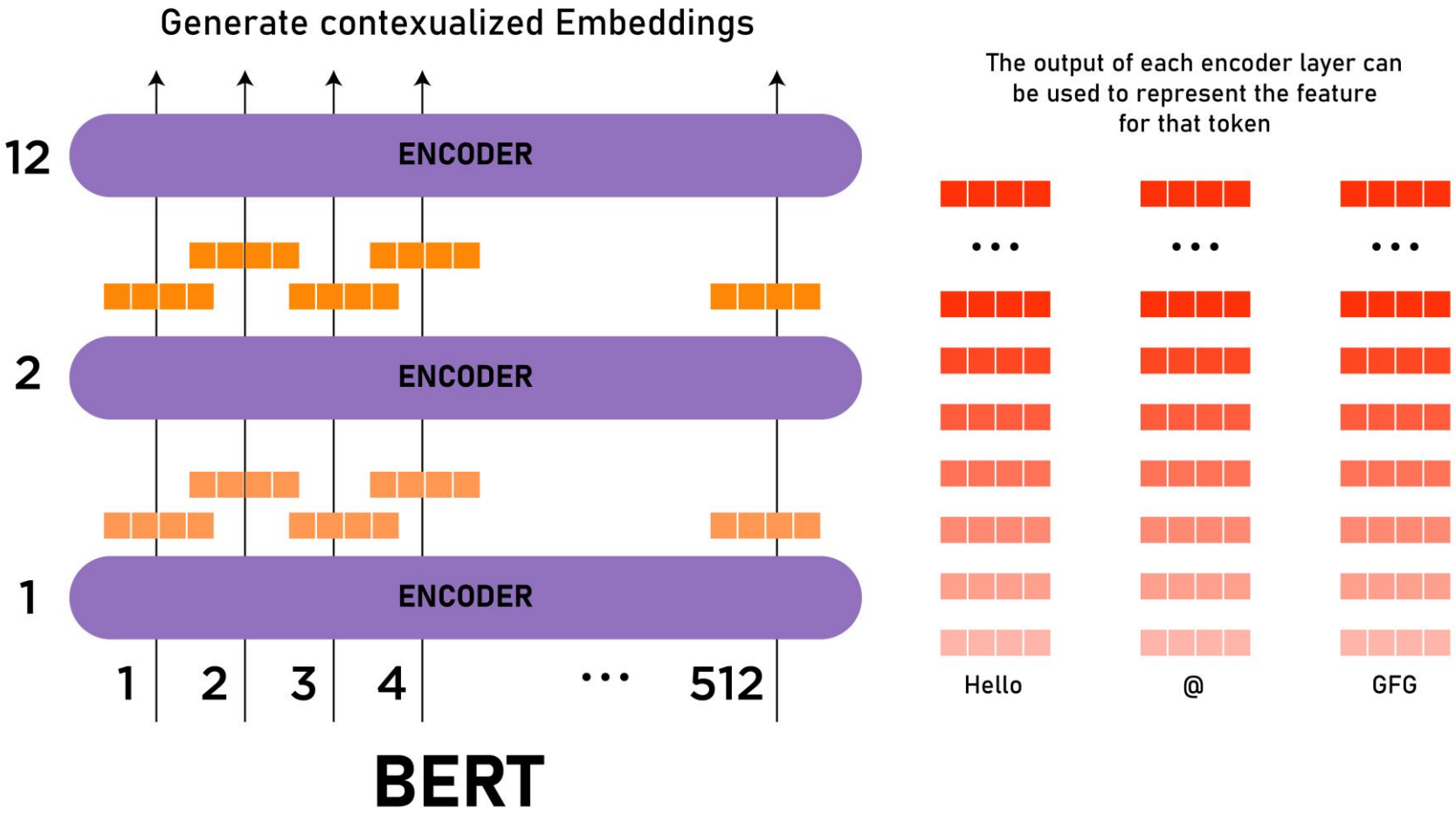


https://dsmilab.github.io/dsmi-lab-website/2020/05/13/BERT%20-%20Bidirectional%20Encoder%20Representations%20from%20Transformers/

BERT

- **Architecture**
  - 12 encoder layer
  - 12 attention head

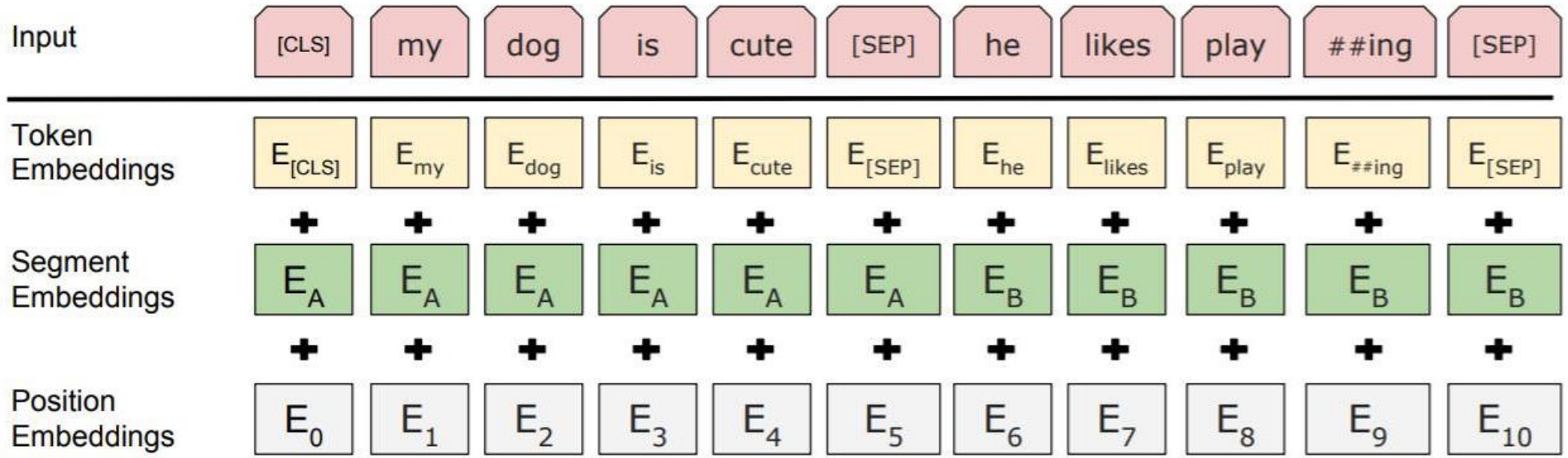
    (for each encoder)
  - Hidden unit size 768



https://www.geeksforgeeks.org/explanation-of-bert-model-nlp/

BERT

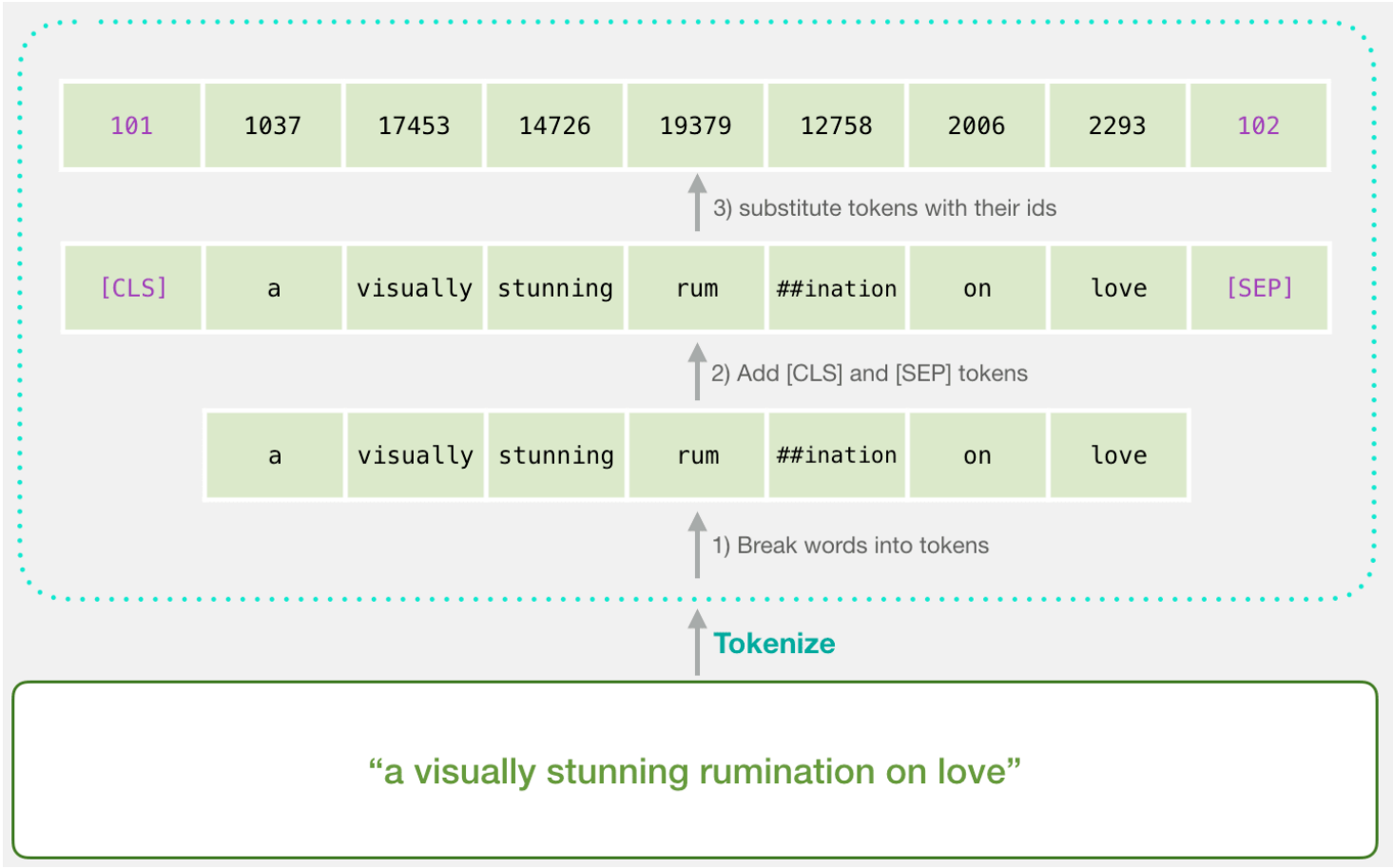
- **Input Data Representation**
  - 1) Token Embedding
  - 2) Segment Embedding
  - 3) Position Embedding



https://medium.com/@_init_/why-bert-has-3-embedding-layers-and-their-implementation-details-9c261108e28a

BERT

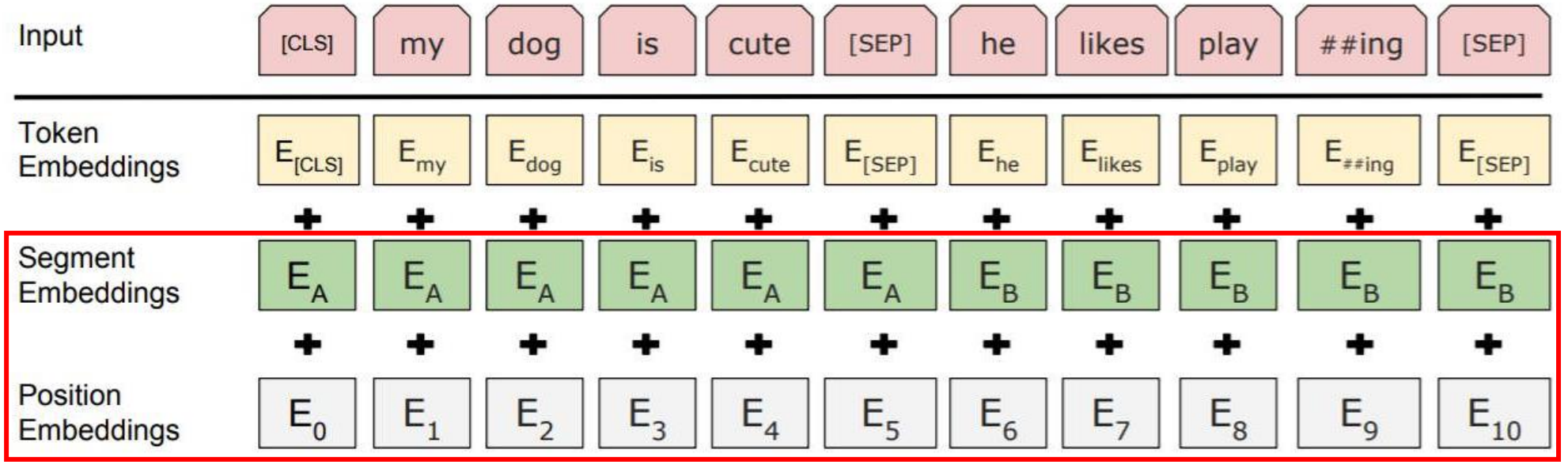
- **Input Data Representation**
  - 1) Token Embedding
    - By WordPiece embedding
    - **[CLS] token**
      - At the start of sentence
      - Represents the aggregate representation of sentence
      - Used for classification task
    - **[SEP] token**
      - At the end of sentence



https://is-rajapaksha.medium.com/bert-word-embeddings-deep-dive-32f6214f02bf

BERT

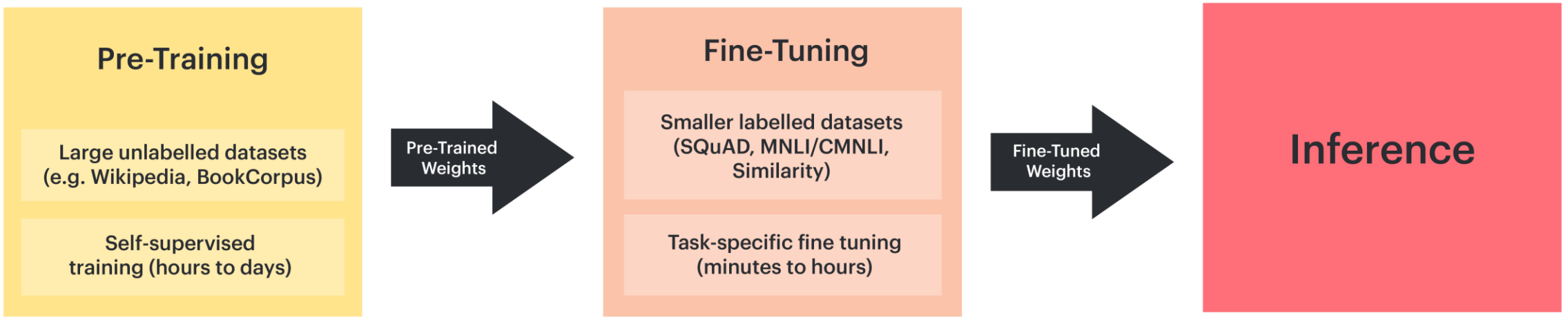
- **Input Data Representation**
  - 2) Segment Embedding → Used to distinguish different sentence
  - 3) Position Embedding → Used to take account the position of token in the sentence



https://medium.com/@_init_/why-bert-has-3-embedding-layers-and-their-implementation-details-9c261108e28a

BERT

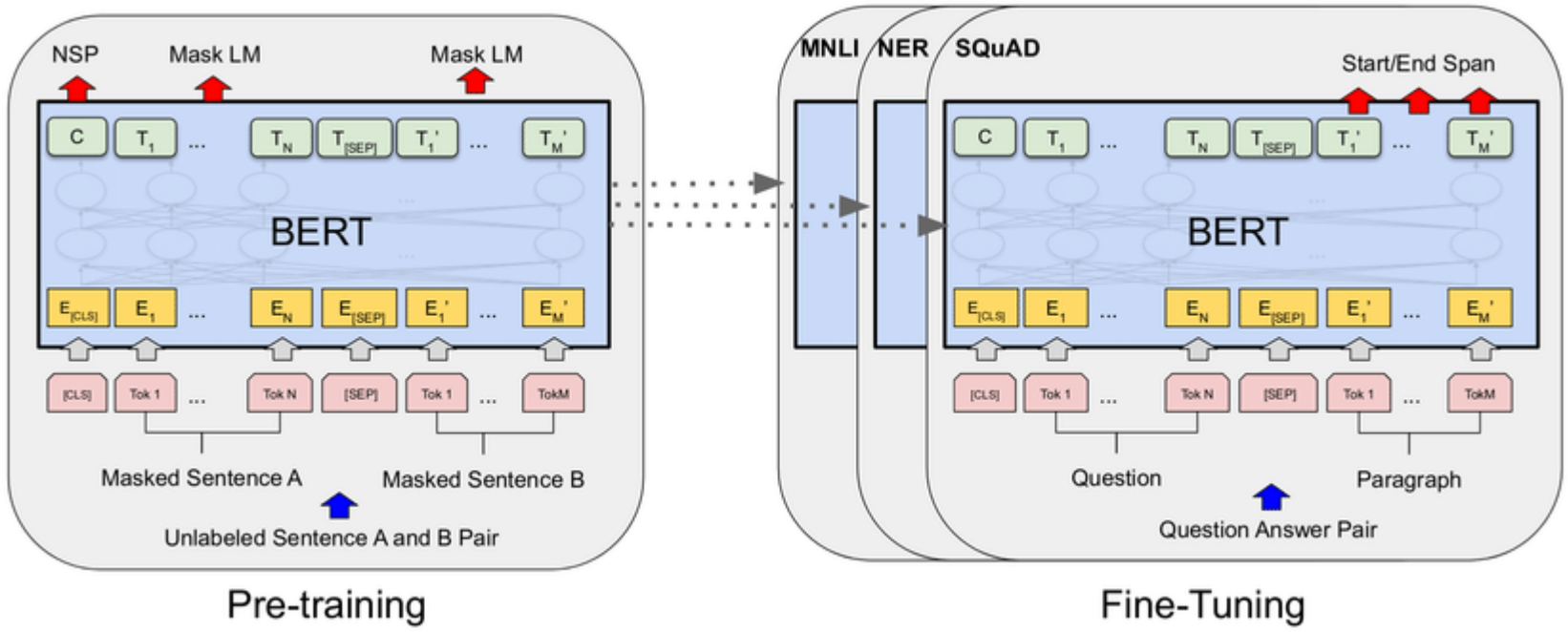
- **Pre-training & Fine-tuning**



https://docs.graphcore.ai/projects/bert-training/en/latest/bert.html

BERT

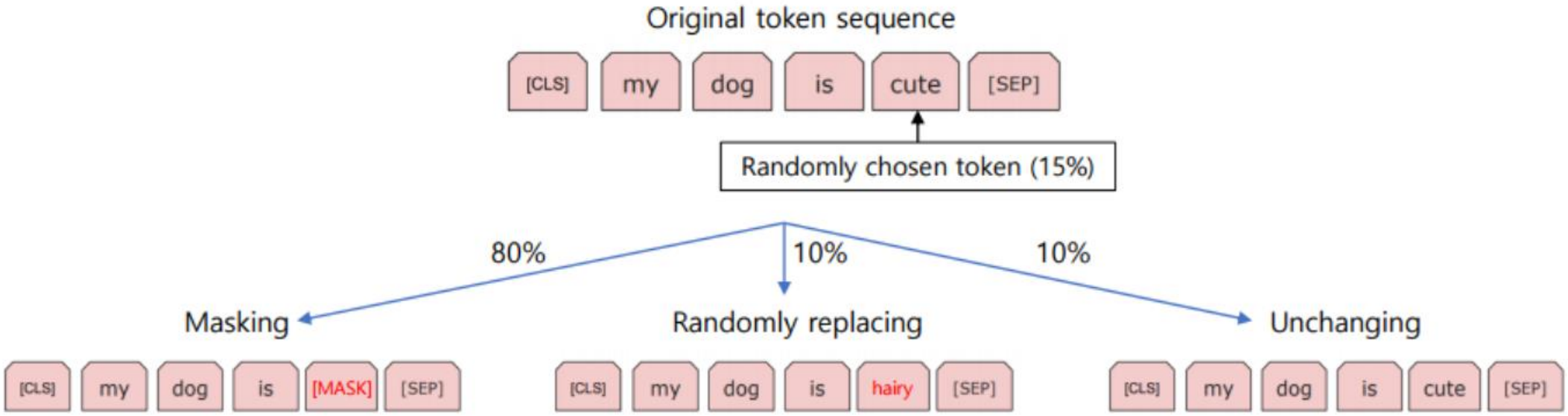
- **Pre-training**
  - 1) MLM (Masked Language Modelling)
  - 2) NSP (Next Sentence Prediction)



https://www.researchgate.net/figure/Overall-BERT-pretraining-and-finetuning-framework-Note-that-the-same-architecture-in_fig1_334160936

BERT

- **Pre-training**
  - **MLM (Masked Language Modelling)**
    - Randomly mask 15% of words and train the network to **predict the masked words**
      - 80-10-10% rule (80% masking, 10% randomly replacing, 10% unchanging)

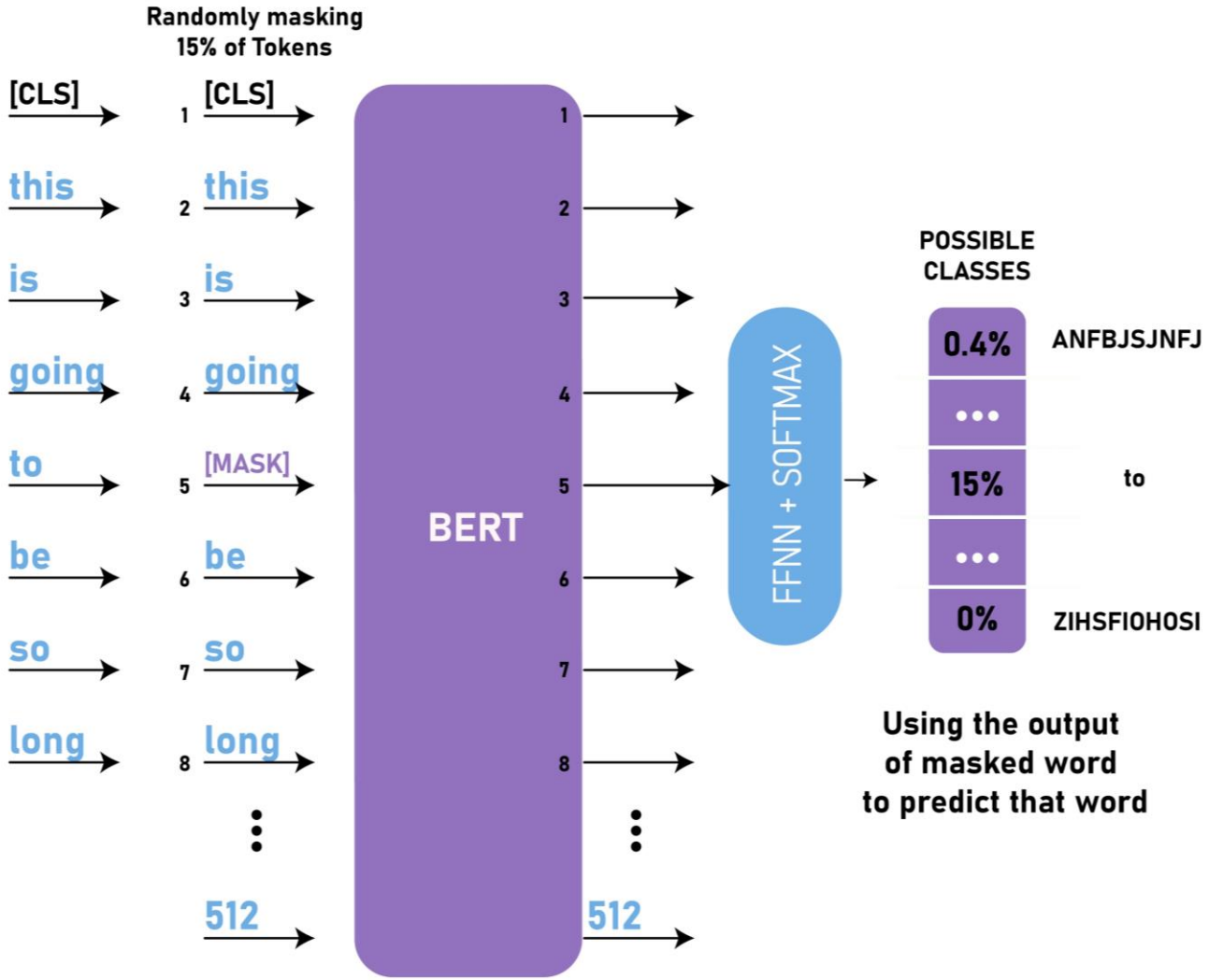


BERT

- **Pre-training**
  - **MLM (Masked Language Modelling)**
    - Reads the sentence

      in both directions

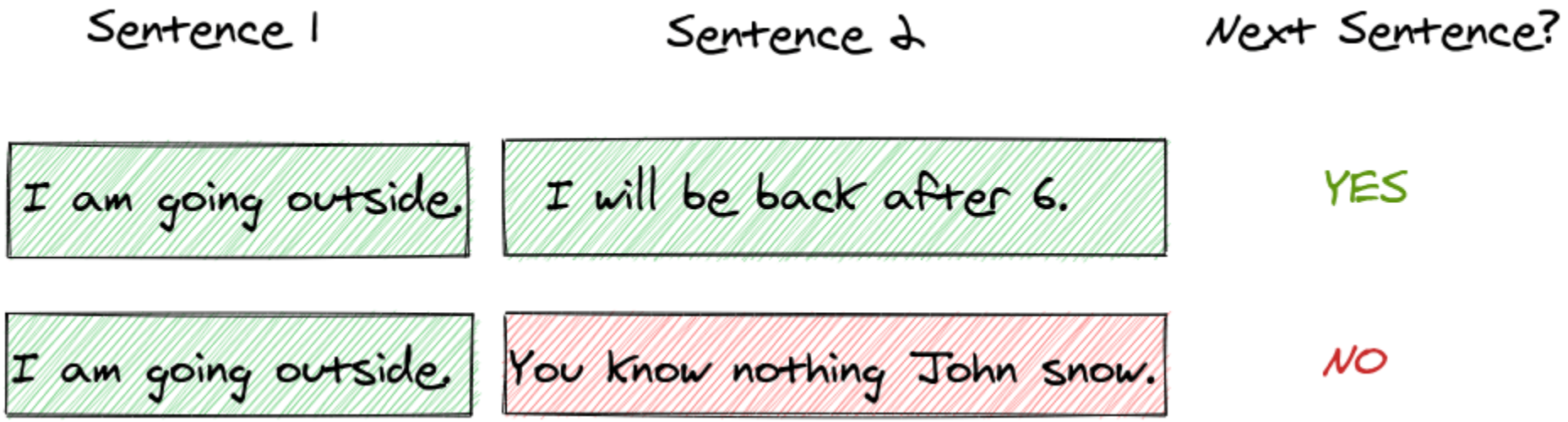
      to **predict the masked words**



https://www.geeksforgeeks.org/understanding-bert-nlp/

BERT

- **Pre-training**
  - **NSP (Next Sentence Prediction)**
    - Binary classification

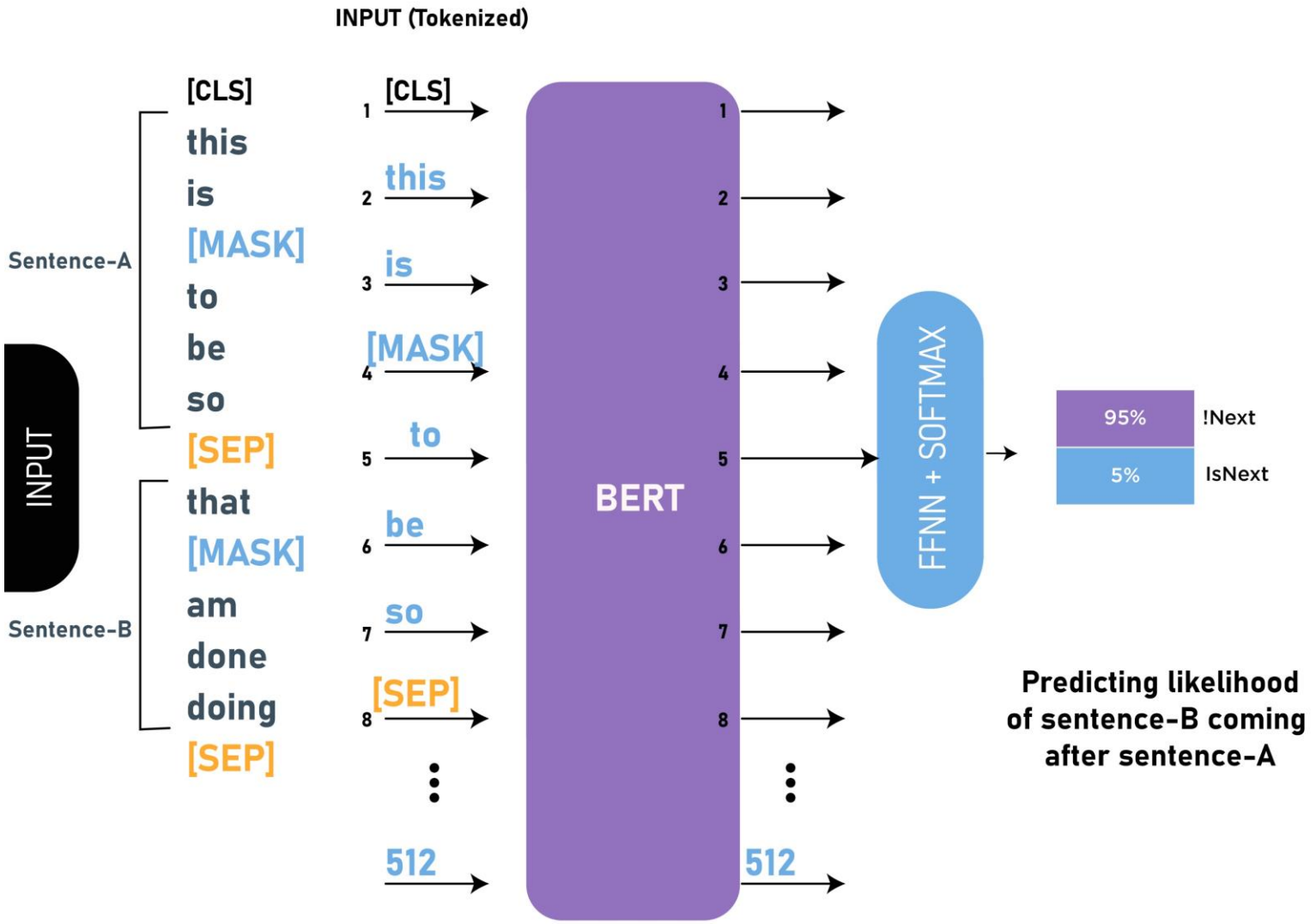


https://amitness.com/2020/02/albert-visual-summary/

BERT

- **Pre-training**
  - **NSP**

    (Next Sentence Prediction)
    - Binary classification

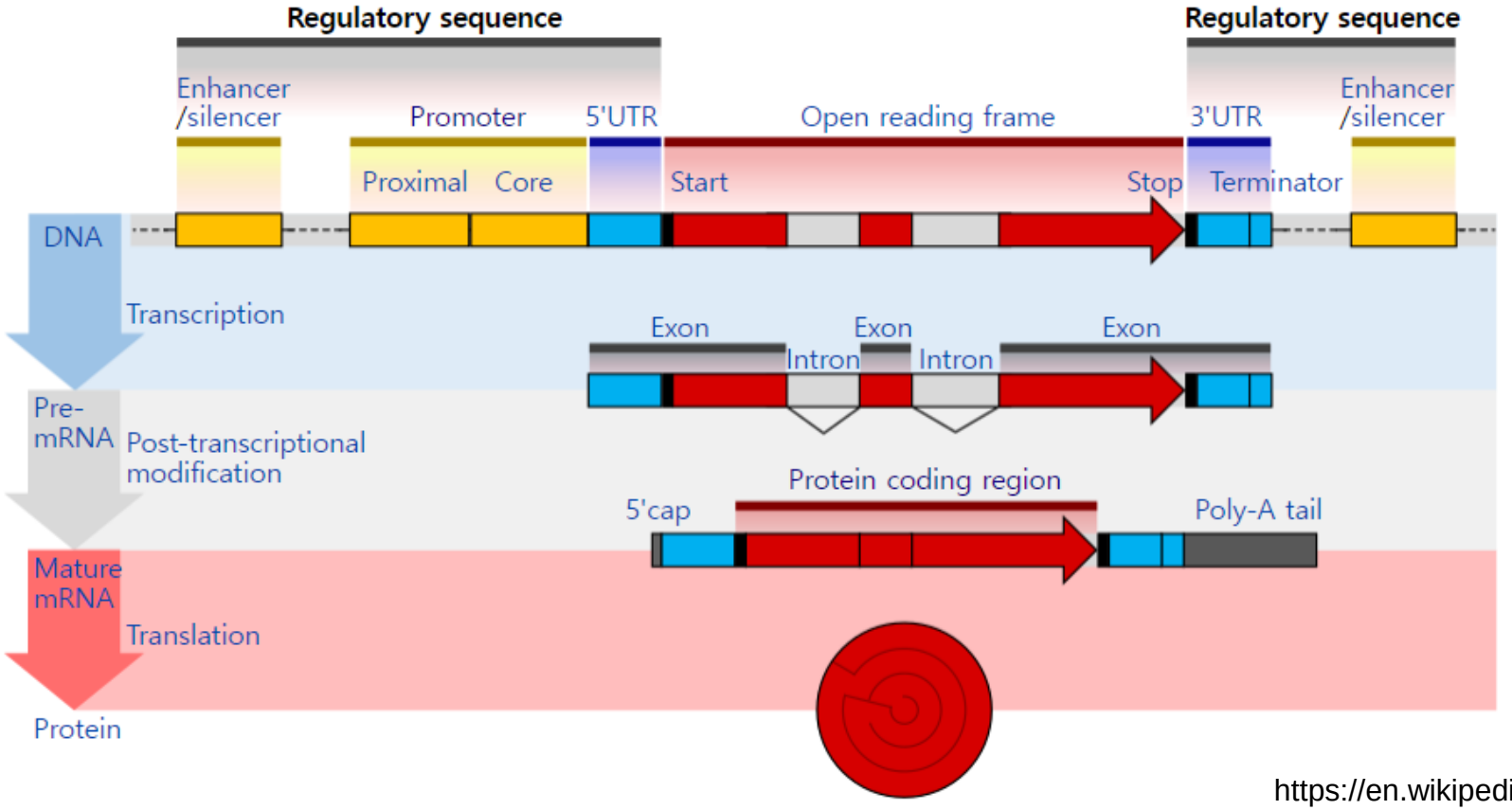


https://www.geeksforgeeks.org/understanding-bert-nlp/

DNABERT

Introduction

- **Deciphering the language of DNA for hidden instructions**
  - Regulatory sequence
    - Capable of increasing or decreasing the expression of specific genes

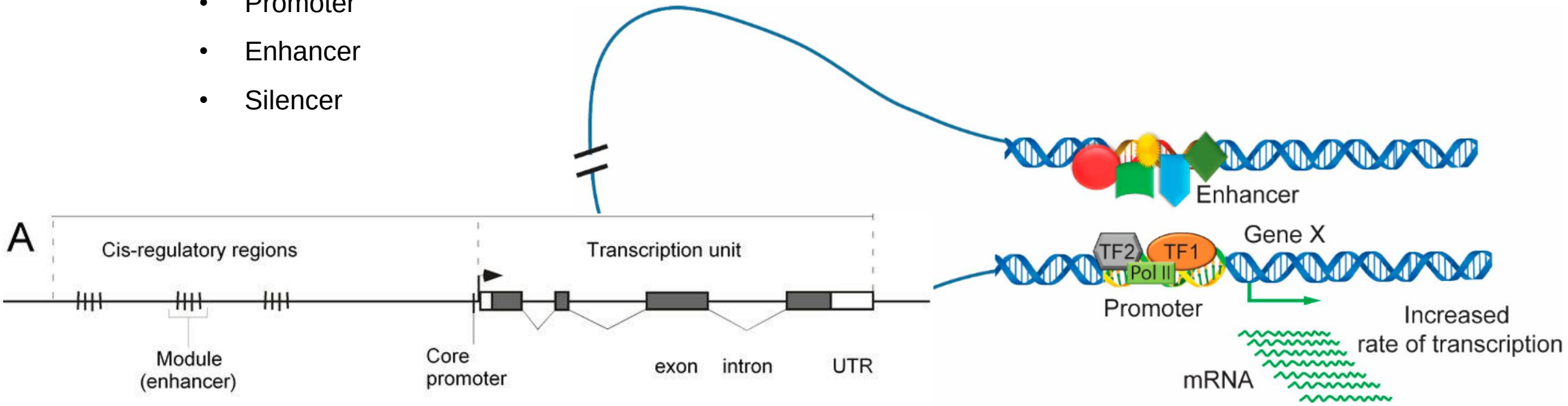


https://en.wikipedia.org/wiki/Regulatory_sequence

Introduction

- **Deciphering the language of DNA for hidden instructions**
  - Regulatory code
    - CRE (*cis*-regulatory elements)

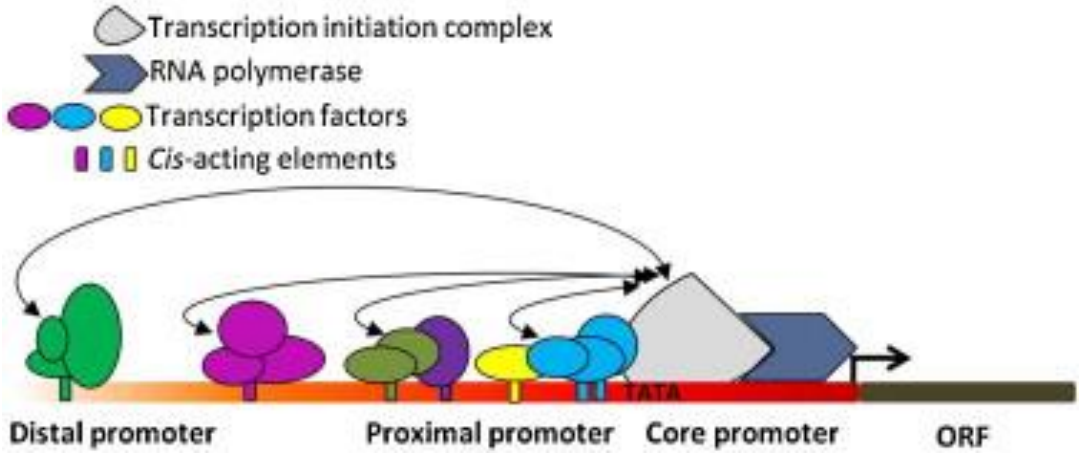
      : regions of non-coding DNA which regulate the transcription of **neighboring** genes
      - Promoter
      - Enhancer
      - Silencer

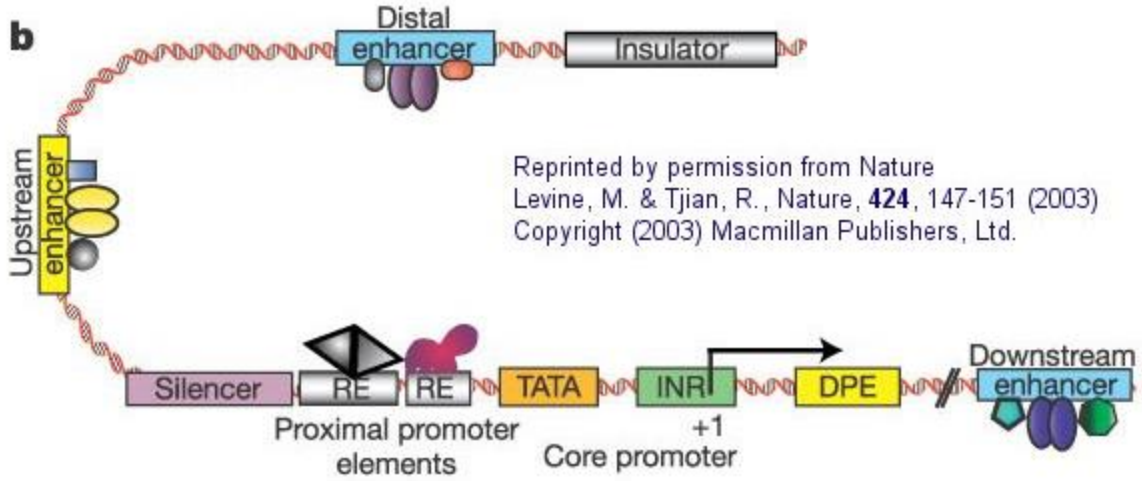


https://www.researchgate.net/figure/Promoter-structure-and-function-in-eukaryotic-genes-A-Relative-position-of-the_fig3_279181186
https://www.mdpi.com/2073-4409/8/10/1281

Introduction

- **Deciphering the language of DNA for hidden instructions**
  - Regulatory code
    - CRE (*cis*-regulatory elements)
      - **Promoter** : Core / Proximal / Distal
        - Core promoter : Include TATA-box
        - Proximal & Distal promoter : Include enhancer / silencer / insulator / etc.





https://www.sciencedirect.com/science/article/pii/S0168945213002689

Introduction

- **Deciphering the language of DNA for hidden instructions**
  - Regulatory code
    - Same CREs often have **distinct functions and activities in different biologic contexts**
      - e.g., different cell-types and organisms
      - Cooperation of widely spread multiple CREs → context-dependent use of alternative promoters
    - Suggest polysemy and distant semantic relationship within sequence codes
    - How the semantics (i.e., functions) of CREs vary across different contexts

Introduction

- **Deciphering the language of DNA for hidden instructions**
  - Ideal computational method
    - 1) Globally take all the **contextual information** into account to distinguish polysemous CREs
    - 2) Develop **generic understanding transferable to various tasks**
    - 3) **Generalize well** when labeled data is limited
    - ⇒ Transformer
      - Develop ***global contextual embedding*** via self-attention

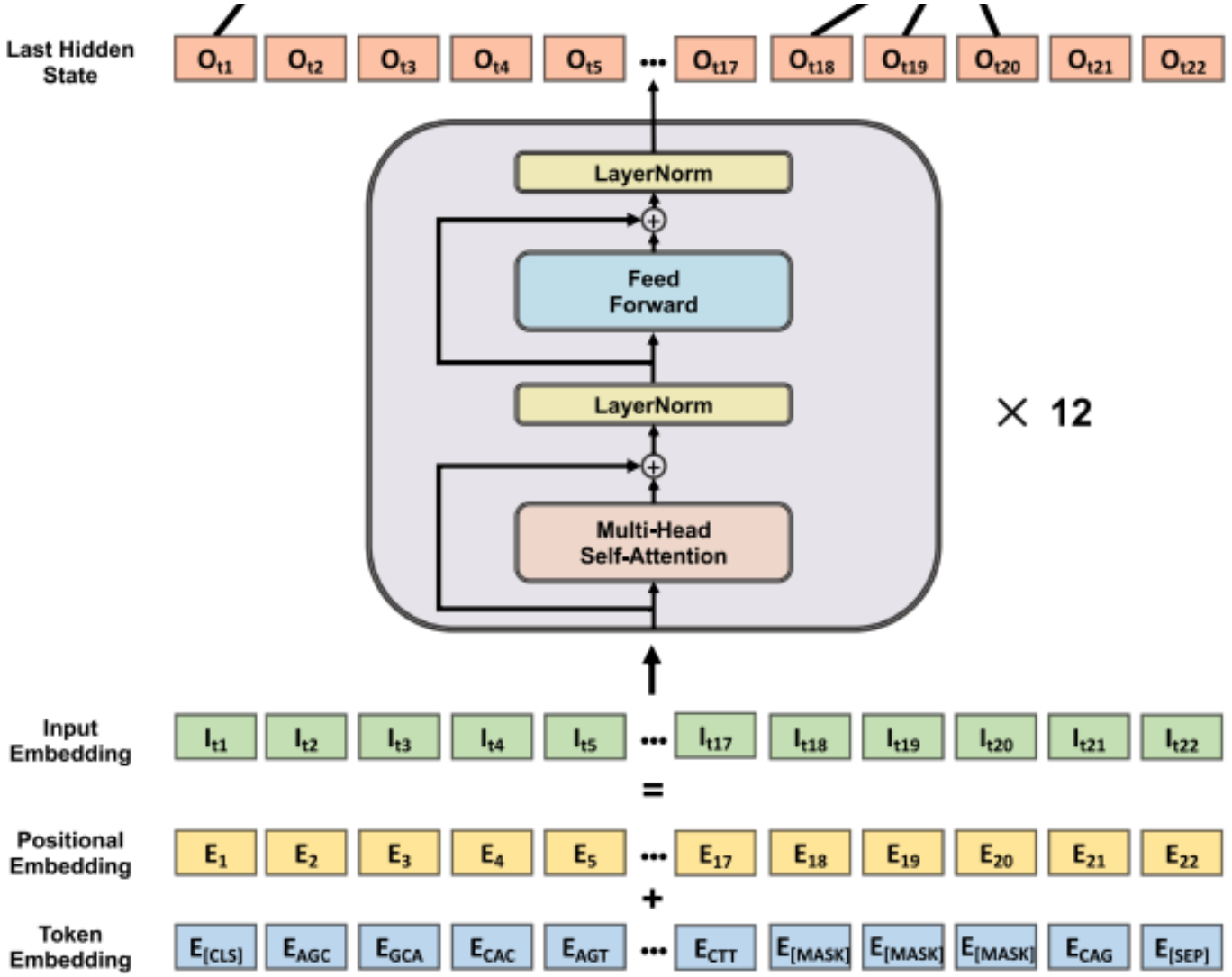
https://imgflip.com/i/425mi0

Introduction

- **DNABERT**
  - Applies transformer (attention-based architecture)
  - It resolves
    - 1) Develop **general and transferable understanding of DNA** from the purely **unlabeled** human genome, and utilize them to generically solve various **sequence related tasks in a *"one-model-does-it-all"* fashion**
    - 2) Globally capture **contextual information** from the entire input sequence with attention mechanism
    - 3) Achieve great performance in **data-scarce scenarios**
    - 4) **Uncover important subregions and potential relationships** between different *cis*-elements of DNA sequence without any human guidance
    - 5) Successfully work in a **cross-organism manner**

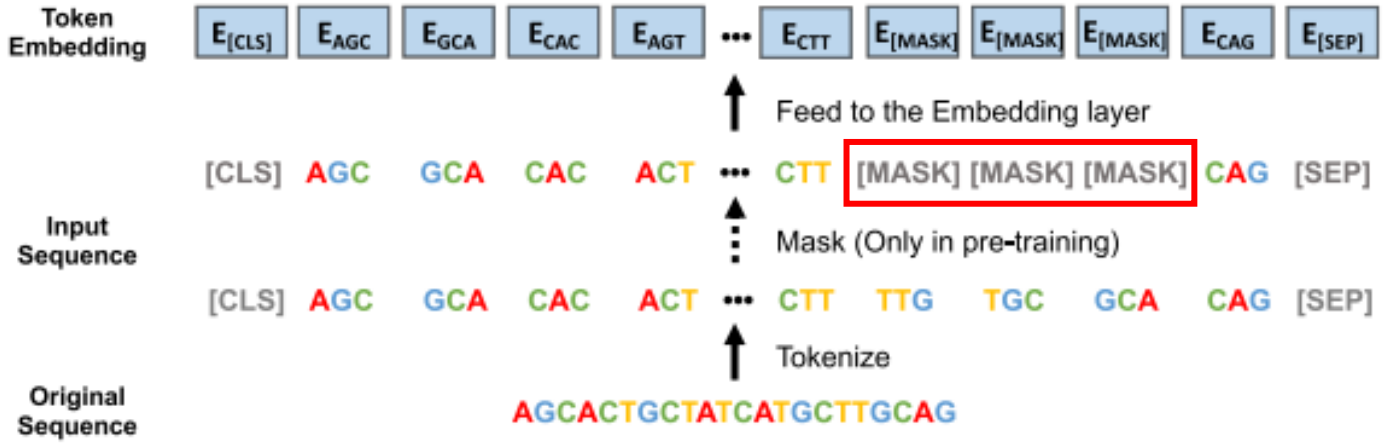
Materials & Method

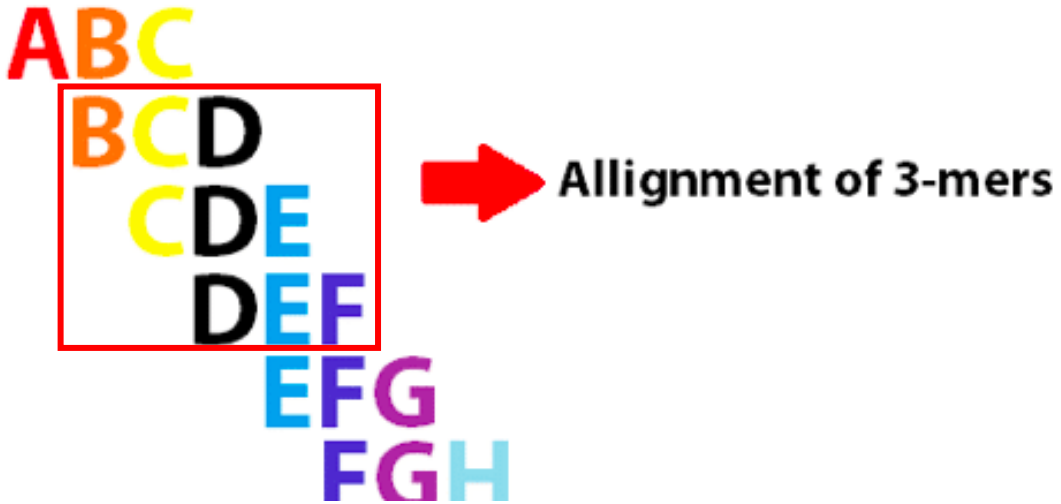
- **DNABERT – Architecture**
  - Same architecture with BERT-base
    - 12 transformer layers
    - 12 attention heads in each layer
    - 768 hidden units



Materials & Method

- **DNABERT – Architecture**
  - Input
    - **k-mer tokens (k = 3, 4, 5, 6)**
    - Masked contiguous k-length spans of k-mers
      - Masked k-mer can be trivially made up based on its previous and next k-mers
      - Prevent the model from learning deep semantic relations among a DNA sequence
      - ex) 3-mer sequence

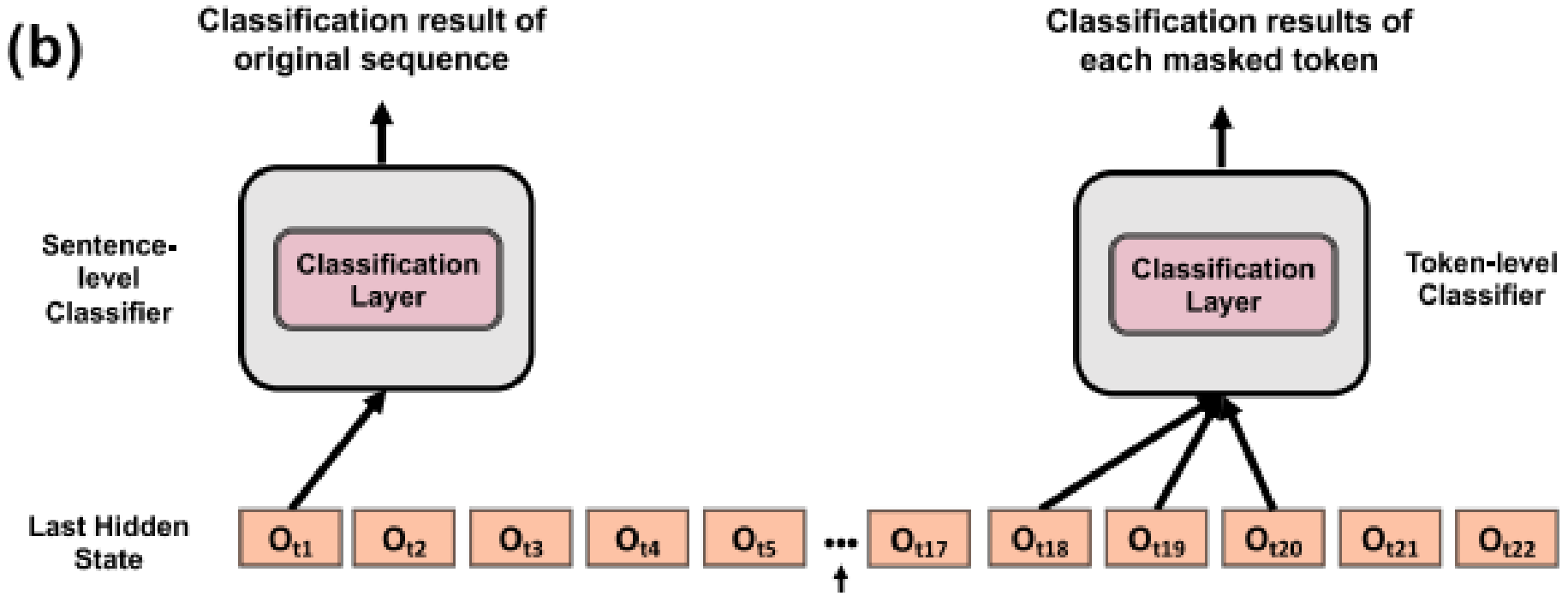




https://bitesizebio.com/38250/de-novo-dna-sequencing-and-the-special-k-mer/

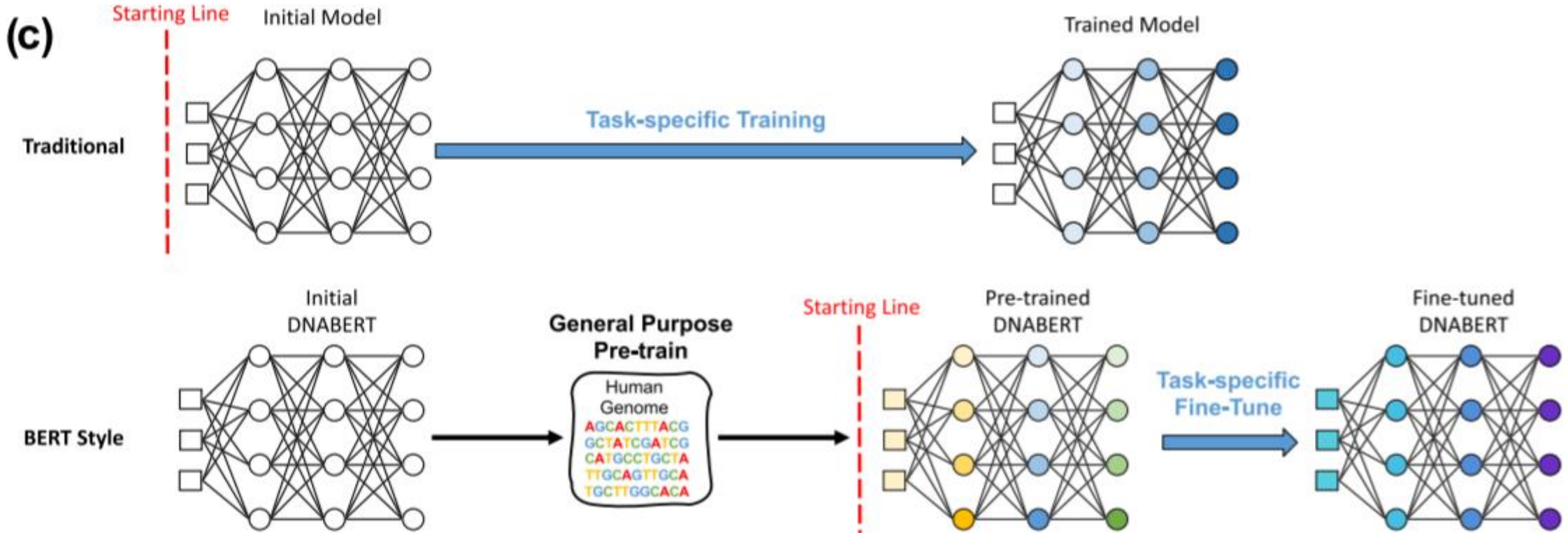
Materials & Method

- **DNABERT – Architecture**
  - Output
    - [CLS] token representation → Sentence-level classification
    - Individual masked token representations → token-level classification



Materials & Method

- **DNABERT – Pre-training**
  - First develops **general-purpose understandings** from massive amount of **unlabeled data**, then resolves various applications with **task-specific data** with minimal architectural modification



Materials & Method

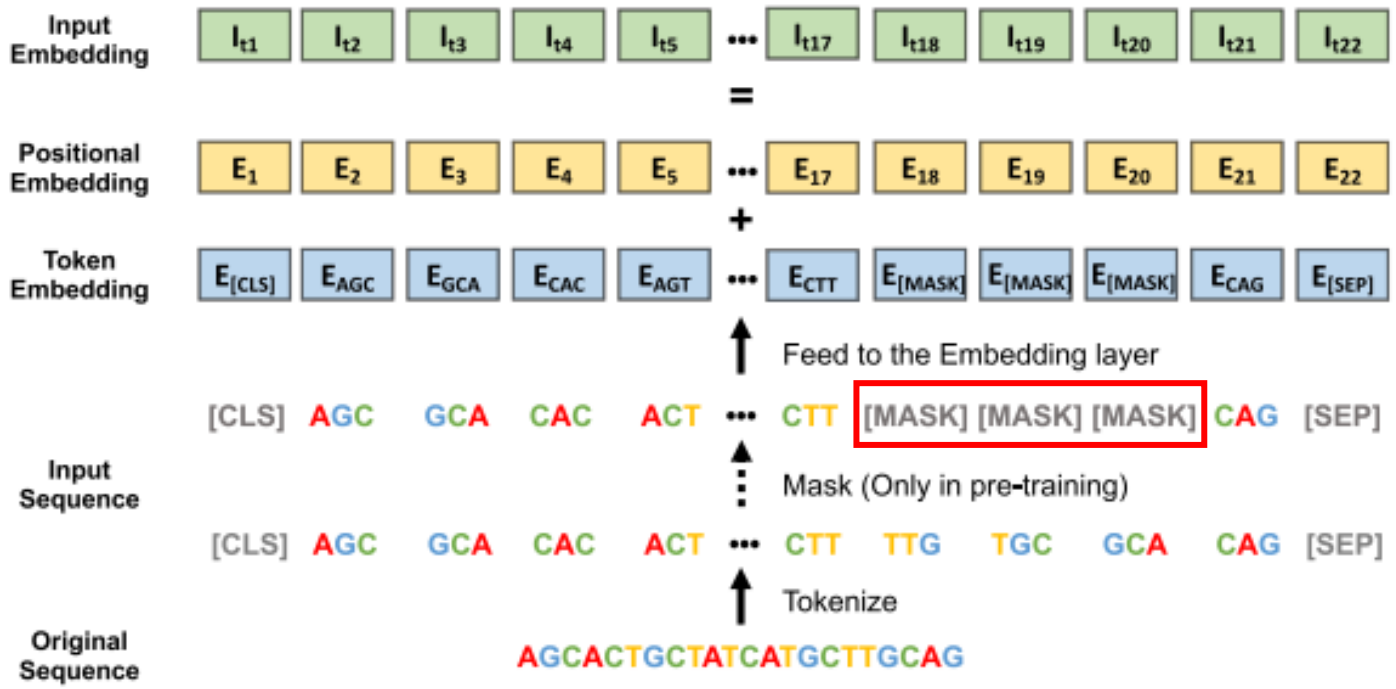
- **DNABERT – Pre-training**

  - Difference from BERT
    - Skipped pre-training with NSP task
    - Adjusted the sequence length and forced the model to predict contiguous k tokens

  - During pre-training
    - DNABERT learns basic syntax and semantics of DNA via self-supervision
    - Based on 10~510 length sequences extracted from human genome via truncation and sampling

Materials & Method

- **DNABERT – Pre-training**
  - Take a sequence
    - Maximum length 512
      - Direct non-overlap sampling
      - Random sampling
  - Masking
    - Mask a contiguous k-length spans of certain k-mers
    - Total 15% of input sequence



Materials & Method

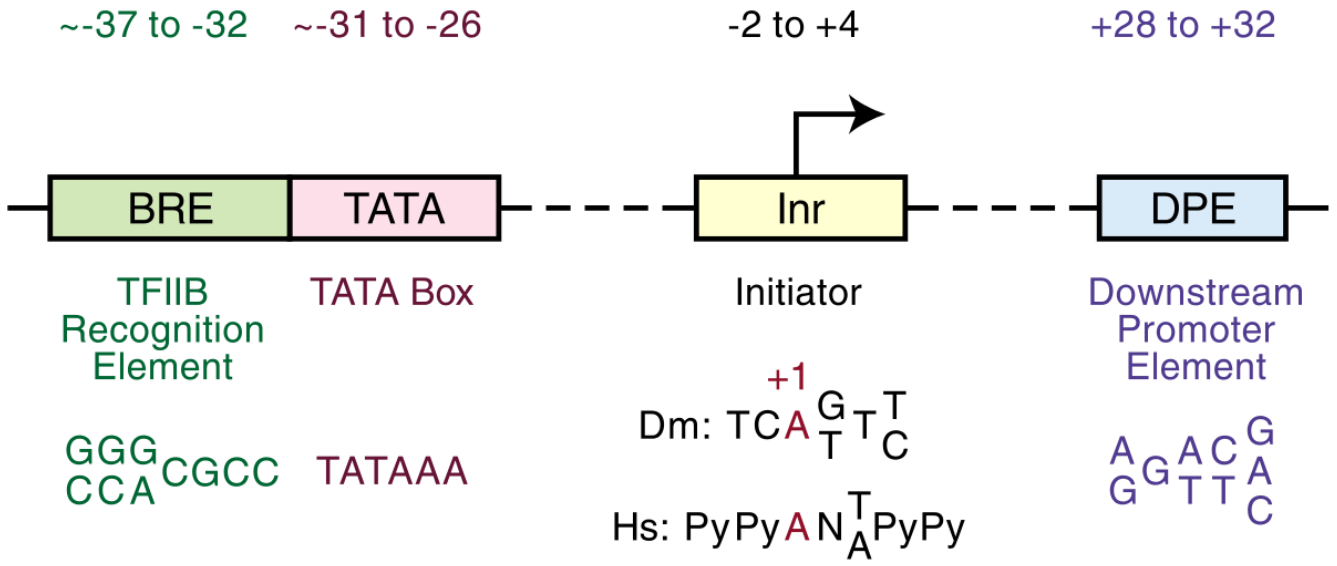
- **DNABERT – Pre-training**
  - Hyper-parameters
    - Epoch : 120k steps
      - Learning rate linearly increased (warm-up) from 0~4e^{-4} in the first 10k steps and then linearly decreased to 0 after 200km steps
      - Stopped training procedure after 120k steps (loss curve showed plateau)
    - Batch size 2000
    - Objective function : Cross-entropy loss calculated over all the masked k-mers
    - Optimizer : AdamW ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1e^{-6}$, weight decay = 0.01)

Materials & Method

- **DNABERT – Fine-tuning**
  - Fine-tuning by task-specific training data
    - 1) Promoter prediction
    - 2) Transcription factor binding sites prediction
    - 3) Splice site prediction

Results

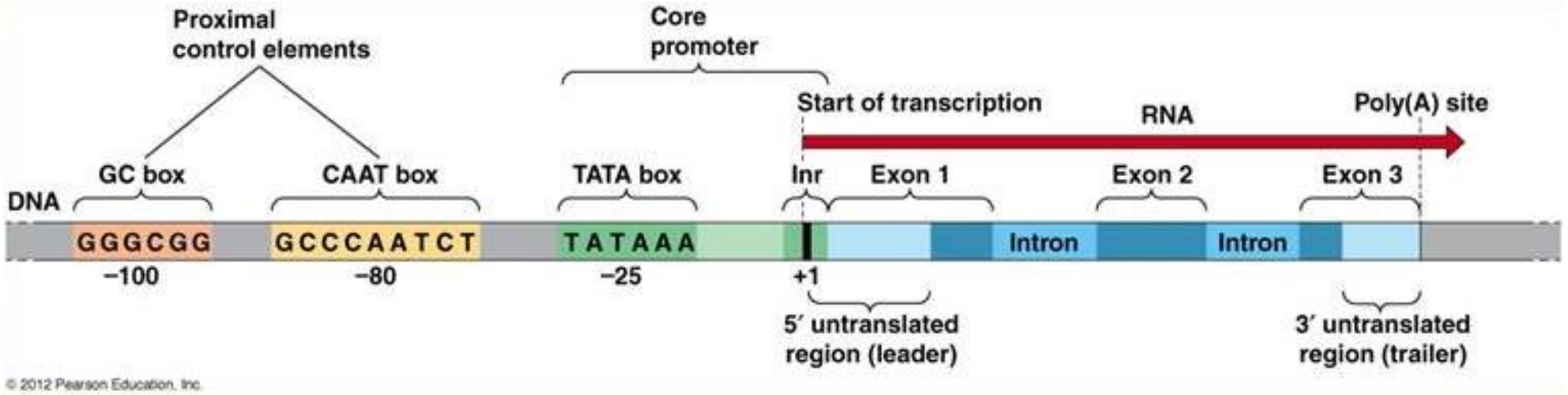
- **1. DNABERT-Prom effectively predicts proximal and core promoter regions**
  - Data
    - EPDnew (Eukaryotic Promoter Database) : Annotated non-redundant / Eukaryotic Pol II promoters
    - 3,065 human TATA & 26,533 non-TATA promoter-containing sequences
      - -5,000 ~ +5,000 bp (+1 : position of TSS)
  - Model
    - DNABERT-Prom-300
    - DNABERT-Prom-scan



https://en.wikipedia.org/wiki/Downstream_promoter_element

Results

- **1. DNABERT-Prom effectively predicts proximal and core promoter regions**
  - DNABERT-Prom-300
    - Binary classification
    - Positive class (promoters)
      - 300bp-long promoter sequences extracted from -249~+50bp around TSS position



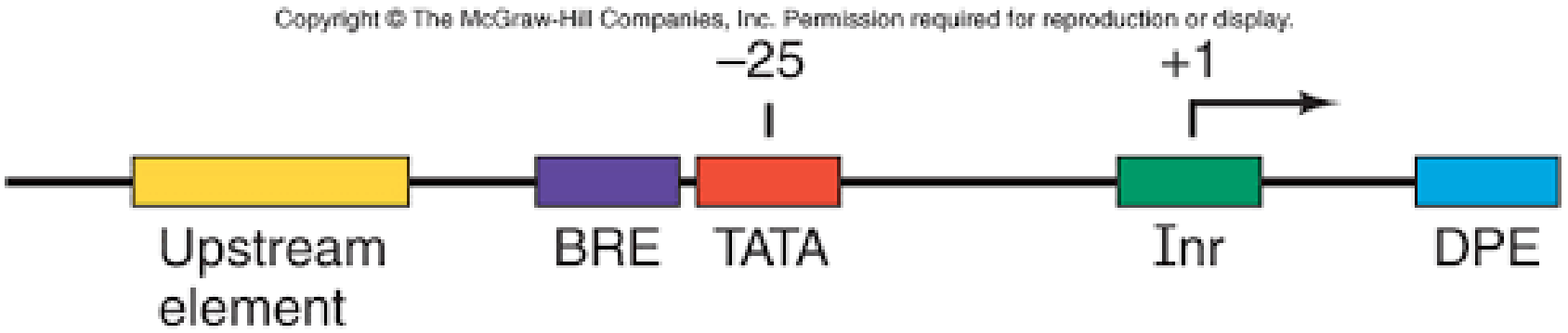
http://mol-biol4masters.masters.grkraj.org/html/Gene_Structure5B-Eukaryotic_Promoter_Structure_for_RNA_Polymerase_II.htm

Results

- **1. DNABERT-Prom effectively predicts proximal and core promoter regions**
  - DNABERT-Prom-300
    - Binary classification
    - Negative class (non-promoters)
      - **TATA-negative** : randomly picked 3,065 of 300bp genomic regions not within the -249~+50bp range

        but contains the TATA motif

        → ensure that the TATA-negative is similar to TATA promoters as possible



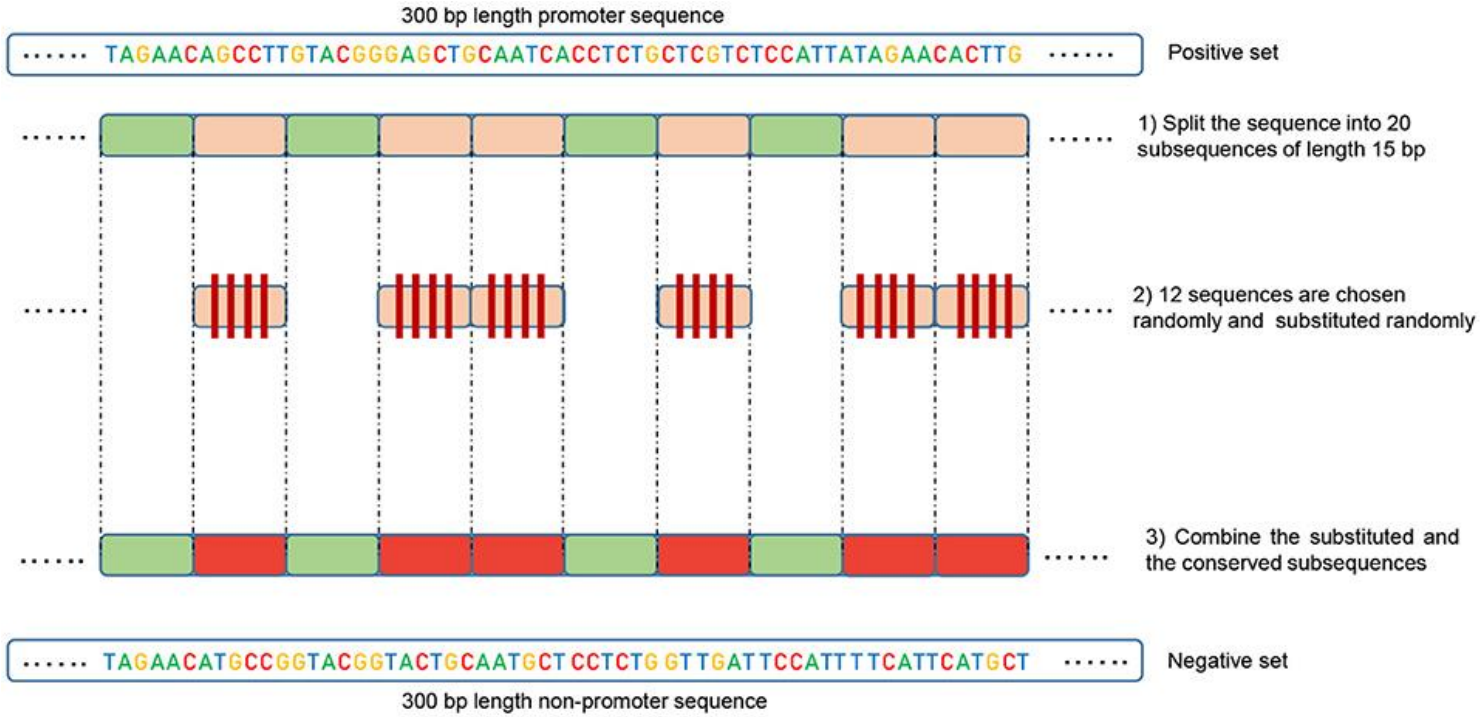
http://faculty.buffalostate.edu/wadswogj/courses/305/lectures/pol%20ii%20promoter.pdf

Results

- **1. DNABERT-Prom effectively predicts proximal and core promoter regions**
  - DNABERT-Prom-300
    - Binary classification
    - Negative class (non-promoters)
      - **Non-TATA negative**

        : random substitution approach

        (from DeePromoter paper)



https://www.frontiersin.org/articles/10.3389/fgene.2019.00286/full

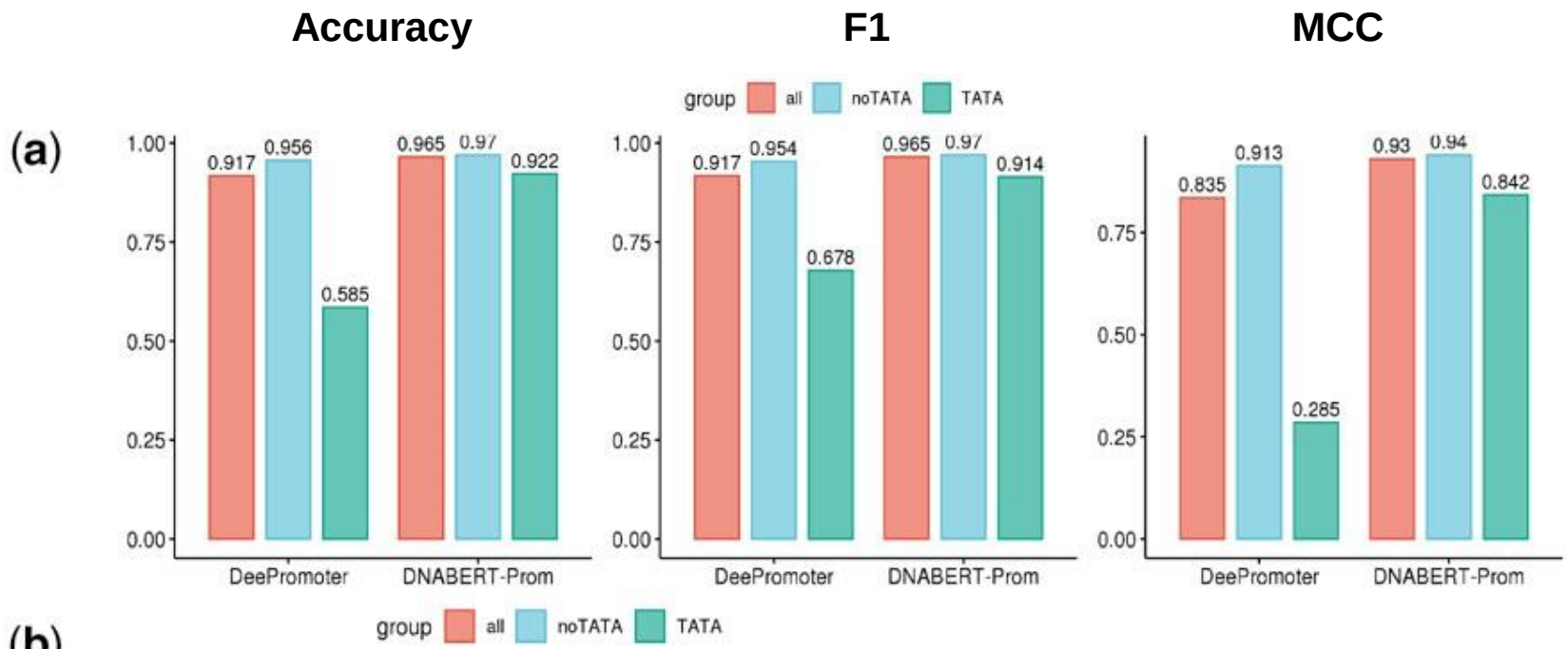
Results

- **1. DNABERT-Prom effectively predicts proximal and core promoter regions**
  - DNABERT-Prom-scan
    - Mimics real-world situations
      - Scan very long genomic regions w/ sliding window and obtain 1001bp-long sequences for promoter region
    - Scanned all 10,000bp-long sequences from EPDnew with a step size of 100
    - Evaluation
      - If the predicted sequence has ≥ 50% (i.e., 500bp) overlap with -500~+500 bp regions of TSS

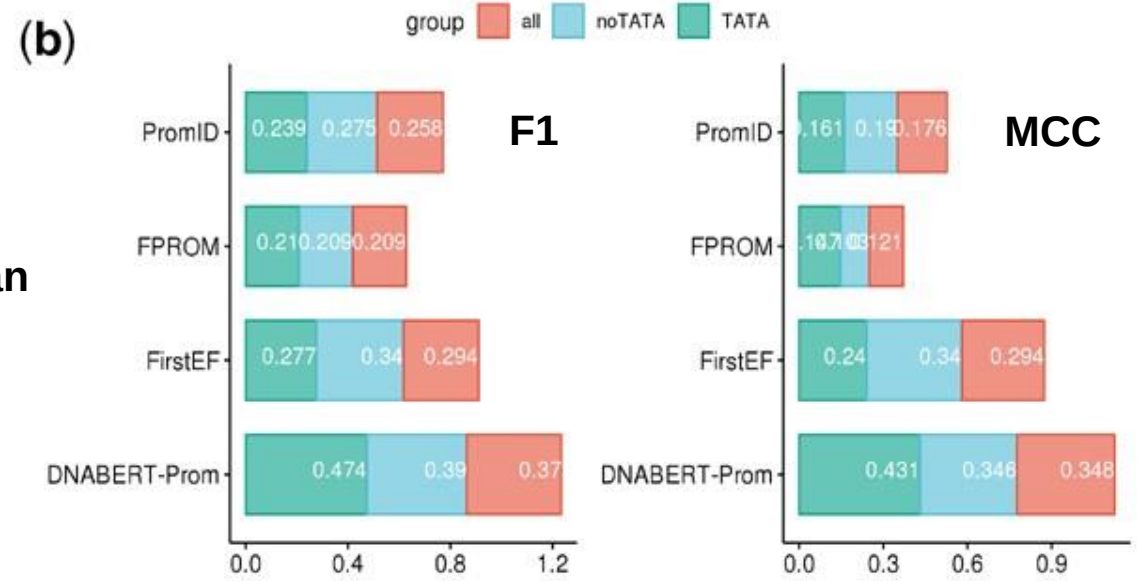
        ⇒ True Positive
      - Otherwise ⇒ False Positive
      - Failure to make prediction in the area of -500~+500 bp ⇒ False Negative

Results

DNABERT-Prom-300



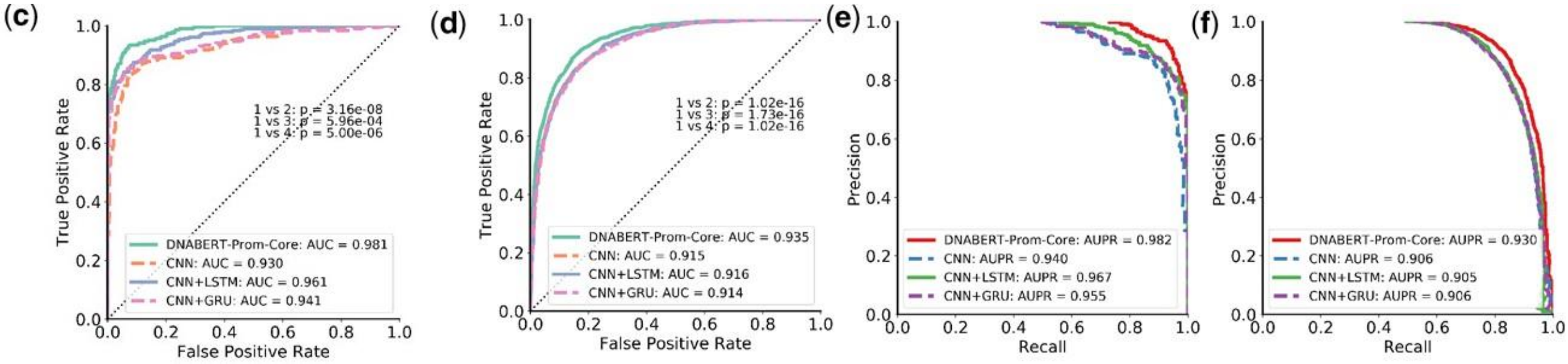
DNABERT-Prom-scan



→ DNABERT significantly outperforms other models in identifying promoter regions

Results

- **1. DNABERT-Prom effectively predicts proximal and core promoter regions**

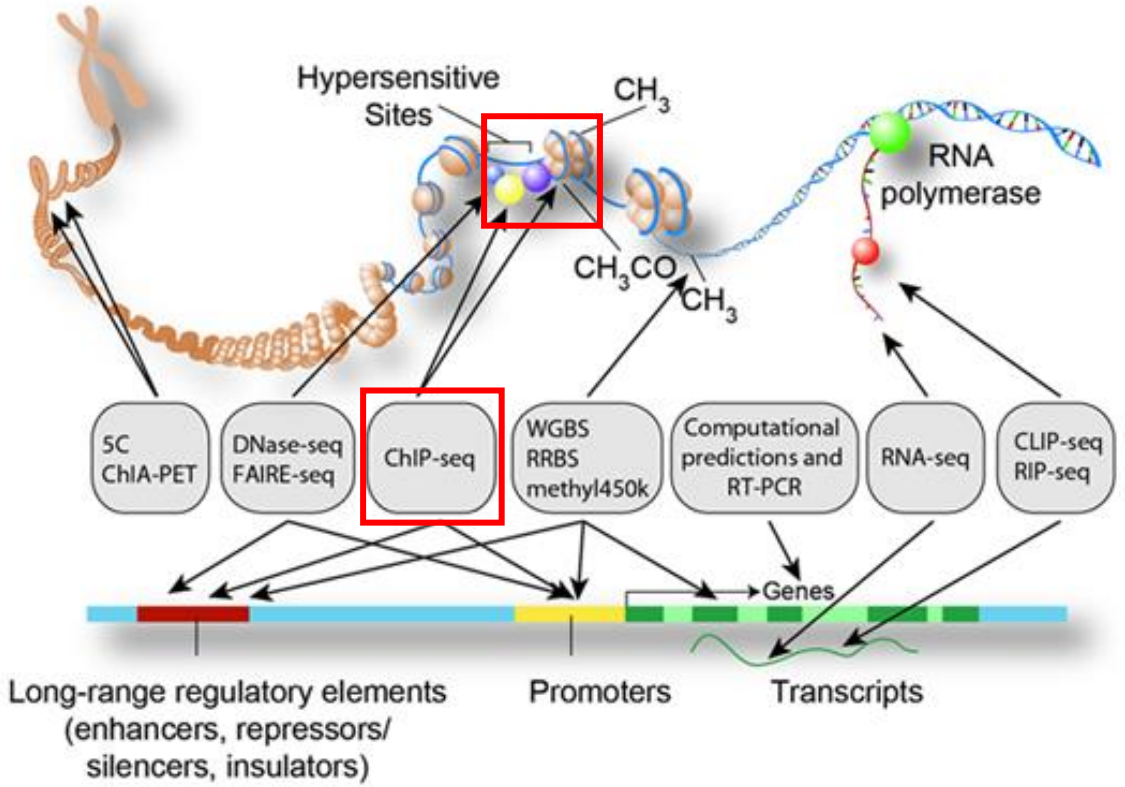


TATA **noTATA** **TATA** **noTATA**

→ DNABERT significantly outperforms other models in identifying promoter regions

Results

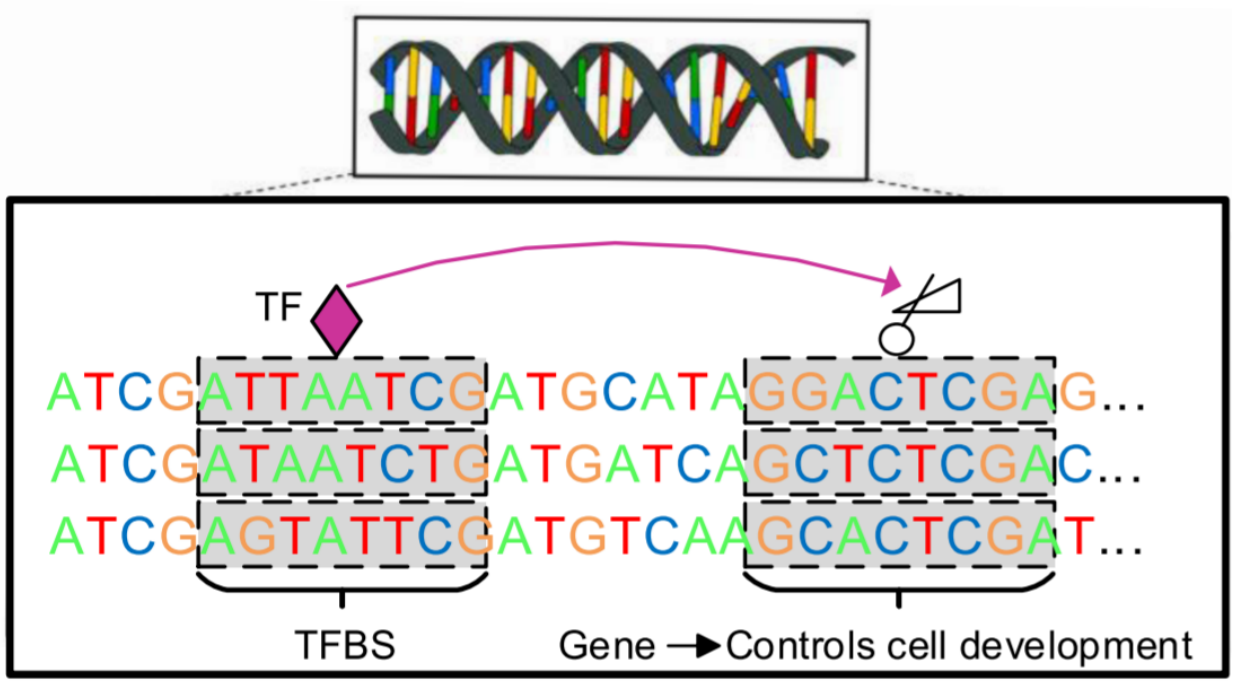
- **2. DNABERT-TF accurately identifies transcription factor binding sites**
  - Fine-tuned DNABERT to predict TFBS in ChIP-seq enriched regions
    - Using 690 TF ChIP-seq uniform peak profiles from ENCODE database
      - Positive set
        - extracted the centering 101-bp region around each ChIP-seq peak
      - Negative set
        - Similar approach with DESSO
        - Pick actual 101-bp long sequences not overlapping with any peaks with same GC content



https://www.encodeproject.org/?walkme=19-976547#!

Results

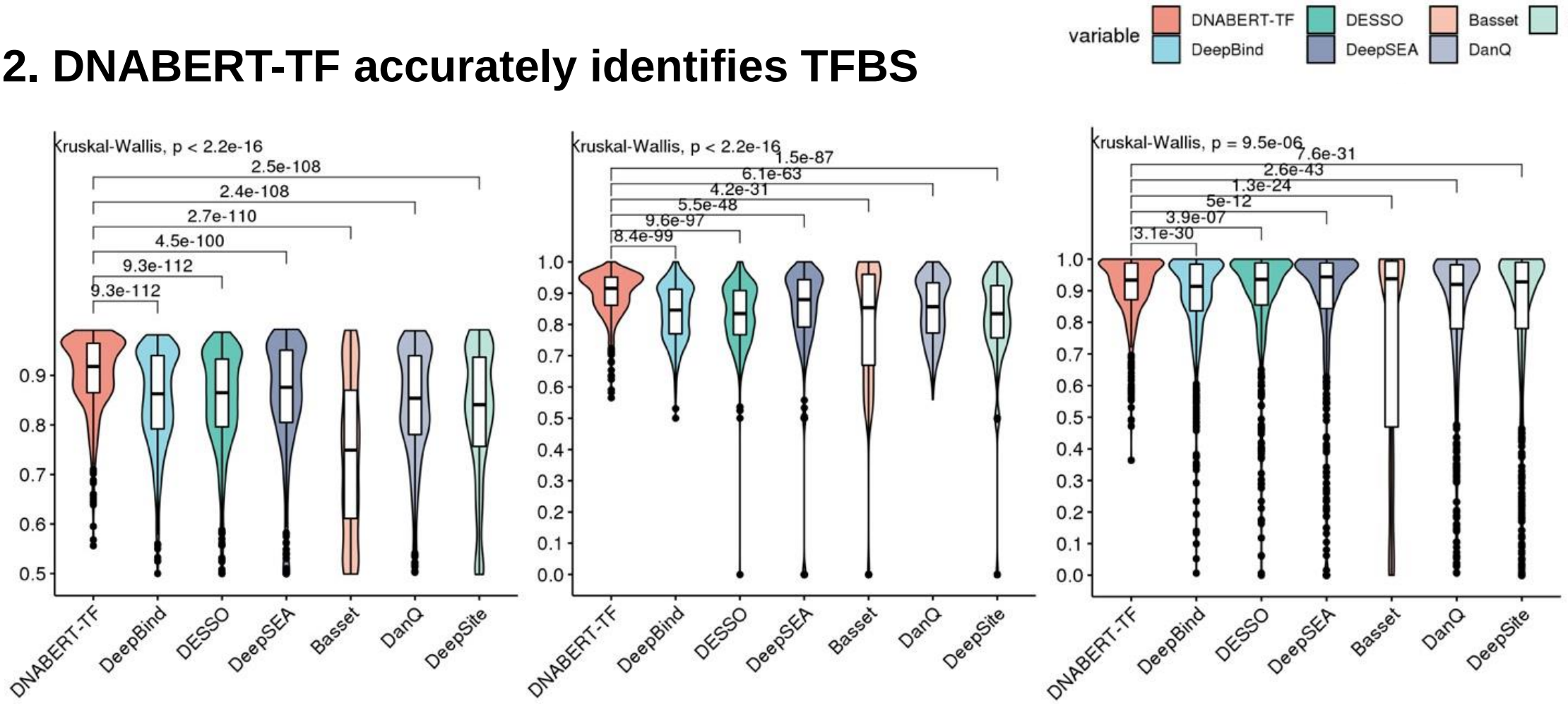
- **2. DNABERT-TF accurately identifies transcription factor binding sites**
  - Fine-tuned DNABERT to predict TFBS in ChIP-seq enriched regions
    - Binary classification using the top 500 even-numbered peaks as test set
    - Compared with well-known and previous published TFBS prediction tools
      - DeepBind, DeepSEA, DeepSite, DESSO



Y. Zeng, M. Gong, M. Lin, D. Gao and Y. Zhang, "A Review About Transcription Factor Binding Sites Prediction Based on Deep Learning," in *IEEE Access*, vol. 8, pp. 219256-219274, 2020

Results

- **2. DNABERT-TF accurately identifies TFBS**

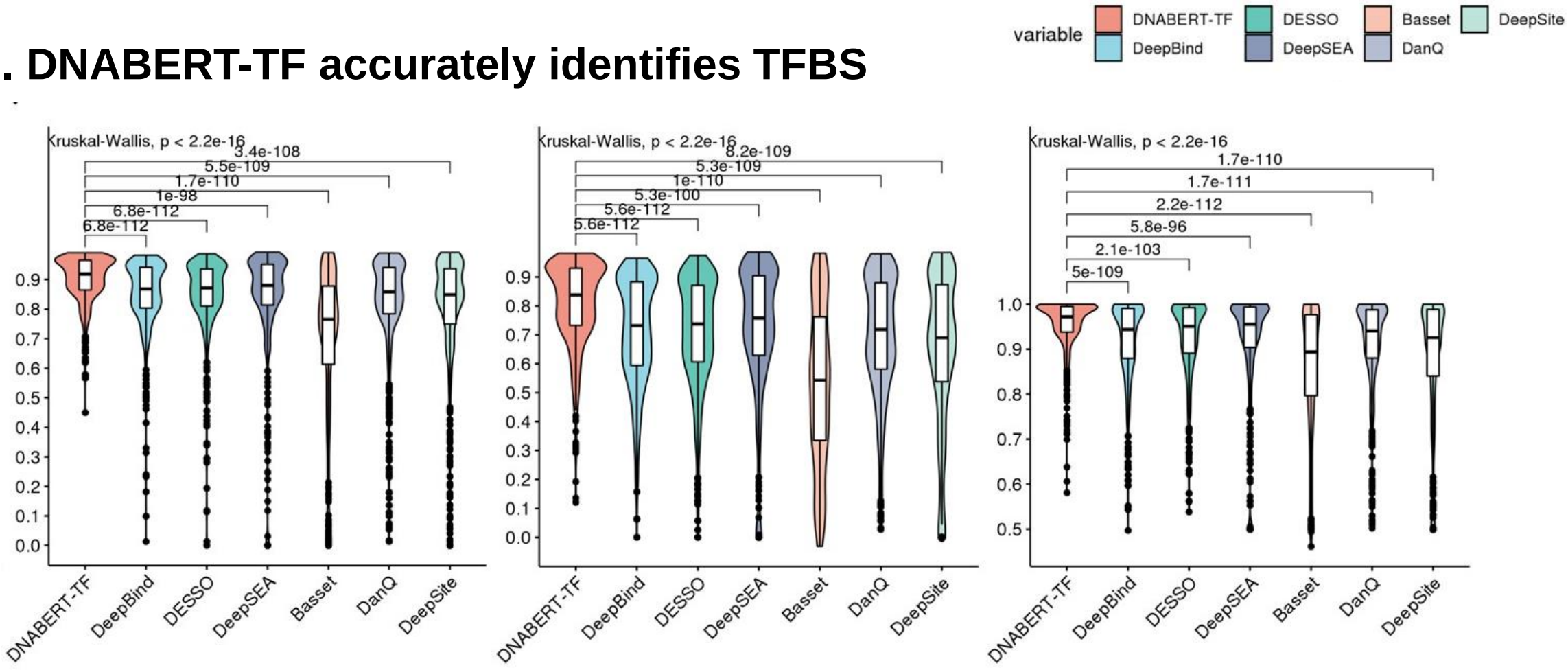


Accuracy **Precision** **Recall**

→ DNABERT accurately identifies TFBSs

Results

- **2. DNABERT-TF accurately identifies TFBS**

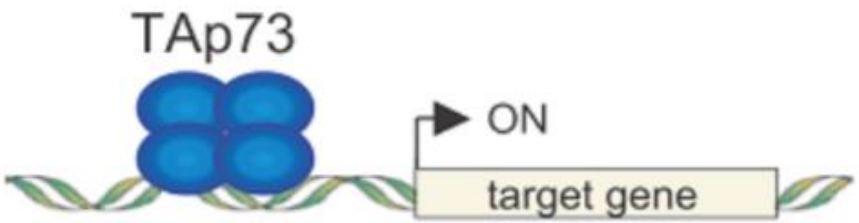


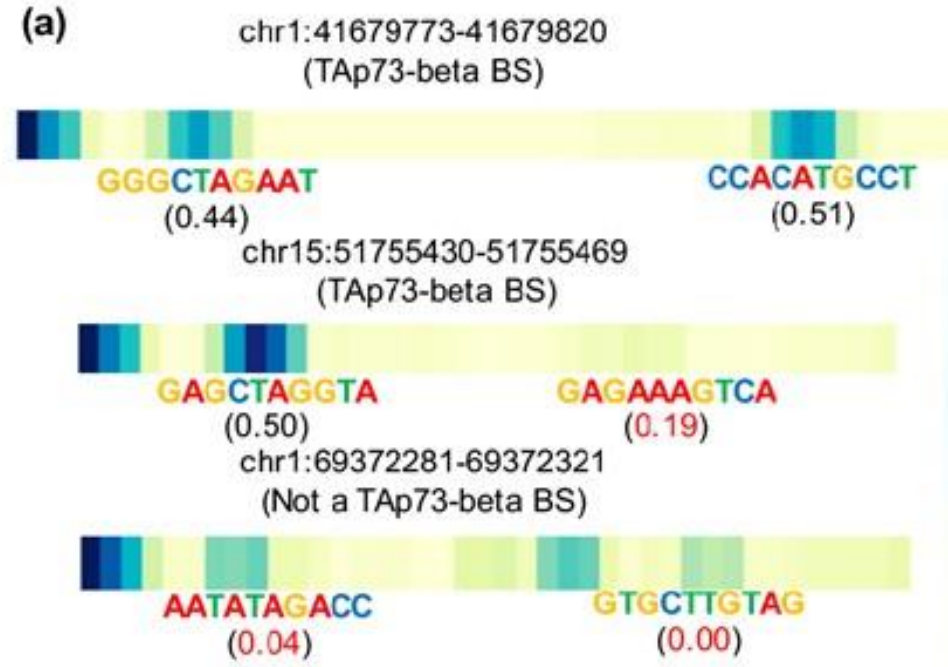
F1 **MCC** **AUC**

→ DNABERT accurately identifies TFBSs

Results

- **3. Visualization of important regions, contexts and sequence motifs**
  - DNABERT-viz module
    - Since it uses attention mechanism, we can directly interpret the scores
    - ex) Attention maps
      - 2 ChIP-seq-validated TAp73-beta binding sites (top, middle)
      - Non-binding site (bottom)





Cam, Hakan. (2006). The role of p53 family members in myogenic differentiation and rhabdomyosarcoma development.

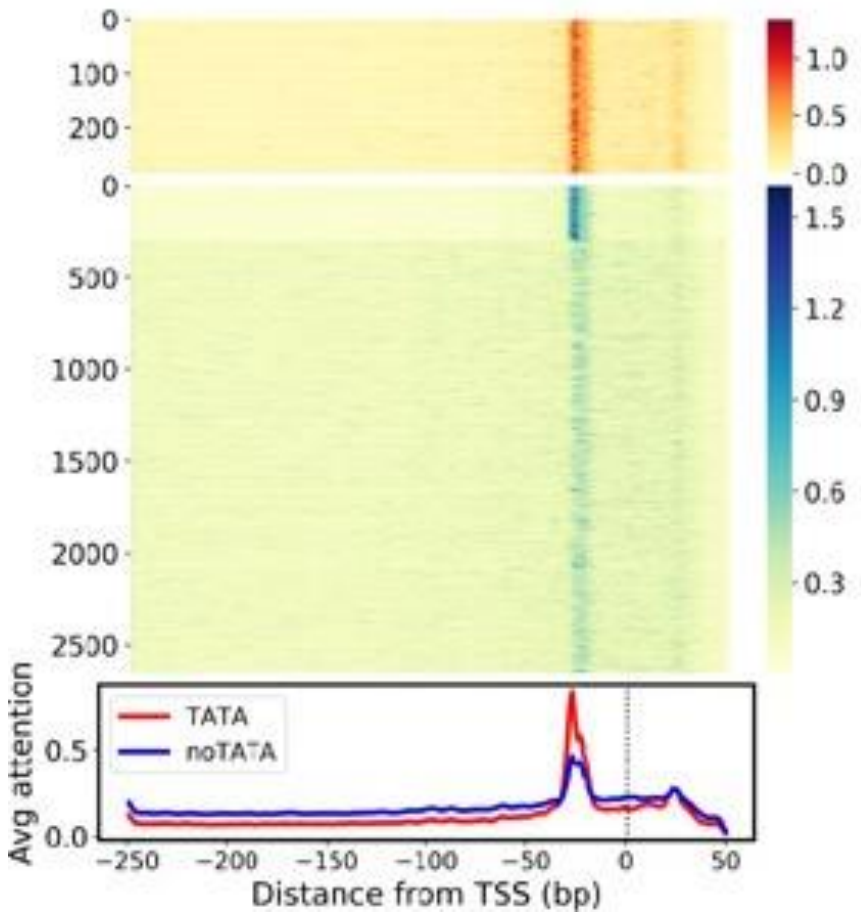
Results

- **3. Visualization of important regions, contexts and sequence motifs**
  - DNABERT-viz module
    - ex) Attention landscape

      (in Prom-300 test set)
      - TATA promoters (top)

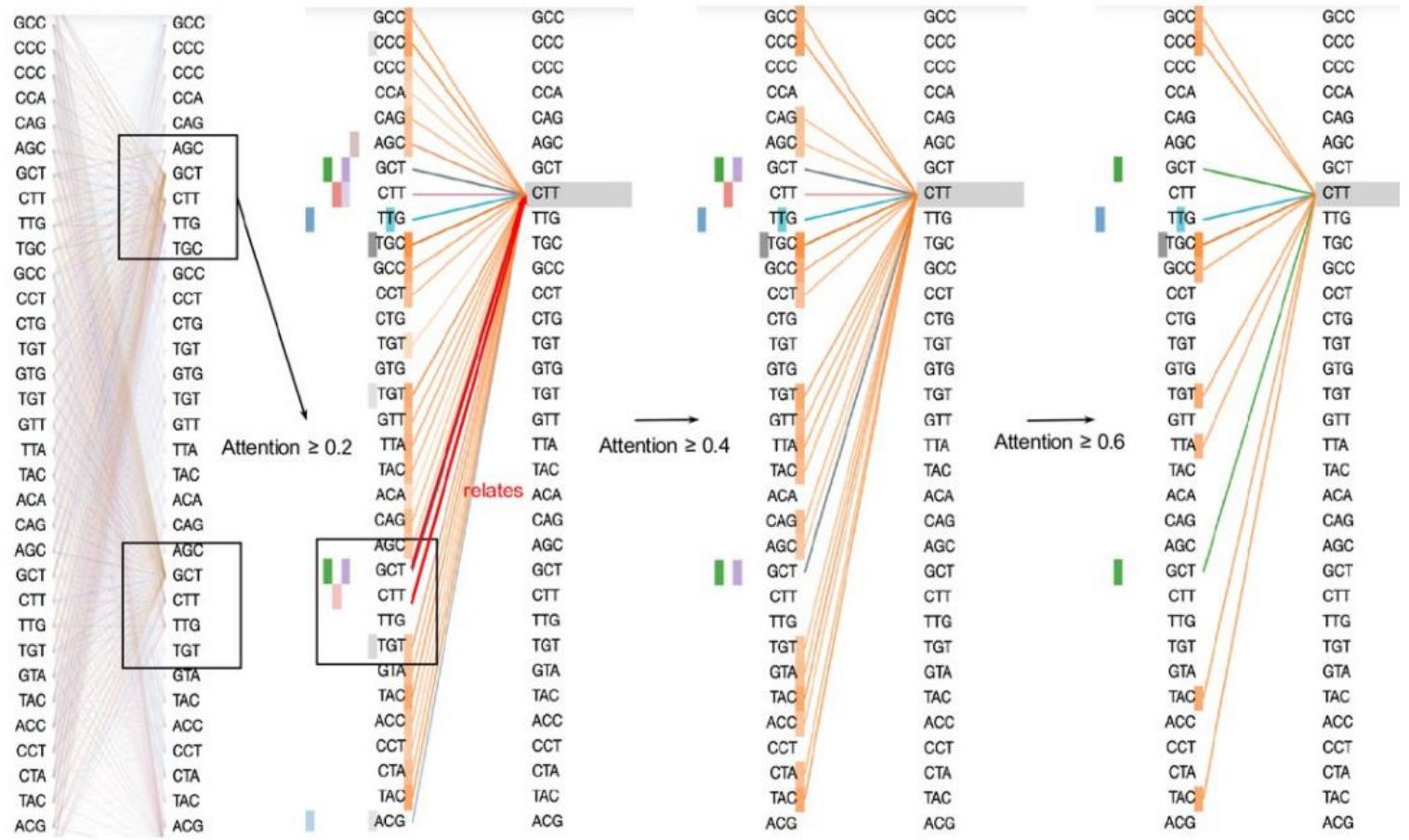
        → DNABERT consistently put high attention upon -20 to -30 bp region upstream of TSS where TATA box is located
      - noTATA promoters(bottom)

        → Majority of non-TATA promoters show a more scattered attention patterm



Results

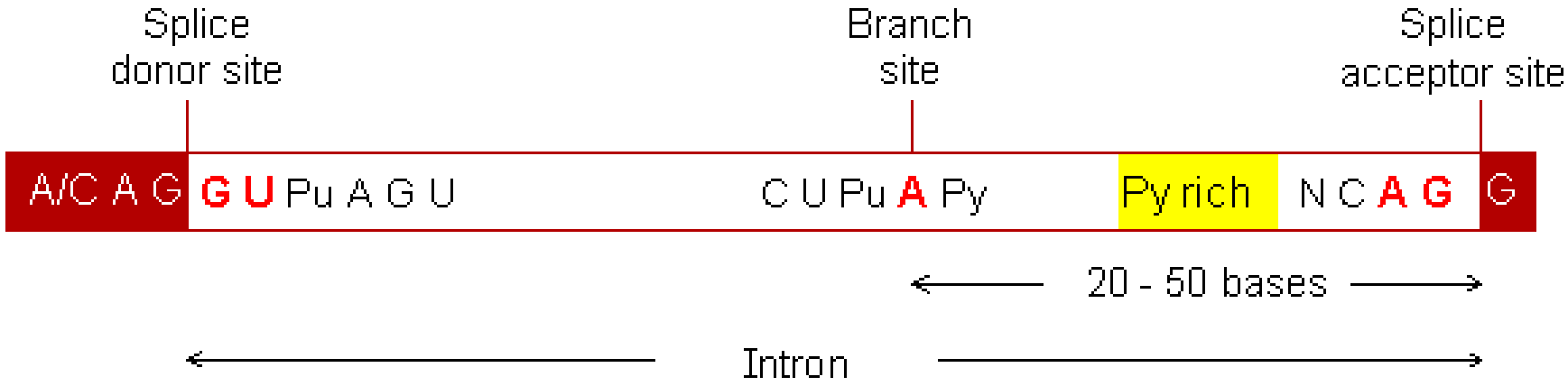
- **3. Visualization of important regions, contexts and sequence motifs**
  - DNABERT-viz module
    - Direct visualization of **contextual relationship within any input sequence**
    - ex) Attention-head plots of p53 binding site



Sentence-level self-attention

Results

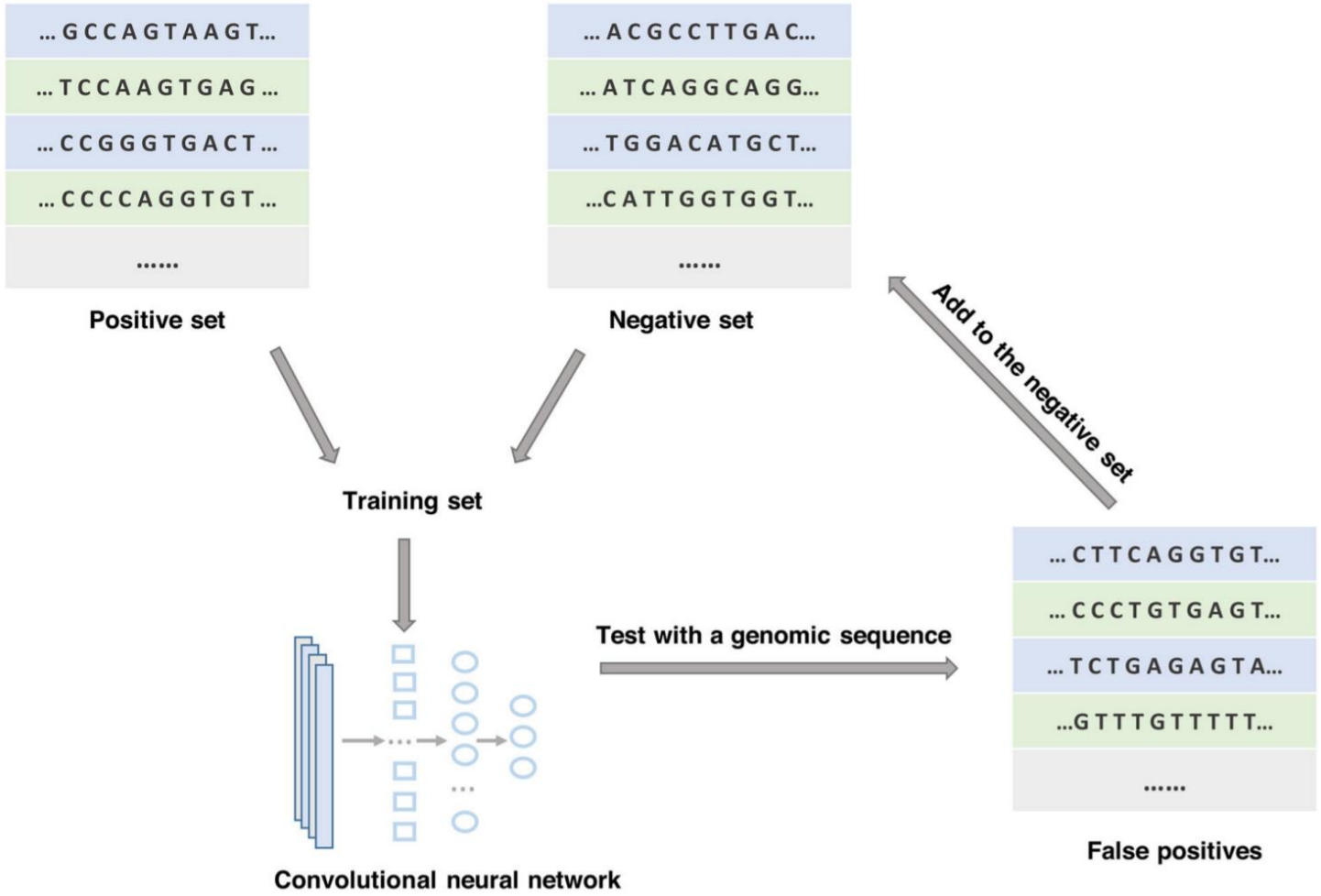
- **4. DNABERT-Splice accurately recognized canonical and non-canonical splice sites**
  - Fine-tuned DNABERT with GRCh38 FASTA file from Ensemble release 99
    - Positive sequences : Extracted from 400-bp long sequences around the donor and acceptor sites of randomly selected exons



https://www.web-books.com/MoBio/Free/Ch5A4.htm

Results

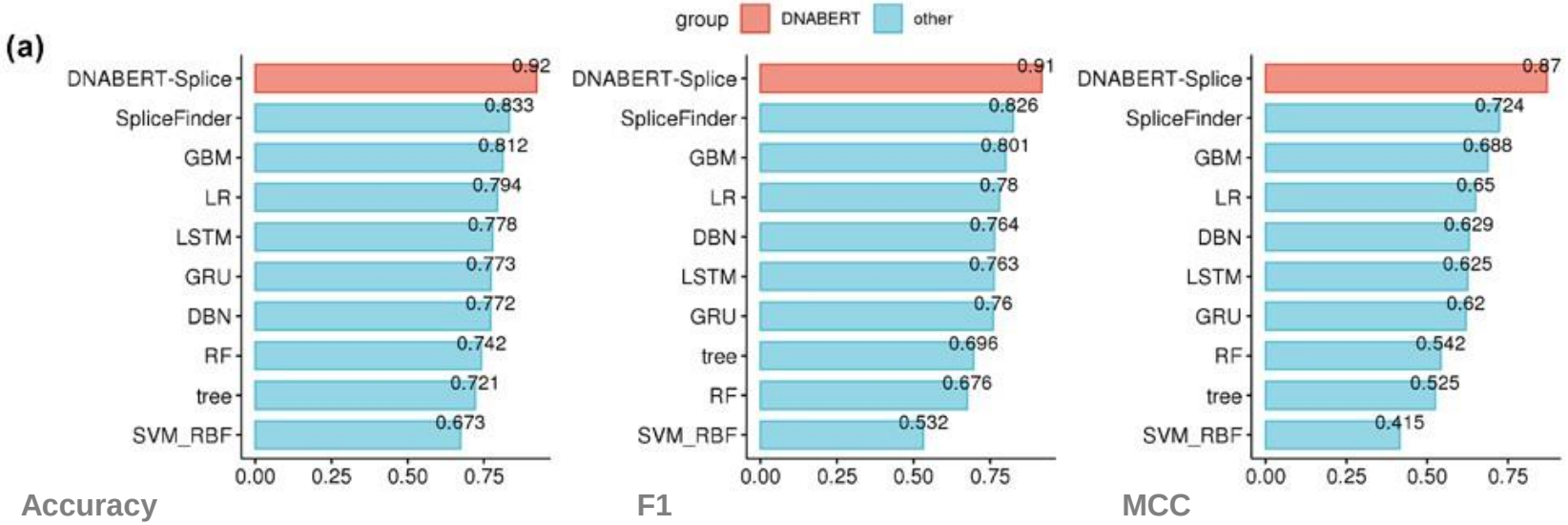
- **4. DNABERT-Splice accurately recognized canonical and non-canonical splice sites**
  - To detect non-canonical splice sites without GT or AG dimers
    : Iterative data augmentation
    - Repeatedly built new models to predict on sliding-window-based scans



Wang, Ruohan et al. “SpliceFinder: ab initio prediction of splice sites using convolutional neural network.” *BMC Bioinformatics* 20 (2019): n. pag.

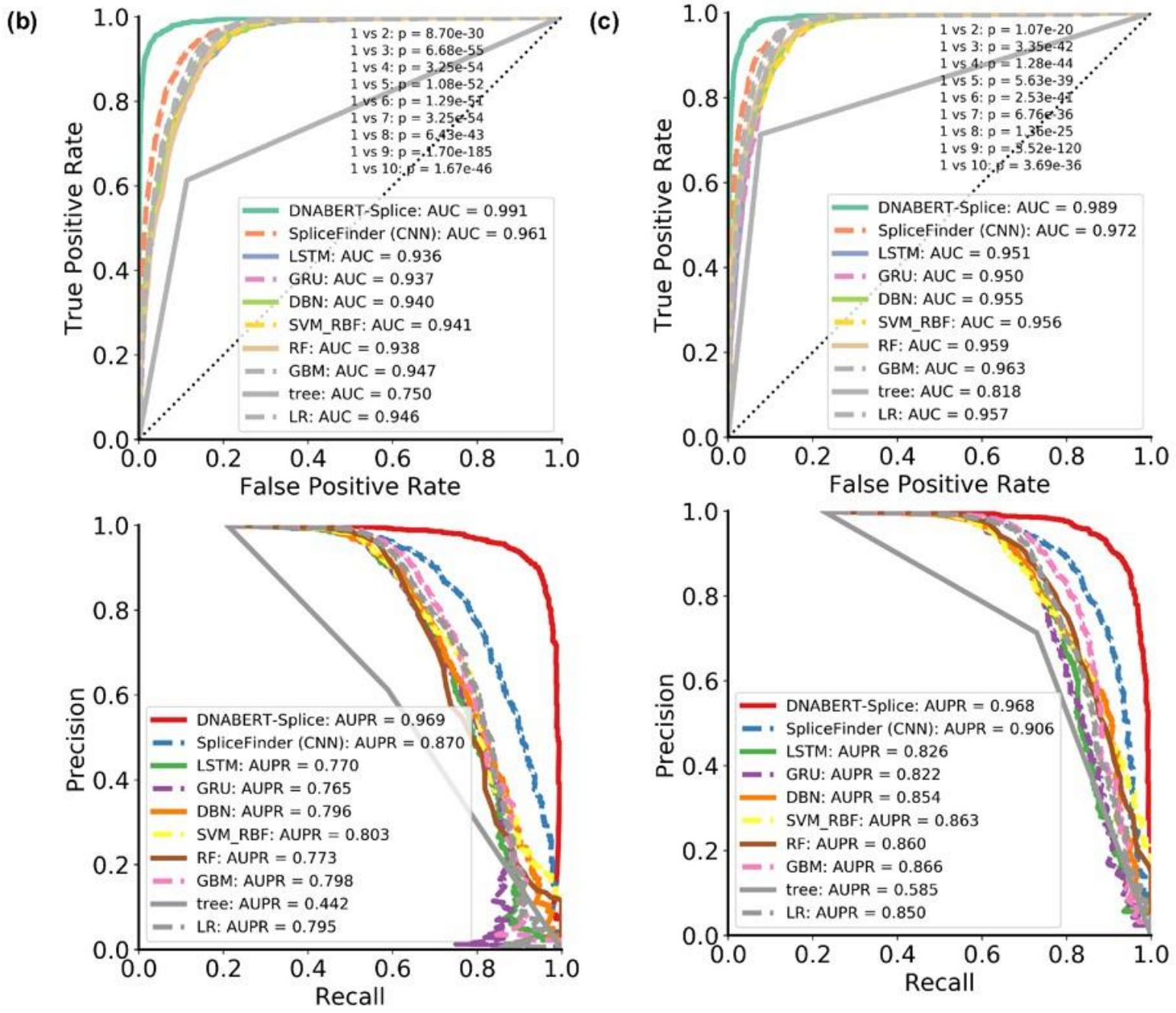
Results

- **4. DNABERT-Splice accurately recognized canonical and non-canonical splice sites**
  - Fine-tuned DNABERT with GRCh38 FASTA file from Ensemble release 99
    - DNABERT-splice showed globally high attention upon intronic regions, highlighting the presence and functional importance of various intronic splicing enhancers (ISEs) and silencers (ISSs) acting as CREs for splicing



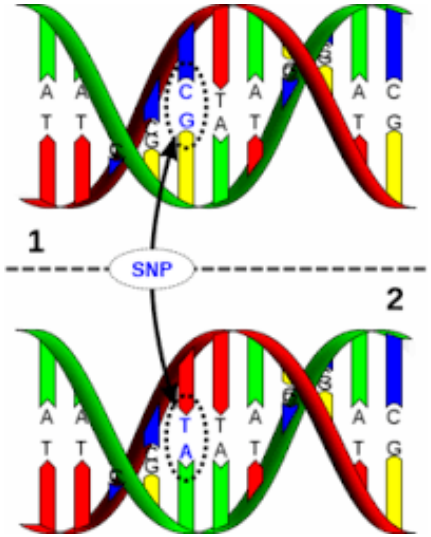
Results

- DNABERT significantly outperforms other models in finding splice sites
  - (b) : Splice donor dataset
  - (c) : Splice acceptor dataset



Results

- **5. Identifying functional genetic variants with DNABERT**
  - Fine-tuned DNABERT with dbSNP release 153 : 700 million functional variants
    - Selected only those variants located inside DNABERT-predicted high-attention regions and repeated the predictions, using sequence with **altered alleles** ($p(s) \rightarrow p(s')$)
      - Score change = $\Delta p = \big(p(s') - p(s)\big) \max(p(s'), p(s))$
      - Log odds ratio = $\log_2 OR = \log \frac{p(s)}{1-p(s)} - \log \frac{p(s')}{1-p(s')}$
    - Candidate variants resulting in significant changes in prediction probability → queried in ClinVar, GRASP, and NHGRI-EBI GWAS catalog
    - In Prom-300 dataset
      - 24.7% and 34.1% of dbSNP common variants we identified using TATA and non-TATA promoters are present in at least one of the three databases

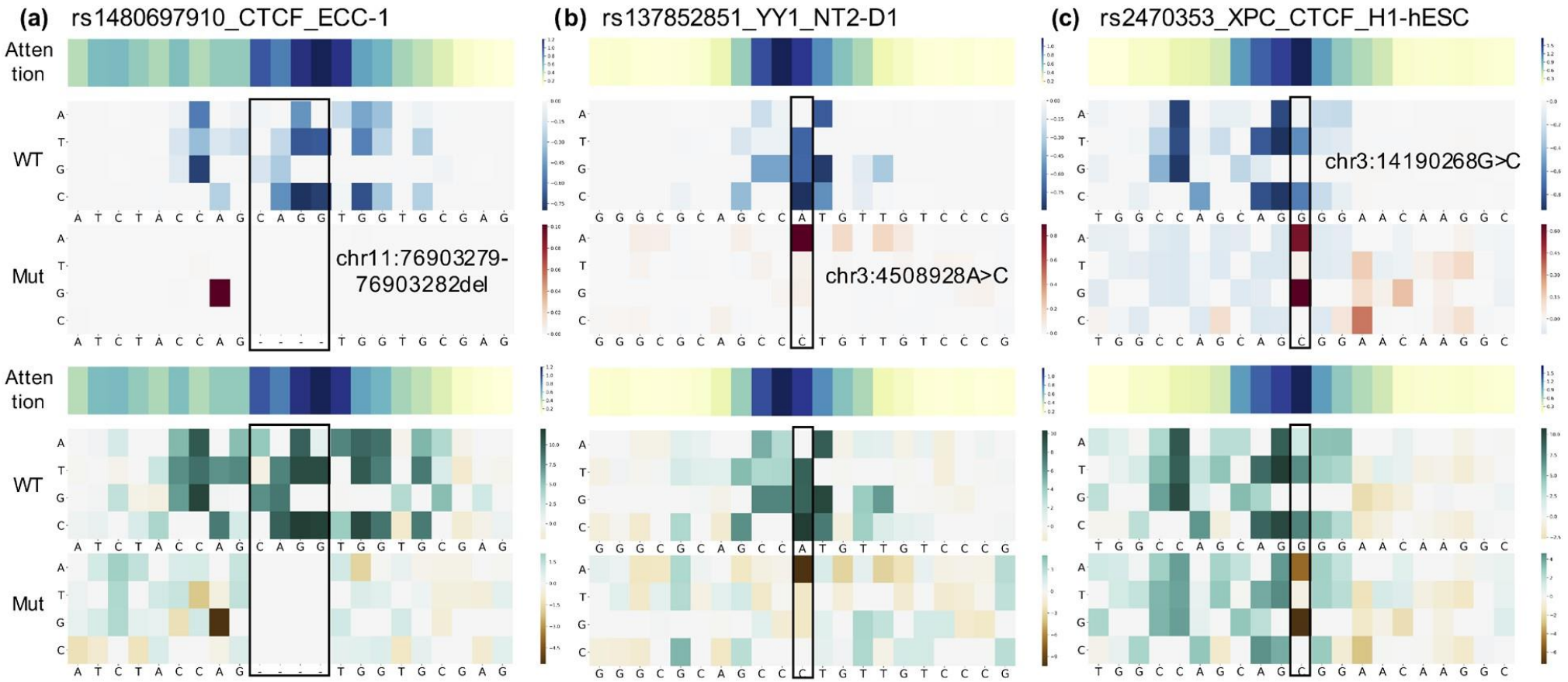


https://isogg.org/wiki/Single-nucleotide_polymorphism

Results

(a) : Rare, pathogenic 4bp deletion
Cause Usher syndrome
(b) : Rare SNV
Multiple sulfates deficiency
(c) : Common risk variant of
pancreatic cancer

- **5. Identifying functional genetic variants with DNABERT**



→ DNABERT identifies functional genetic variants
DNABERT consistently shows highest attention at/around the variants of interest

Discussion

- DNABERT achieved superior performance across various downstream DNA sequence prediction tasks by largely surpassing existing tools
- DNABERT tackles the problem of sequence specificity prediction with a “top-down” approach by first developing general understanding of DNA language via self-supervised pre-training and then applying it to specific tasks, in contrast to the traditional “bottom-up” approach using task-specific data
- Effectively learn from DNA context with great flexibility adapting to multiple situations, and enhanced performance with limited data

Google Colab

Thank You!