自然语言处理技术与大语言模型

Outline

- AI & NLP
- Tokenize
- Word Embedding
- Transformer
- BERT
- GPT

AI & NLP

- What is NLP？

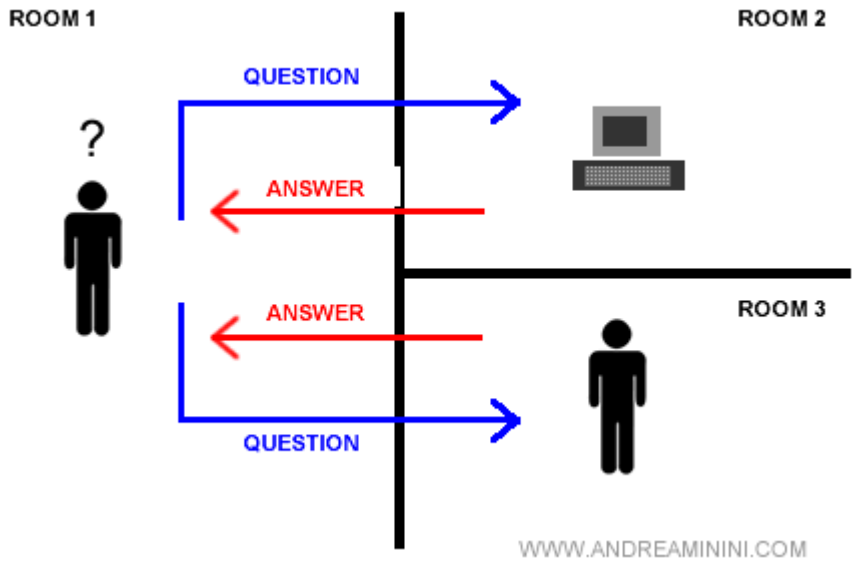
  Definition：

  “NLP（Natural Language Processing） is a subfield of AI that focuses on the interaction between computers and humans through natural language. It enables computers to understand, interpret, and generate human language in a meaningful way.”

——Jurafsky & Martin, *Speech and Language Processing*

AI & NLP

- Turing Test



https://www.andreaminini.net/computer-science/artificial-intelligence/turing-test

AI & NLP

- Text Classification
- Named Entity Recognition (NER)
- Machine Translation
- Speech Recognition and Synthesis
- Sentiment Analysis
- ……

Outline

- AI & NLP
- Tokenize
- Word Embedding
- Transformer
- BERT
- GPT

Tokenize

“I felt uneasy about the uncertain future.”

我们该如何处理这样一个文本？？

- 字符级（char）

I felt uneasy about the uncertain future. ➡ I f e l t u n e a s y a b o u t t h e u n c e r…

- 单词级（word）

I felt uneasy about the uncertain future. ➡ I felt uneasy about the uncertain future .

Question：easy、uneasy、certain、uncertain……似乎存在一些语法规则？

- ★子词级（sub word）

I felt uneasy about the uncertain future. ➡ I felt un easy about the un certain future .

Tokenize

- BPE算法

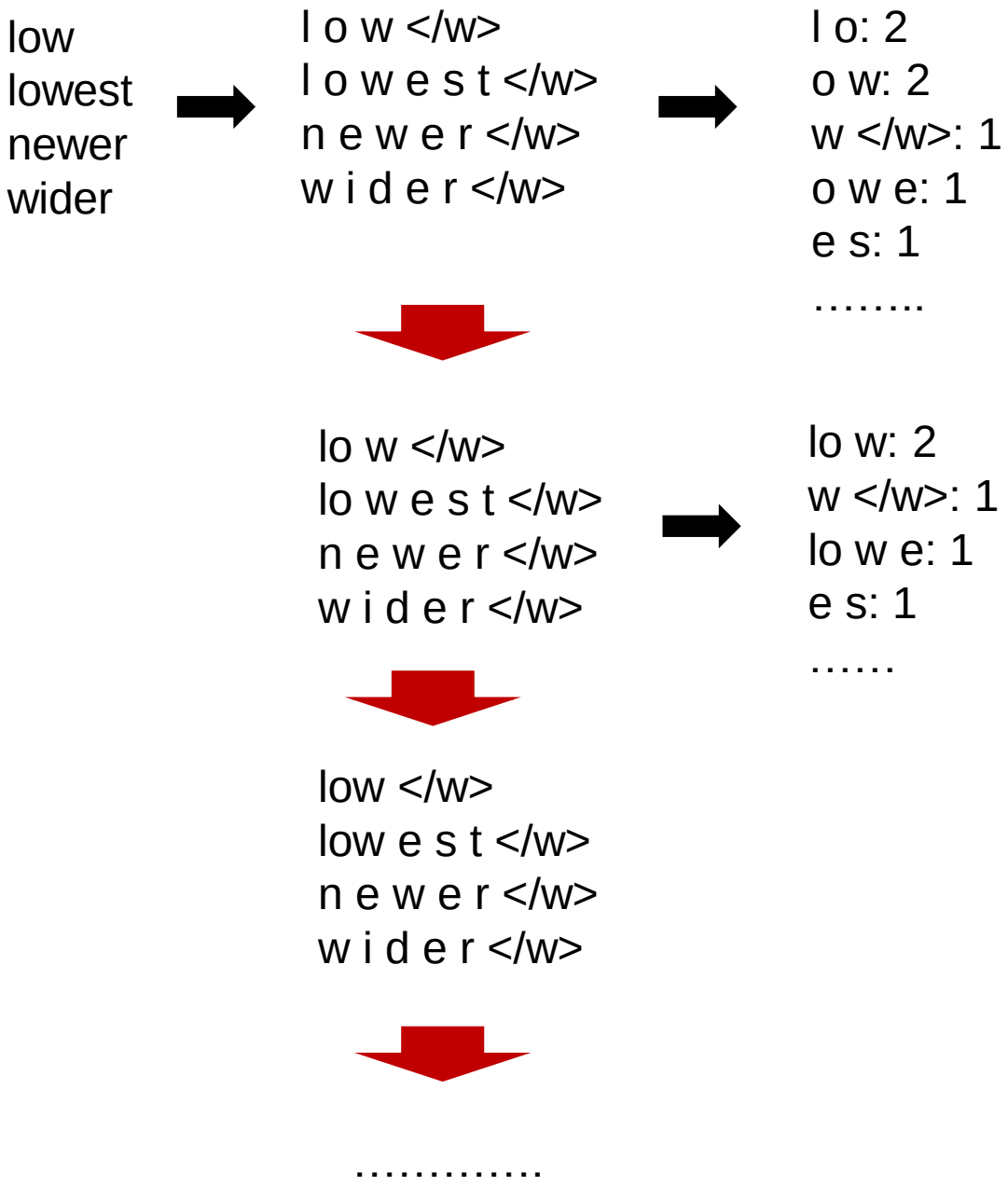
```
2. While N > L:
      2.1 统计字符对频率:
          pairs_freq = {}
          For word in corpus:
              For each adjacent pair of chars in word:
                  pairs_freq[pair] += 1

      2.2 找出频率最高的字符对:
          best_pair = argmax(pairs_freq)

      2.3 合并该字符对:
          For word in corpus:
              Replace best_pair with new symbol in word

      2.4 更新词汇表:
          Add best_pair to vocabulary

3. 输出分词结果 vocabulary
```



Outline

- AI & NLP
- Tokenize
- Word Embedding
- Transformer
- BERT
- GPT

Word Embedding

语料：I felt un easy about the un certain future .

词表：I felt un easy about the certain future .

这样子计算机就能处理了吗？？

- ASCII 字符

66	B 大写字母 B	67	C 大写字母 C	68	D 大写字母 D
69	E 大写字母 E	70	F 大写字母 F	71	G 大写字母 G
72	H 大写字母 H	73	I 大写字母 I	74	J 大写字母 J
75	K 大写字母 K	76	L 大写字母 L	77	M 大写字母 M

https://www.ibm.com/docs/zh/sdse/6.4.0?topic=administering-ascii-characters-from-33-126

Question：该怎么让文字变得可计算？？

Answer：　文字向量化！！

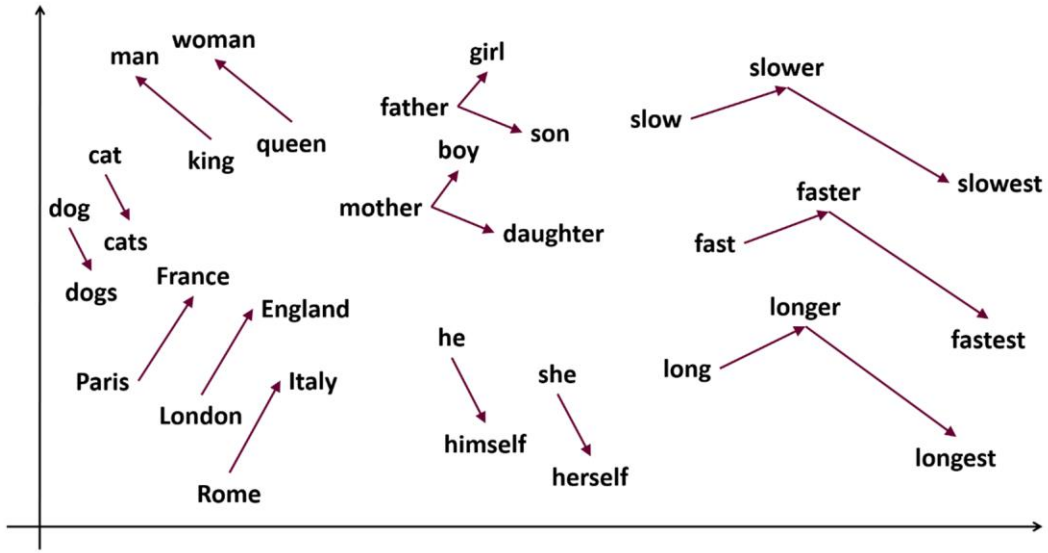
Word Embedding

- One hot encoding

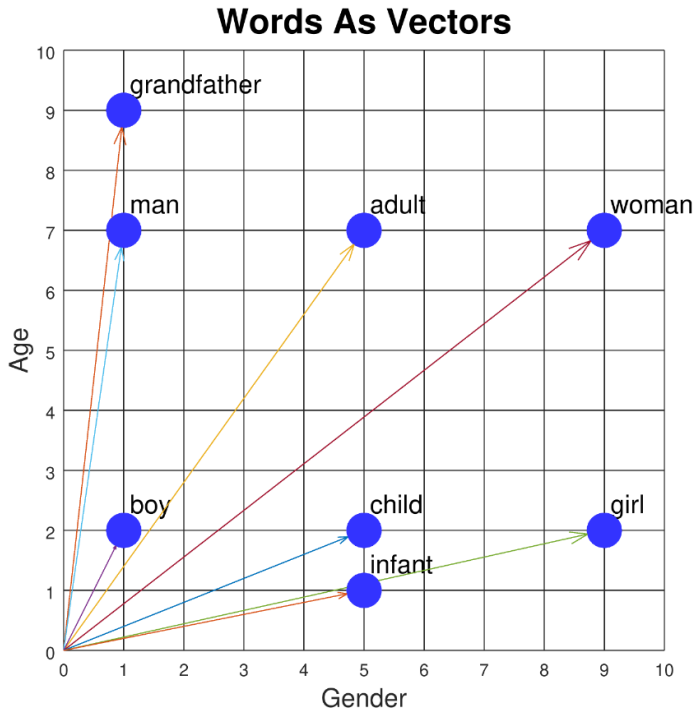
I felt un easy about the certain future .

word								
I	1	0	0	0	0	0	0	0
felt	0	1	0	0	0	0	0	0
un	0	0	1	0	0	0	0	0
easy	0	0	0	1	0	0	0	0
about	0	0	0	0	1	0	0	0
the	0	0	0	0	0	1	0	0
certain	0	0	0	0	0	0	1	0
future	0	0	0	0	0	0	0	1

• Word Embedding



Word Embedding
(https://samyzaf.com/ML/nlp/nlp.html)



Word Embedding
(https://iq.opengenus.org/one-hot-encoding-in-tensorflow/)

Word Embedding

- Word2Vec

- CBOW

给定一组上下文词来预测中心词，最小化预测与真实中心词之间的误差

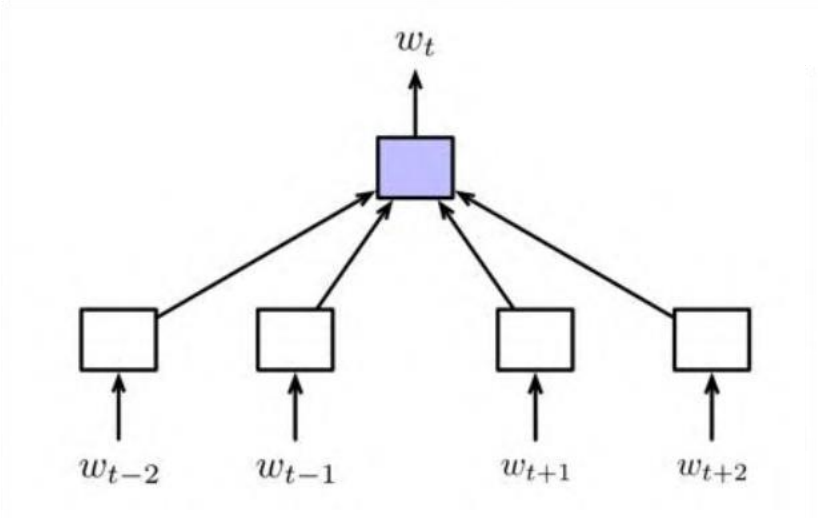
- Skip-Gram

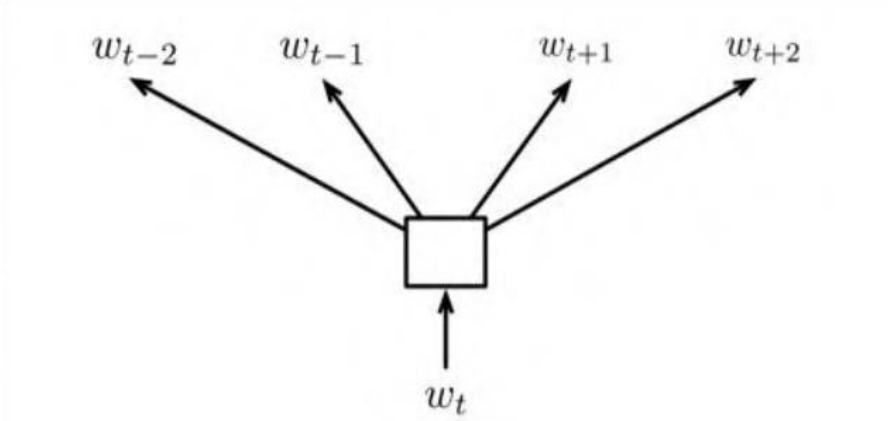
给定一个词，来预测该词的上下文内容，最小化预测误差

Output

Hidden layer

Input



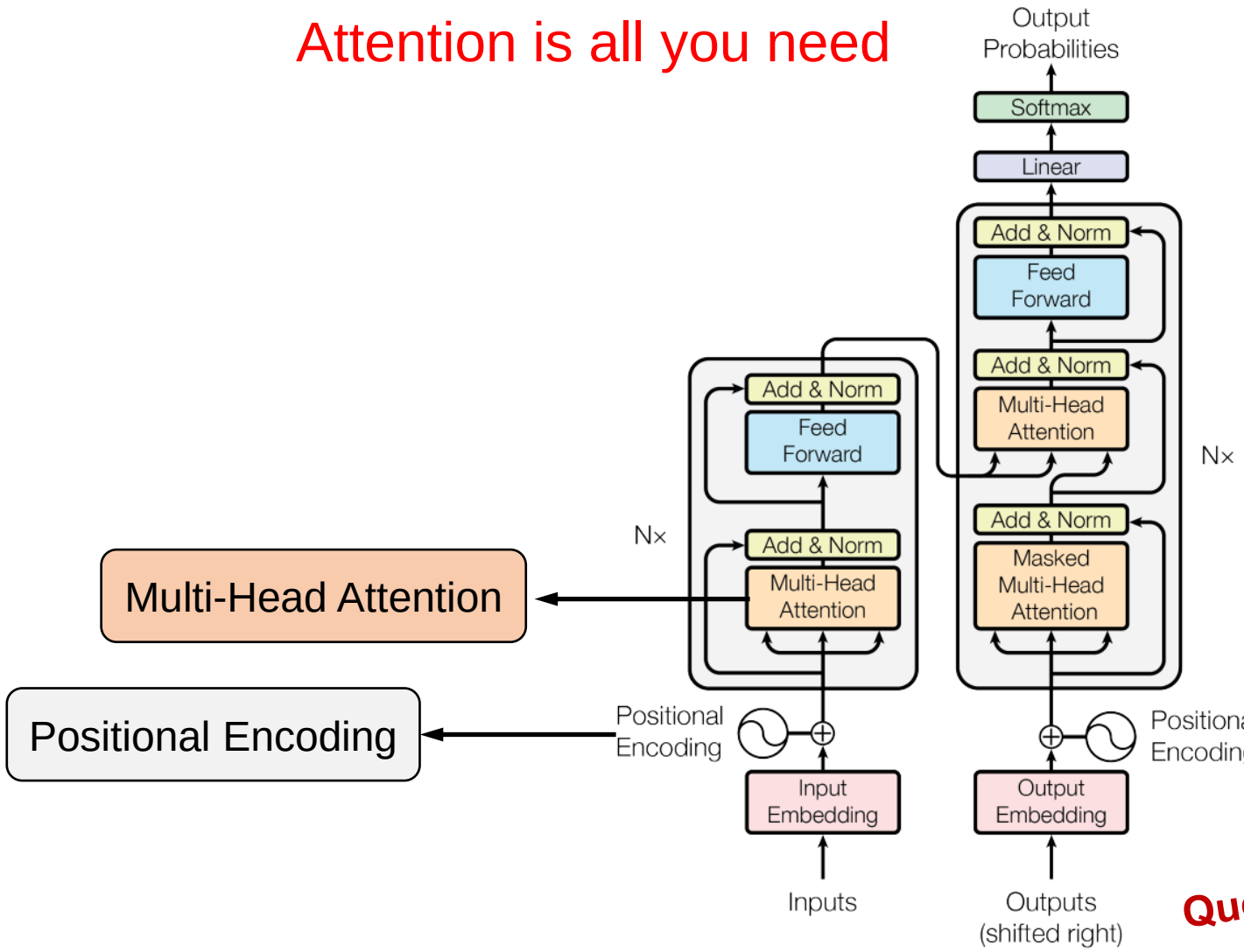


Outline

- AI & NLP
- Tokenize
- Word Embedding
- Transformer
- BERT
- GPT

Transformer

Attention is all you need



Multi-Head Attention

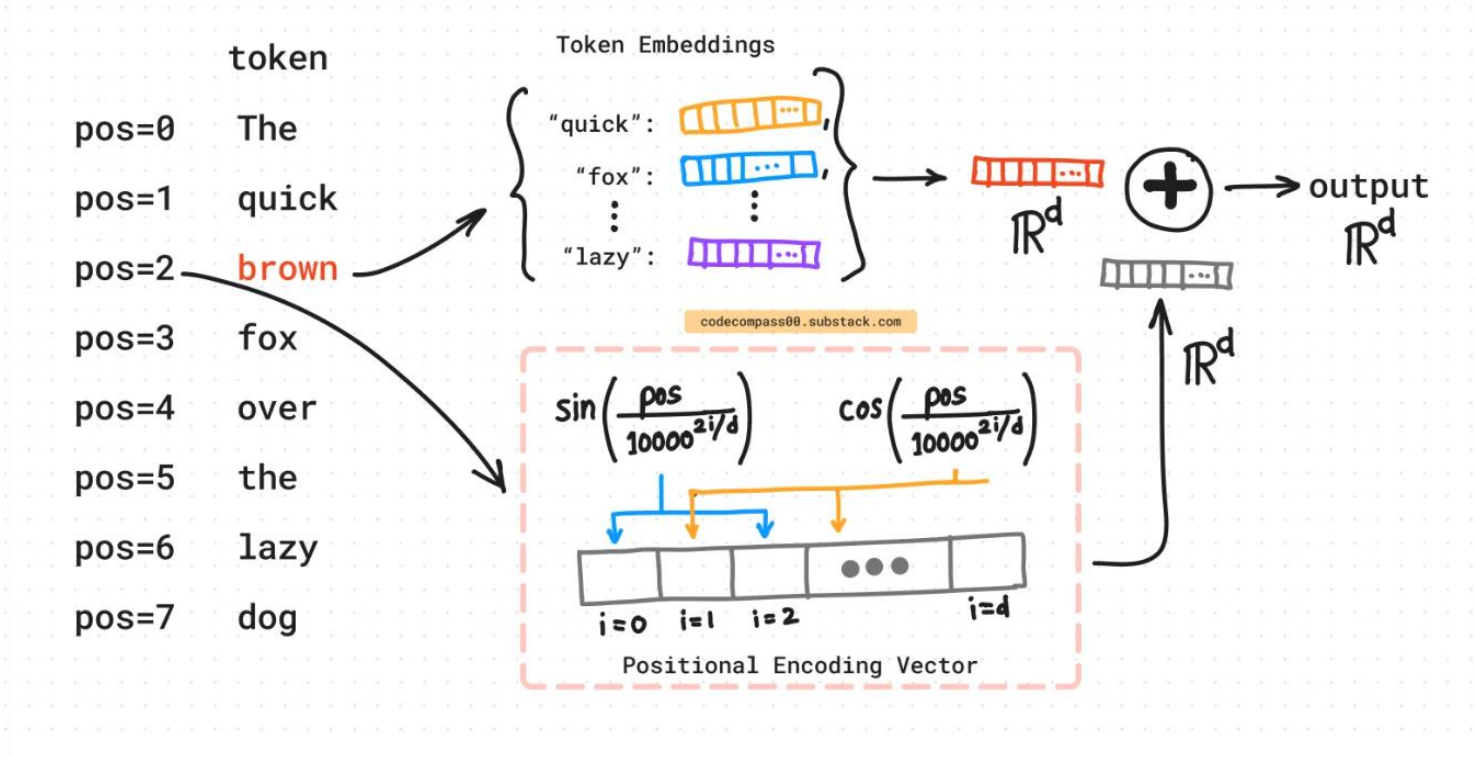
Positional Encoding

Question：输出都给模型了？还训练个啥？

https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Transformer

- Positional Encoding



https://codecompass00.substack.com/p/positional-encoding-transformers

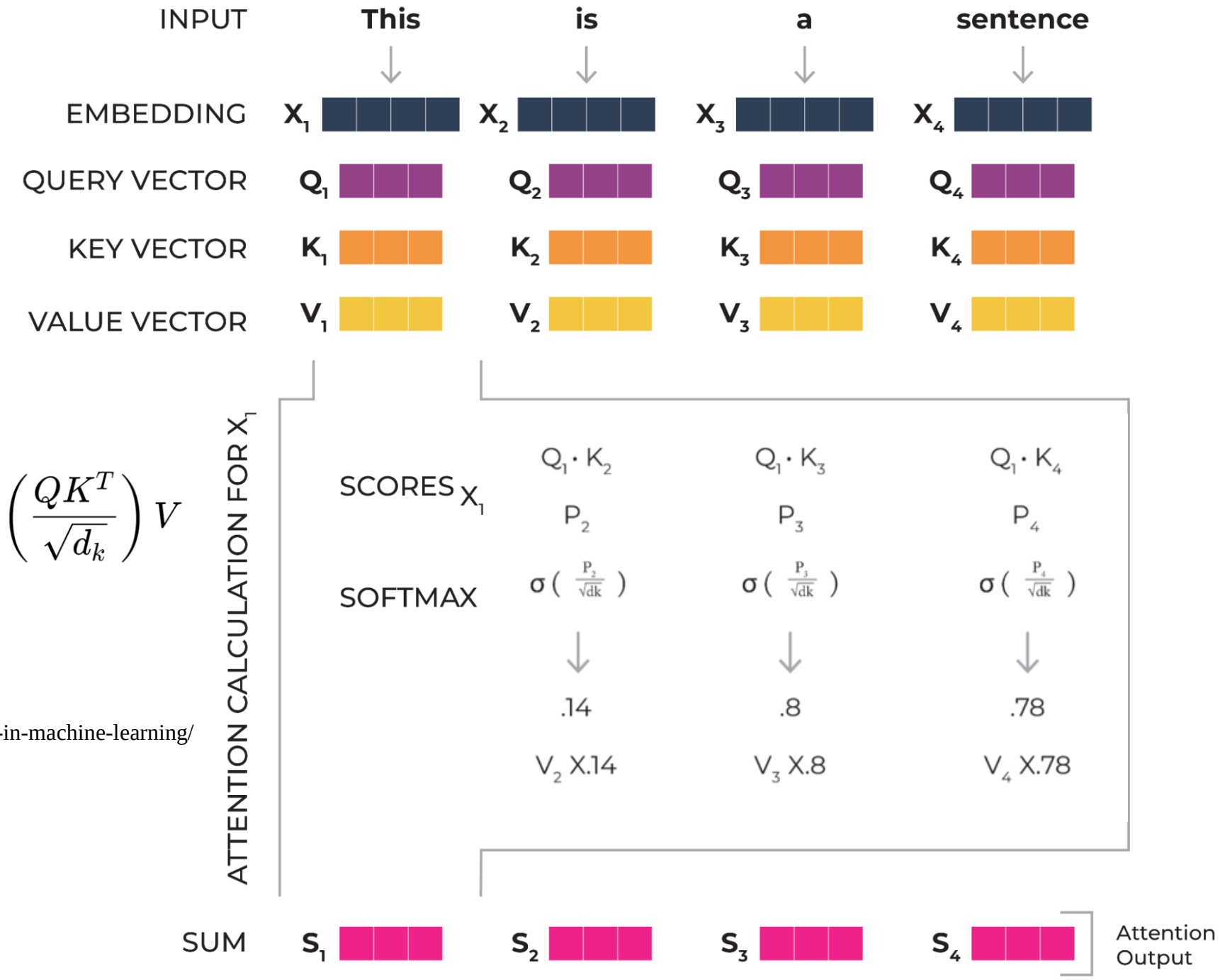
- 不是单一数值，而是包含句子中特定位置信息的d维向量（非常像词向量）。
- 没有整合进模型，而是通过注入词的顺序信息来增强模型输入。

Transformer

- Multi-Head Attention

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

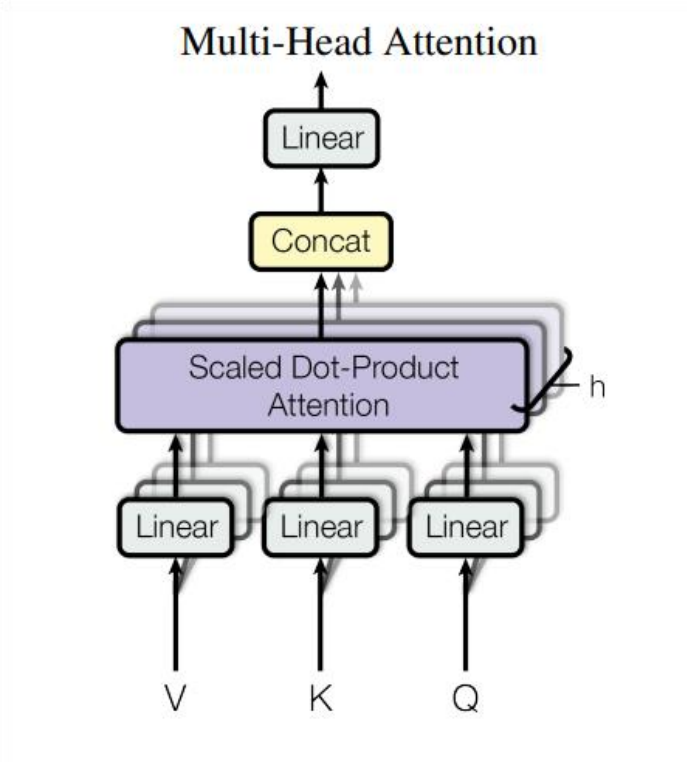
https://arize.com/blog-course/attention-mechanisms-in-machine-learning/



Transformer

- Multi-Head Attention

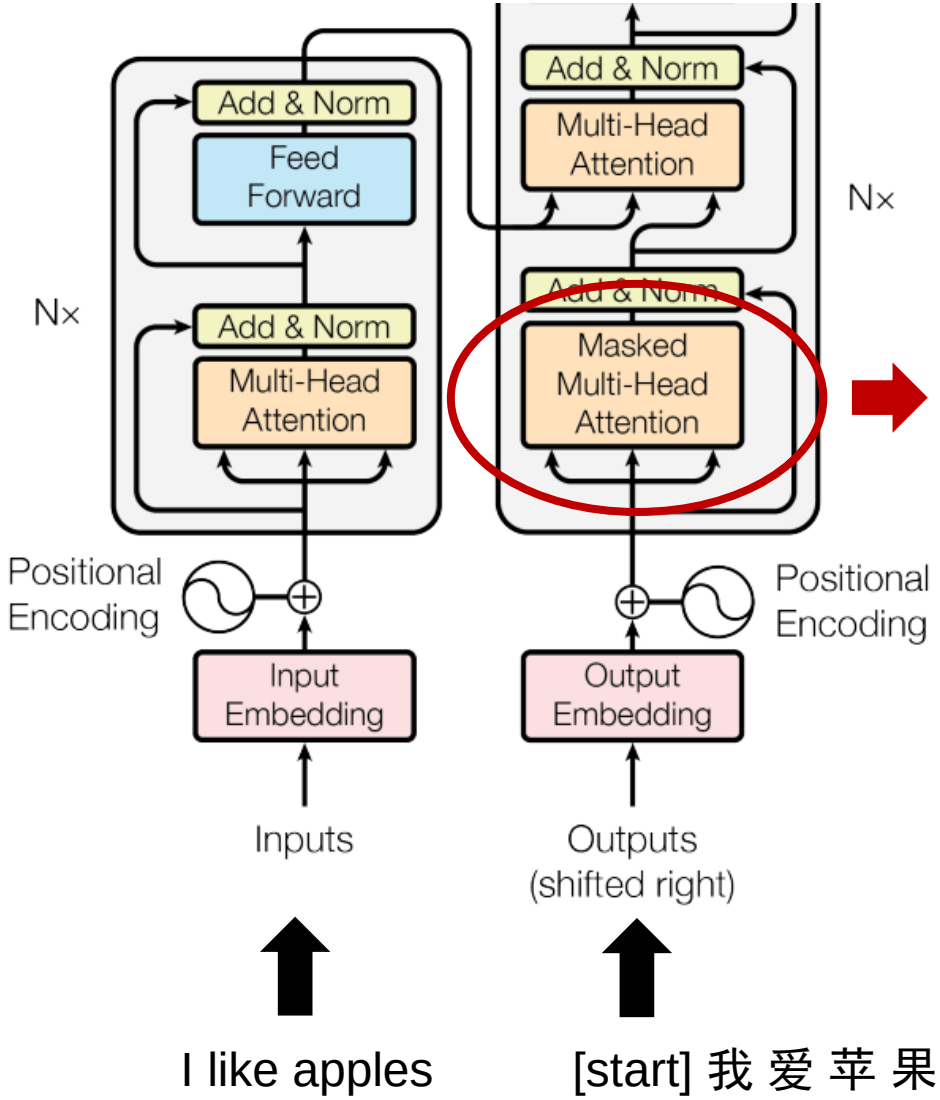
$$\text{MultiHead}(Q, K, V) = Concat(\text{head}_1, ..., \text{head}_h)W^0$$
$$\text{where head} = Attention(QW_i^Q, KW_i^K, VW_i^V)$$



https://proceedings.neurips.cc/

Transformer

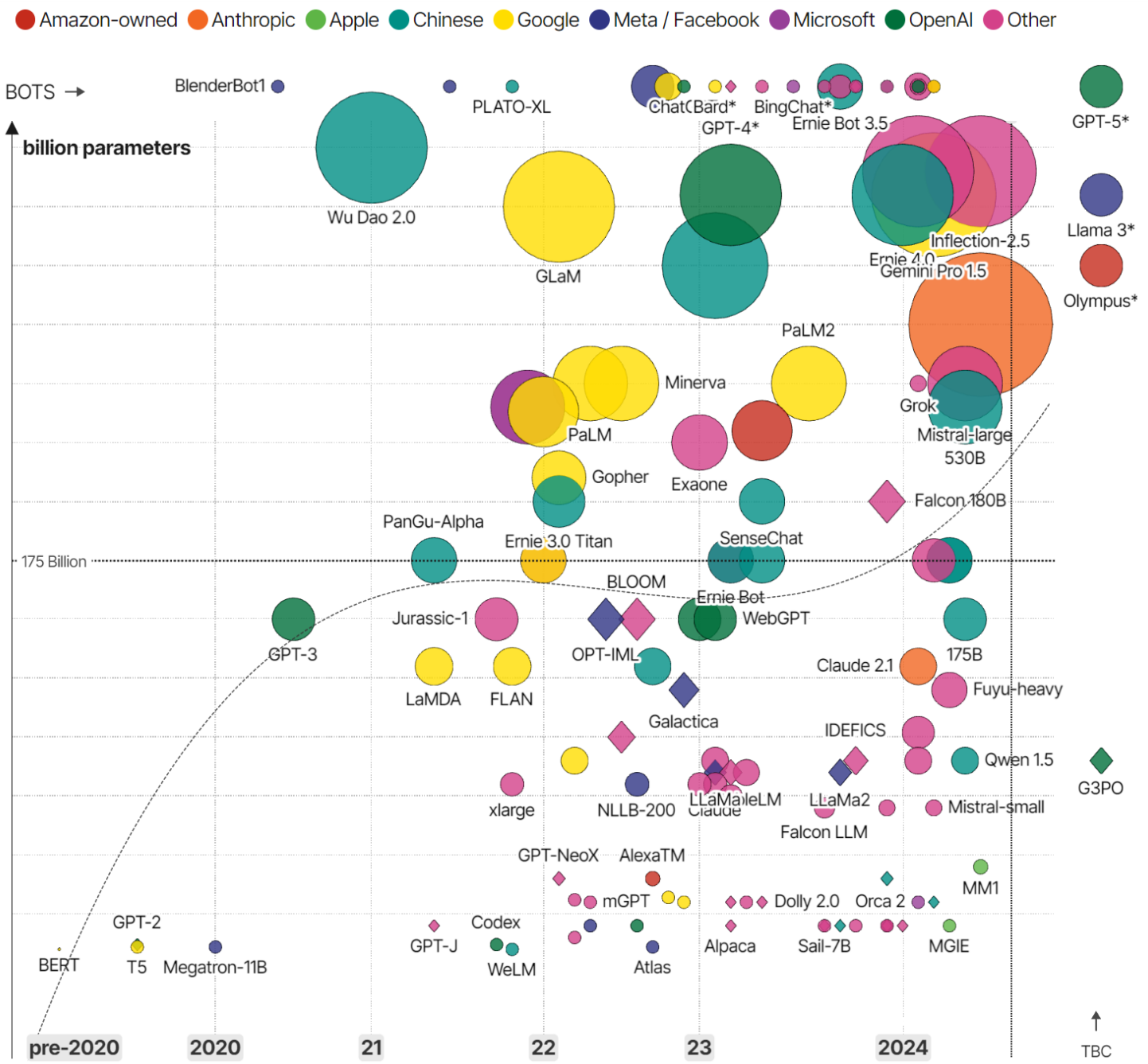
- Masked Multi-Head Attention



注意！！这里红色的字模型看不见！！

[start] 我 爱 苹 果

Transformer



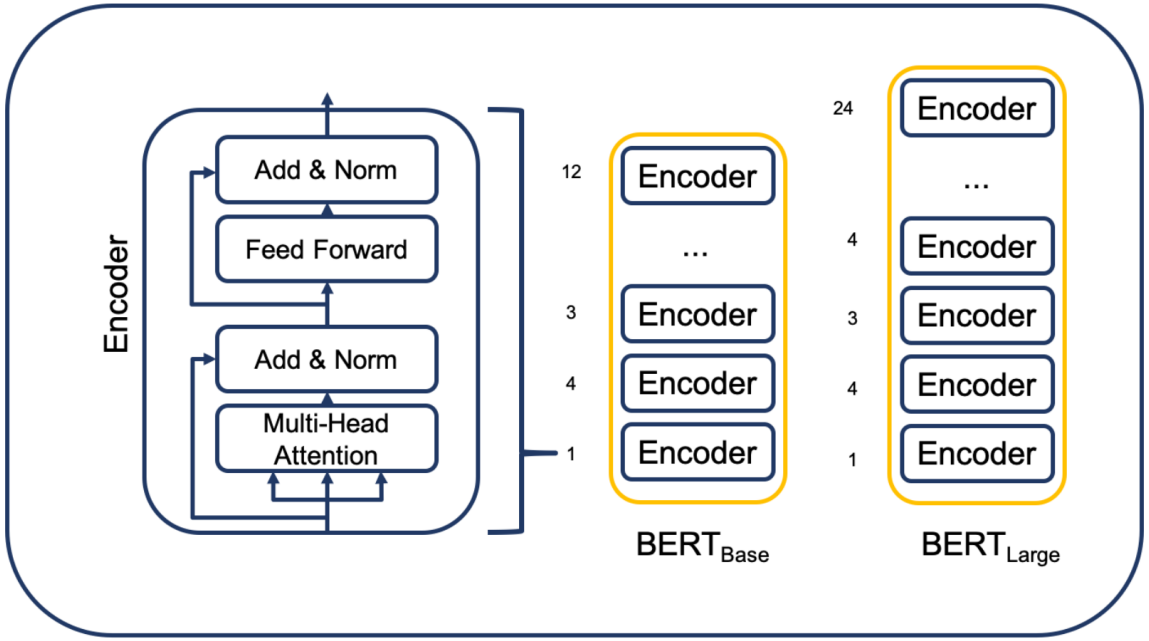
https://informationisbeautiful.net/visualizations/the-rise-of-generative-ai-large-language-models-llms-like-chatgpt/

Outline

- AI & NLP
- Tokenize
- Word Embedding
- Transformer
- BERT
- GPT

BERT

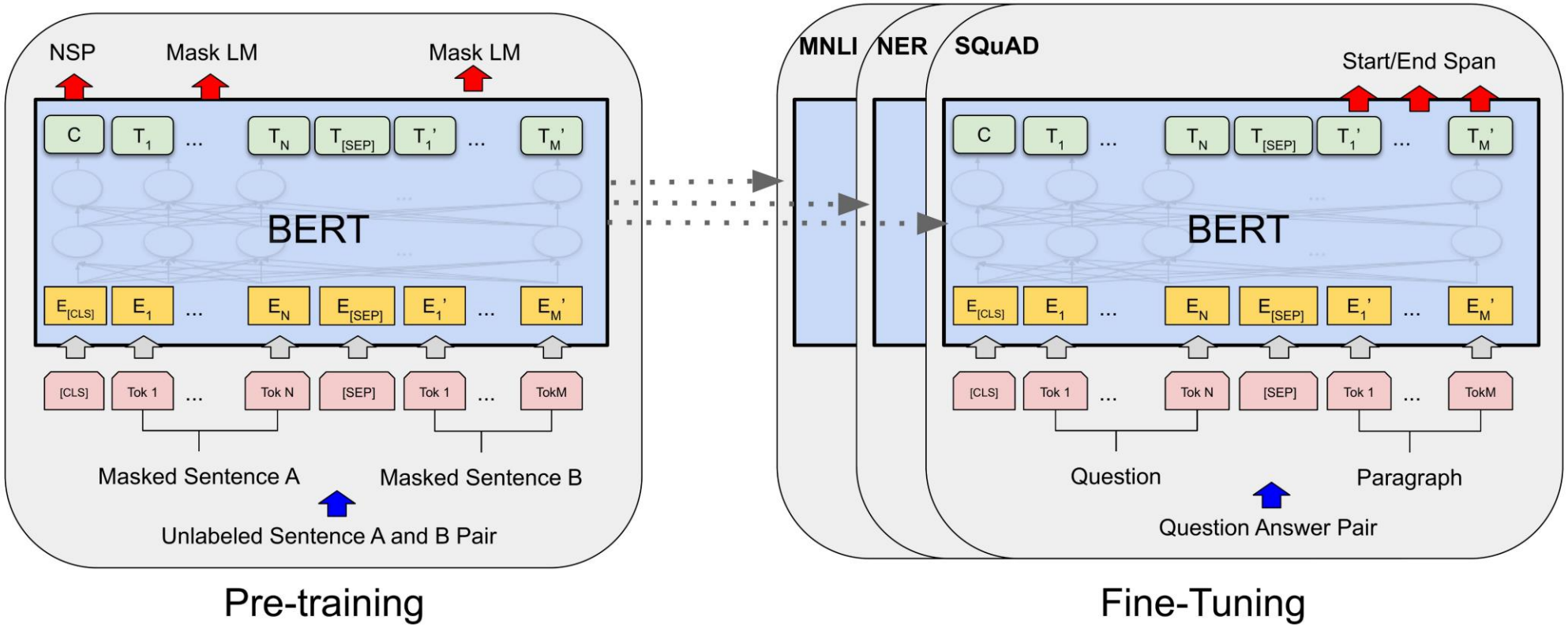
- Encoder only



BERT
(https://humboldt-wi.github.io/blog/research/information_systems_1920/bert_blog_post/)

BERT

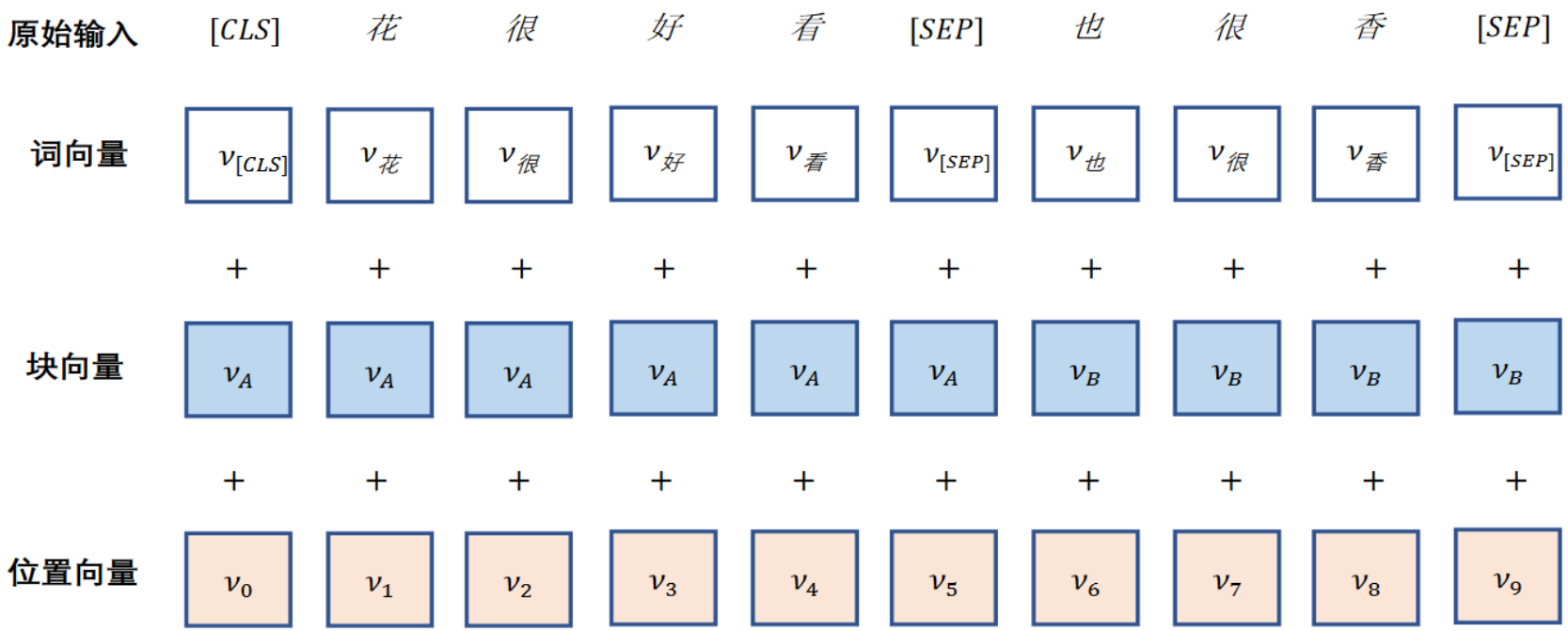
- Pre-training and Fine-Tuning



https://aclanthology.org/N19-1423.pdf

BERT

- BERT input

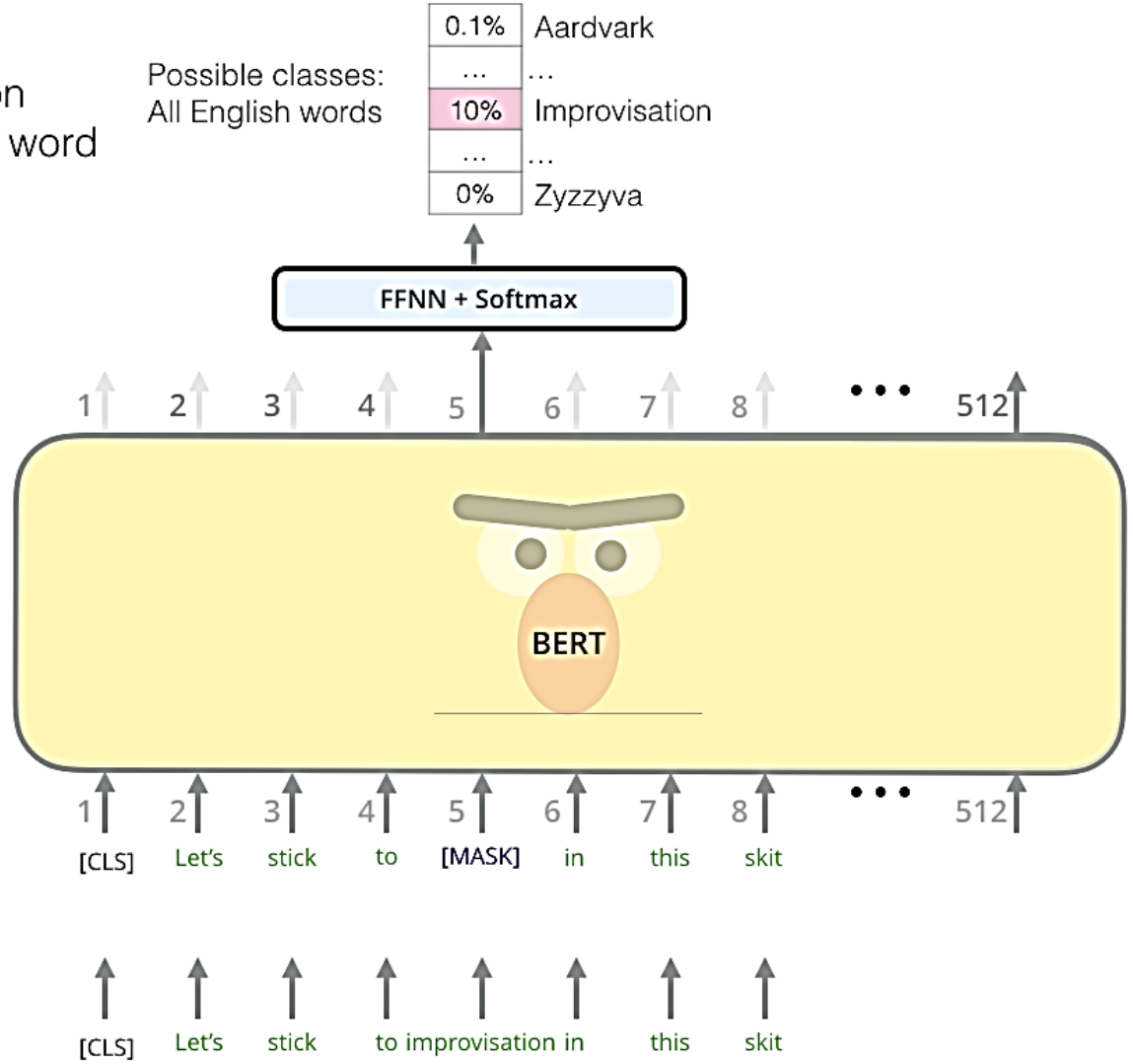


BERT

- Pre-training task

Use the output of the masked word's position to predict the masked word

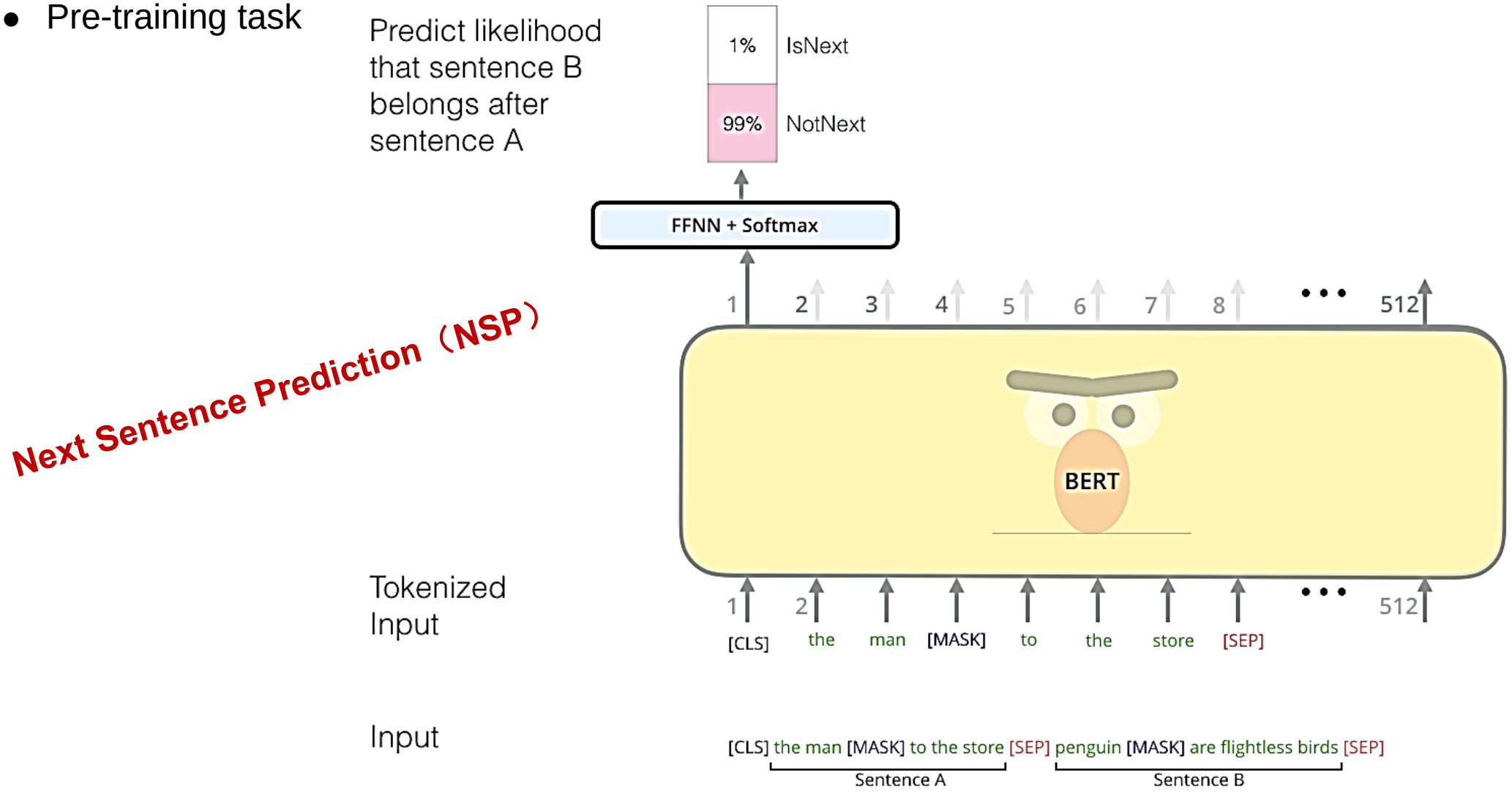
Possible classes: All English words



Masked Language Model（MLM）

BERT

- Pre-training task



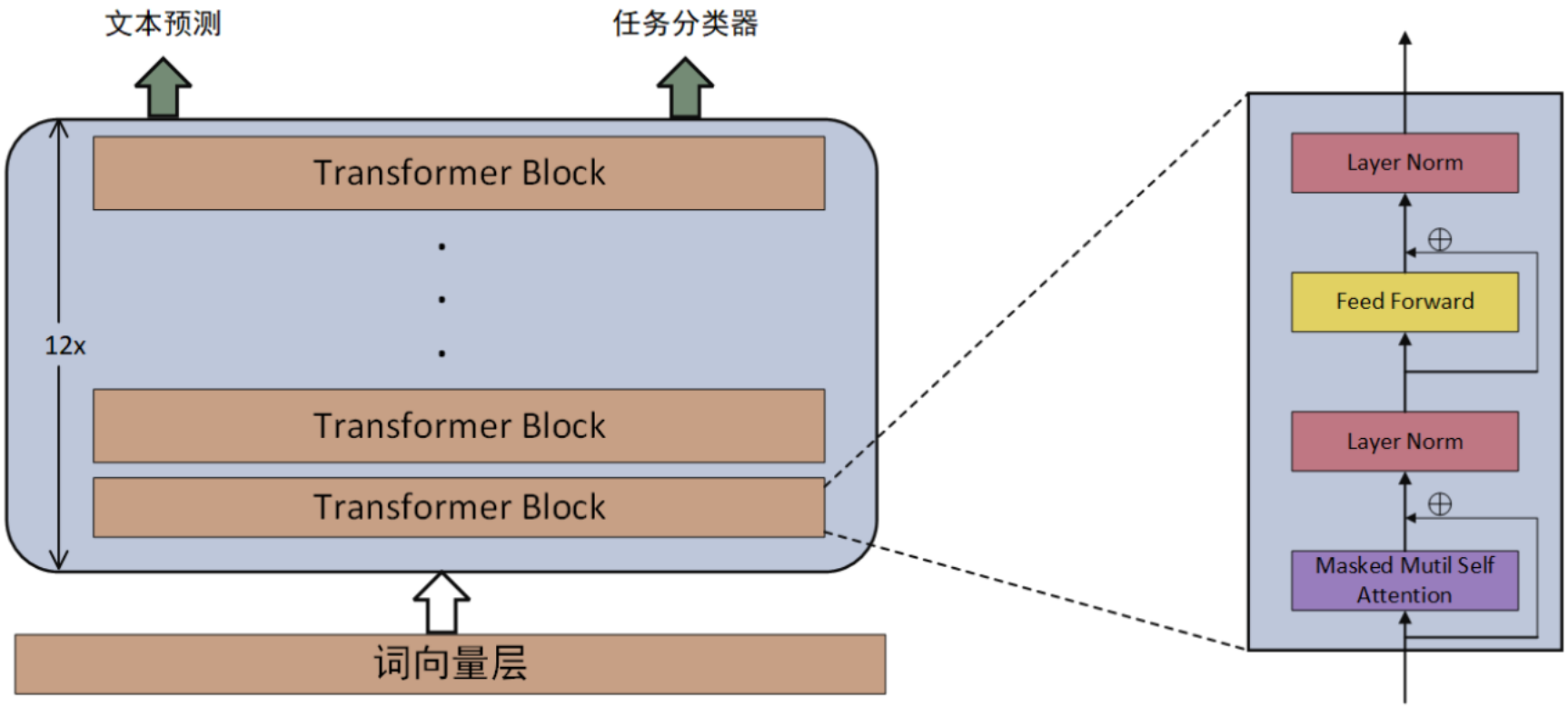
Next Sentence Prediction（NSP）

Outline

- AI & NLP
- Tokenize
- Word Embedding
- Transformer
- BERT
- GPT

GPT

- Decoder only



GPT model

GPT

- Pre-training and Fine-Tuning

Next word Prediction

$$L(\theta) = -\sum_{t=1}^{n} \log P_\theta(x_t | x_1, x_2, ..., x_{t-1})$$