中国科学技术大学
University of Science and Technology of China

《人工智能数学原理与算法》

第6章 自监督学习

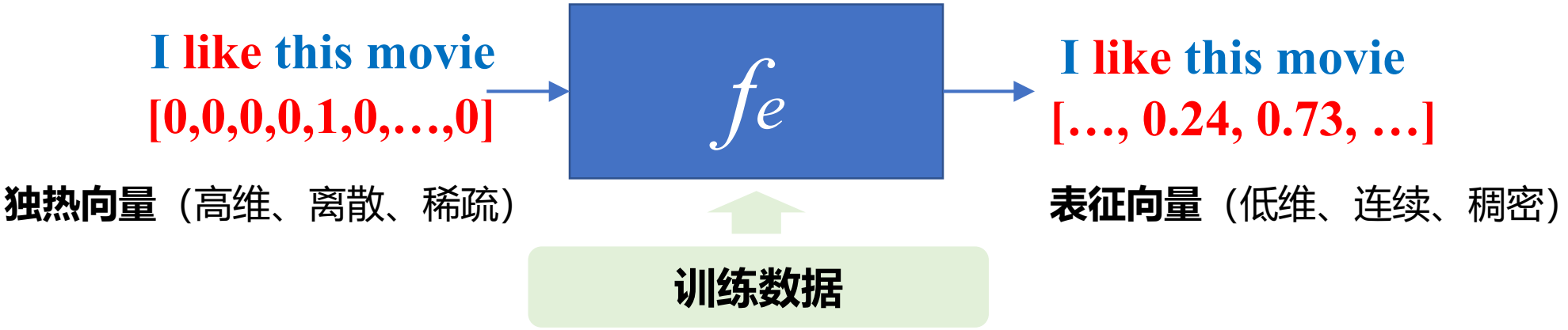
6.2 word2vec与BERT模型

凌震华

zhling@ustc.edu.cn

回顾：表征学习与自监督学习

- **表征学习**



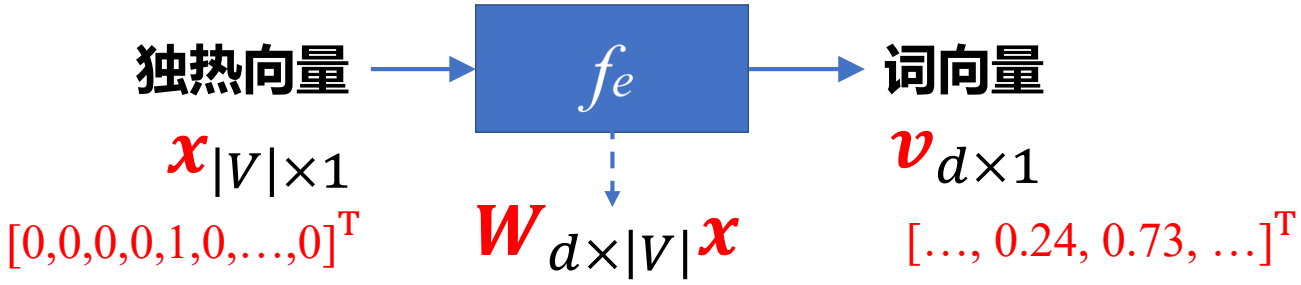
- **自监督学习**
  - 是一种特殊的表征学习，能够从无标签数据集中学习良好的数据表征

本节将重点介绍两种得到单词表征向量的自监督学习模型 word2vec & BERT

目录

词向量

- 词向量又称词嵌入(word embedding)
- 将每个单词**独立**地映射为固定维度的实数向量



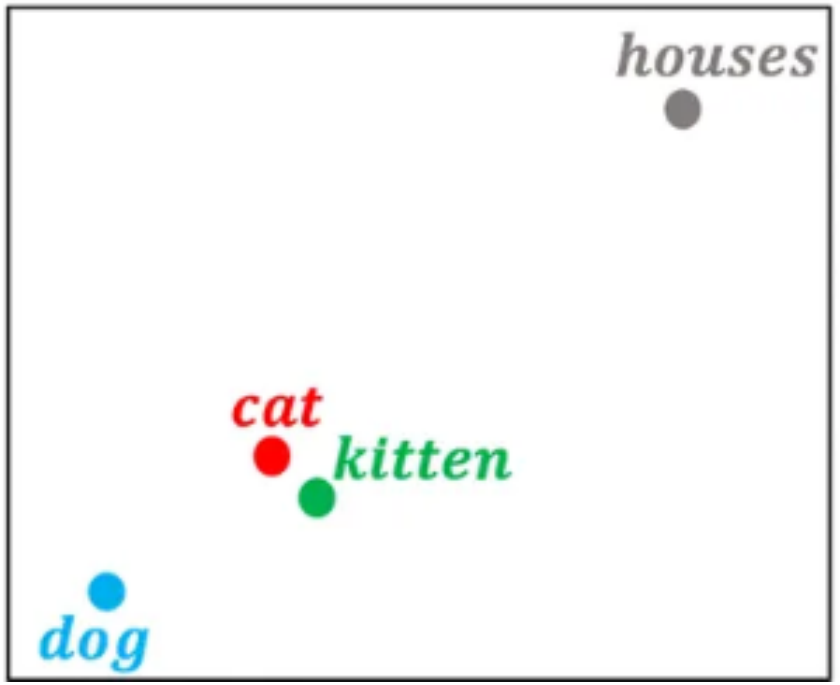
  - 映射过程通过矩阵相乘实现，$\boldsymbol{W}$为词向量矩阵
  - 词表中第 i 个单词的词向量为词向量矩阵中的第 i 列
- 词向量的性质
  - $d \ll |\mathrm{V}|$
  - 能够捕捉单词间的语义相似性和语义关联性

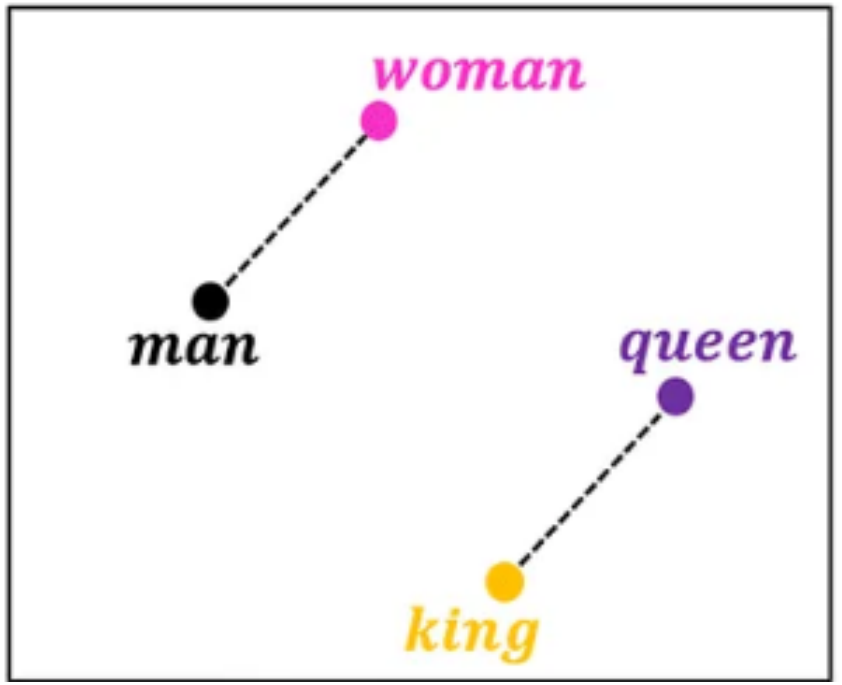
词向量

- 为什么叫**词“嵌入”**？
  - “嵌入”(embedding) 原本是数学上的一个概念
    - 将一个对象映射到另一个空间，同时保留其关键结构或属性
    - 例如：将三维物体投影到二维平面（保留形状关系）；交通线路图（保留位置关系）

词向量

- 为什么叫**词“嵌入”**？
  - 在表征单词时，借用了数学中“嵌入”的概念
    - 将离散语言符号映射到连续向量空间，并保留语义关系





word2vec

- Google 于 2013 年开源推出的一个用于获取词向量的工具包

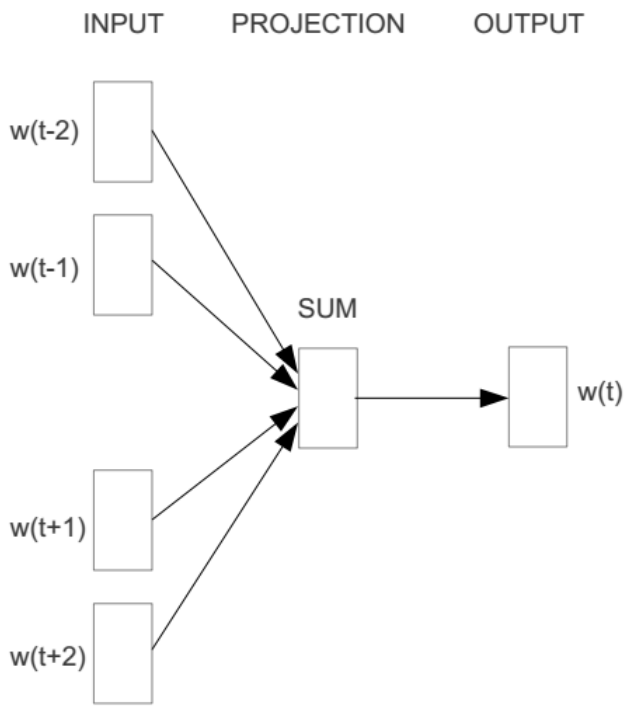
  https://code.google.com/archive/p/word2vec/

- 基于**分布式语义(distributional semantics)假设**
  - 一个单词的意义是通过其上下文体现的
  - 例如：“猫”和“狗”常出现在“宠物”“动物”“尾巴”等相似上下文中

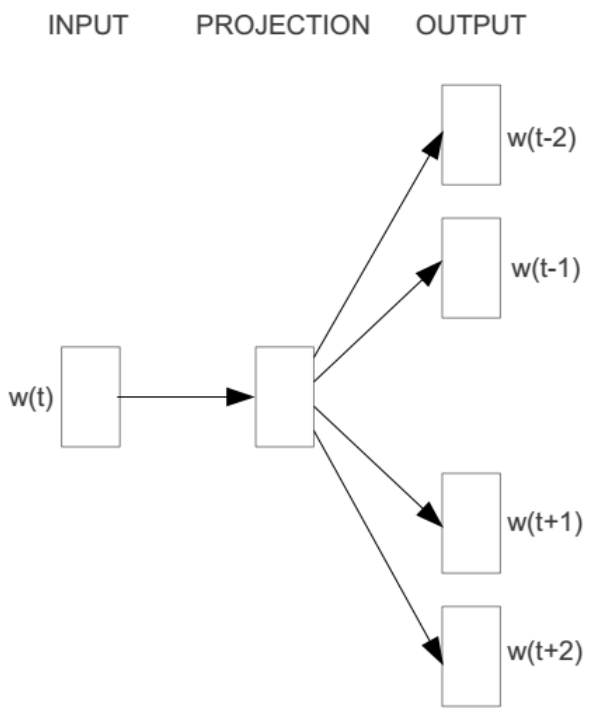
- 核心思想：**基于神经网络预测临近单词**
- 包括**skip-gram**和**CBOW**两种基础结构 [Mikolov et al. 2013]

word2vec

- Continuous Bag of Word (CBOW)：使用邻近词预测中心词
- Skip-gram (SG): 使用中心词预测邻近词



CBOW



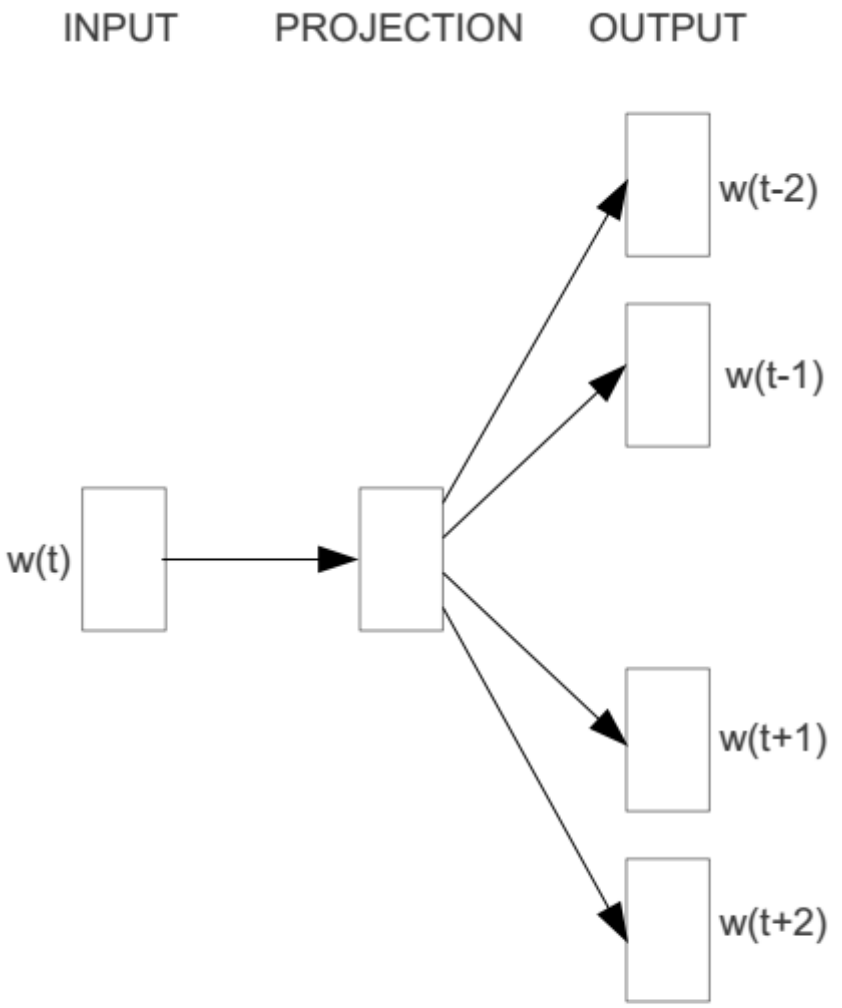
Skip-gram

目录

Skip-gram

- 以当前单词为输入，预测其上下文窗口（大小为 2K 个单词）内的每个相邻单词

- 例如当 K = 2 时，给定单词w(t)，需要预测[w(t-2), w(t-1), w(t+1), w(t+2)]

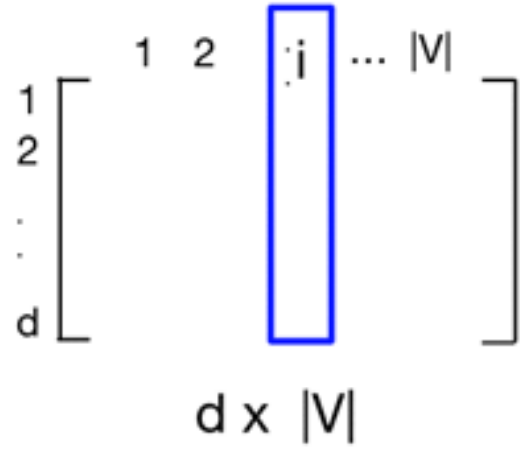


Skip-gram

Skip-gram

- 为每个单词学习 2 种词向量

  - **输入词向量**v：位于输入矩阵 W 中，输入矩阵的第 i 列是词汇表中第 i 个单词的 d×1 维词向量



  - **输出词向量**c：位于输出矩阵 C 中，输出矩阵的第 i 列是词汇表中第 i 个单词的 d×1 维词向量

Skip-gram

- 训练时遍历语料库，指向单词$w(t)$，其在词汇表中的索引为j，我们将其记为$w_j(1 < j < |V|)$

- 假设要预测$w(t+1)$，其在词汇表中的索引为$k(1 < k < |V|)$

- 因此我们的任务是计算$P(w_k|w_j)$

Skip-gram——基于相似度的概率计算

- **如何计算$P(w_k|w_j)$**：使用目标单词（中心词）向量与上下文单词（邻近词）向量的点积衡量相似度，进一步将相似度转化为概率
  - 目标单词w_j的词向量v_j；上下文单词w_k的词向量c_k
  - 两个向量的点积越高，它们就越相似，$Similarity(j,k) \propto c_k \cdot v_j$

$$c_k \cdot v_j = \sum_{m=1}^{d} c_{k,m} v_{j,m}$$

  - 使用 softmax 函数将其转化为概率

$$p(w_k|w_j) = \frac{exp(c_k \cdot v_j)}{\sum_{i \in |V|} exp(c_i \cdot v_j)}$$

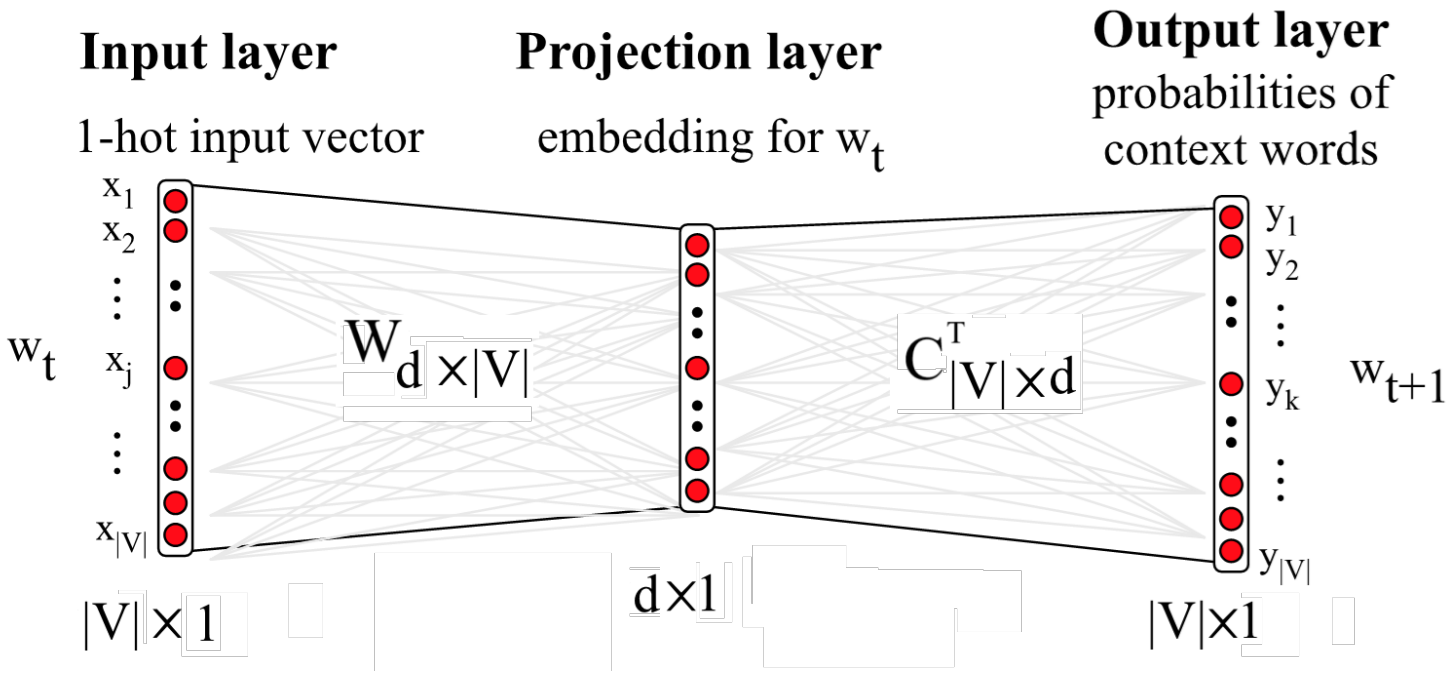
Skip-gram——基于相似度的概率计算

- 来自 W 和 C 的词向量
  - 由于每个单词w_j都有两个词向量v_j和c_j
  - 在下游任务中，我们可以仅使用其中某一个、将它们相加或拼接

- 学习过程
  - 从初始词向量开始（例如随机初始化）
  - 迭代调整单词的词向量，使目标单词的词向量更接近邻近单词的词向量，远离其他单词的词向量

Skip-gram——基于神经网络的概率计算

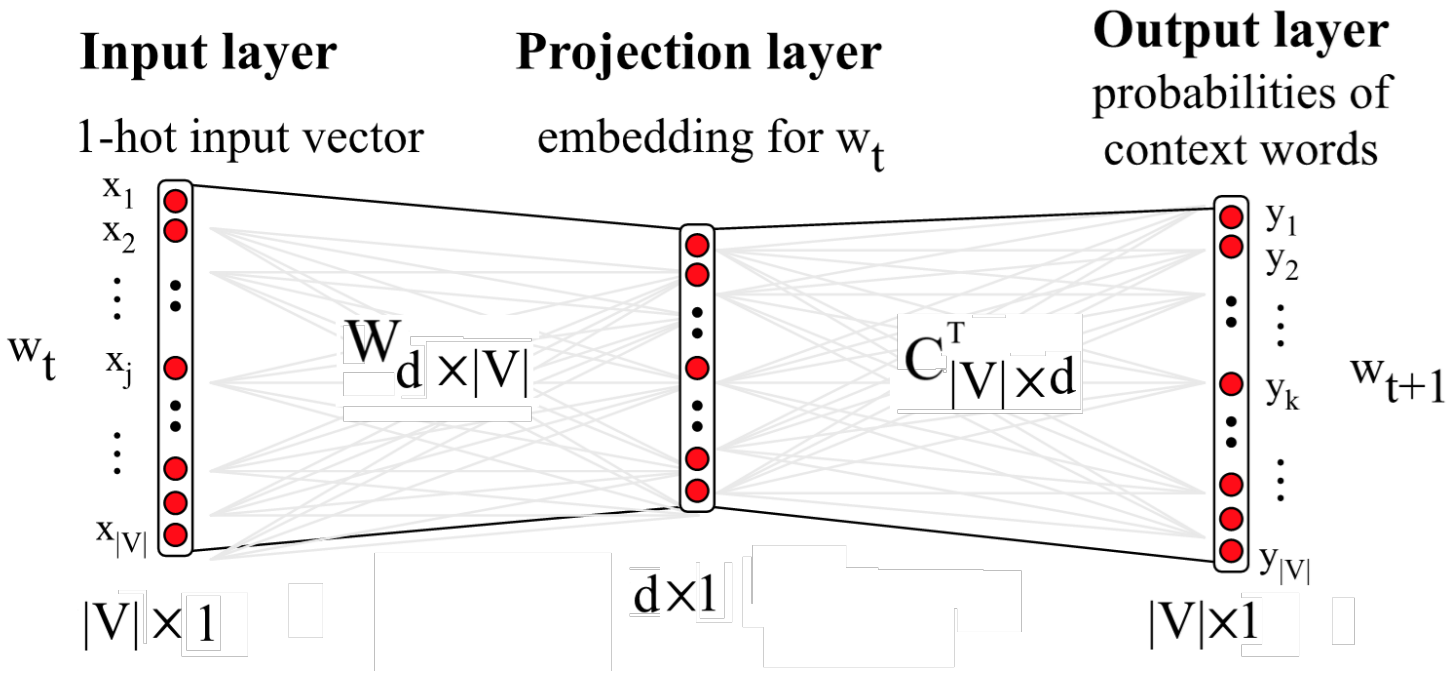
- 使用神经网络表示$P(w_k|w_j)$的计算过程
  - 输入层：独热向量



Skip-gram——基于神经网络的概率计算

- 使用神经网络表示$P(w_k|w_j)$的计算过程
  - 输入层：独热向量
  - 投影层

$$\boldsymbol{h} = \boldsymbol{v}_j = \boldsymbol{W}\boldsymbol{x}$$



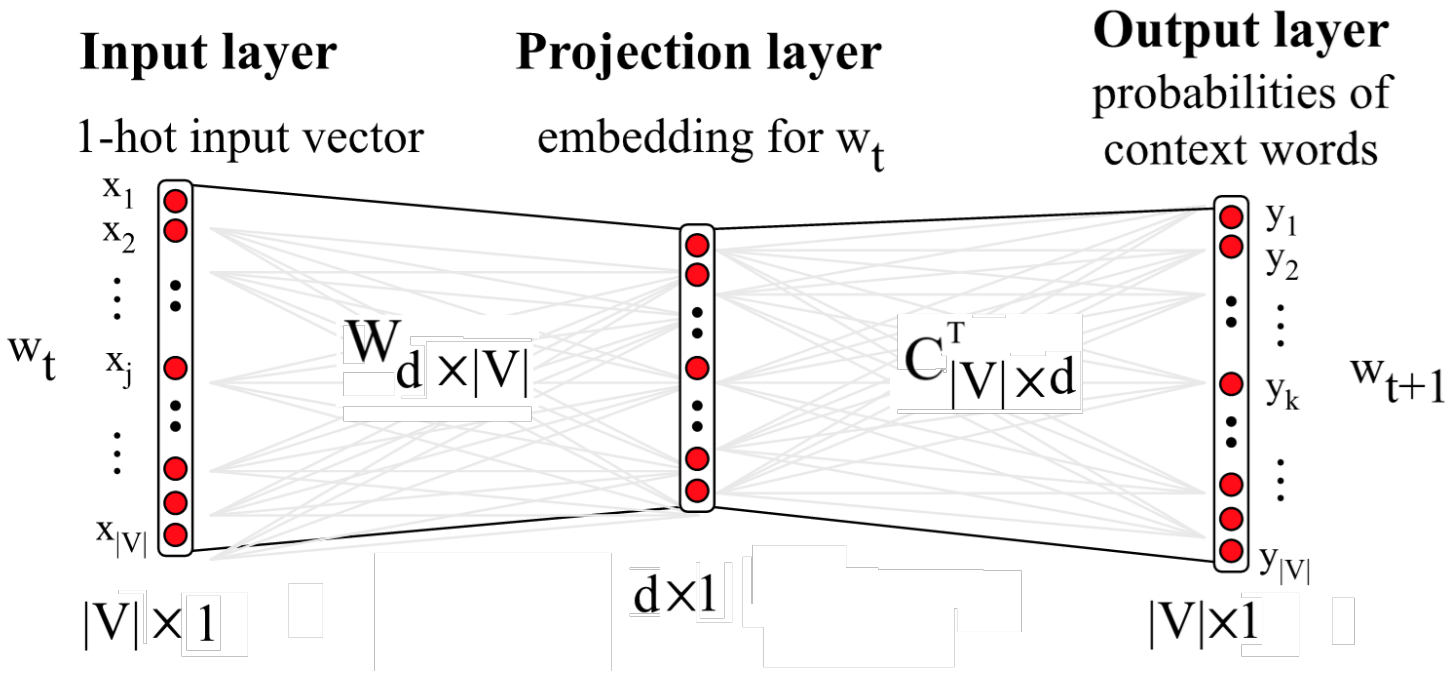
Skip-gram——基于神经网络的概率计算

- 使用神经网络表示$P(w_k|w_j)$的计算过程
  - 输入层：独热向量
  - 投影层
  - 输出层

$$\boldsymbol{h} = \boldsymbol{v}_j = \boldsymbol{W}\boldsymbol{x}$$

$$\boldsymbol{o} = \boldsymbol{C}^{\mathrm{T}}\boldsymbol{h}$$

$$o_k = \boldsymbol{c}_k \cdot \boldsymbol{v}_j$$



Skip-gram——基于神经网络的概率计算

- 输出层
  - 使用softmax函数计算输出概率

$$P(w_k|w_j) = \frac{\exp(o_k)}{\sum_{i=1}^{|V|}\exp(o_i)}$$

  - **存在问题**：分母需要对词汇表中的所有单词进行计算
  - **解决方法**：只对少数负样本单词进行采样计算

Skip-gram

- **负采样(negative sampling)损失函数**
  - 希望目标单词与上下文单词相似

lemon, a [tablespoon of apricot preserves or] jam

c1 c2 w c3 c4

$\sigma(c1 \cdot w) + \sigma(c2 \cdot w) + \sigma(c3 \cdot w) + \sigma(c4 \cdot w)$ 尽量高，其中 $\sigma(x) = \frac{1}{1+e^{-x}}$

  - 希望目标单词与随机选择的个 “噪声单词” (负样本) 不相似

[cement metaphysical dear coaxial apricot attendant whence forever puddle]

n1 n2 n3 n4 n5 n6 n7 n8

$\sigma(n1 \cdot w) + \sigma(n2 \cdot w) + \ldots + \sigma(n8 \cdot w)$ 尽量低

$$\text{最大化 } \log \sigma(c \cdot w) + \sum_{i=1}^{\kappa} \mathbb{E}_{w_i \sim p(w)} \left[\log \sigma(-w_i \cdot w)\right]$$

词向量的性质

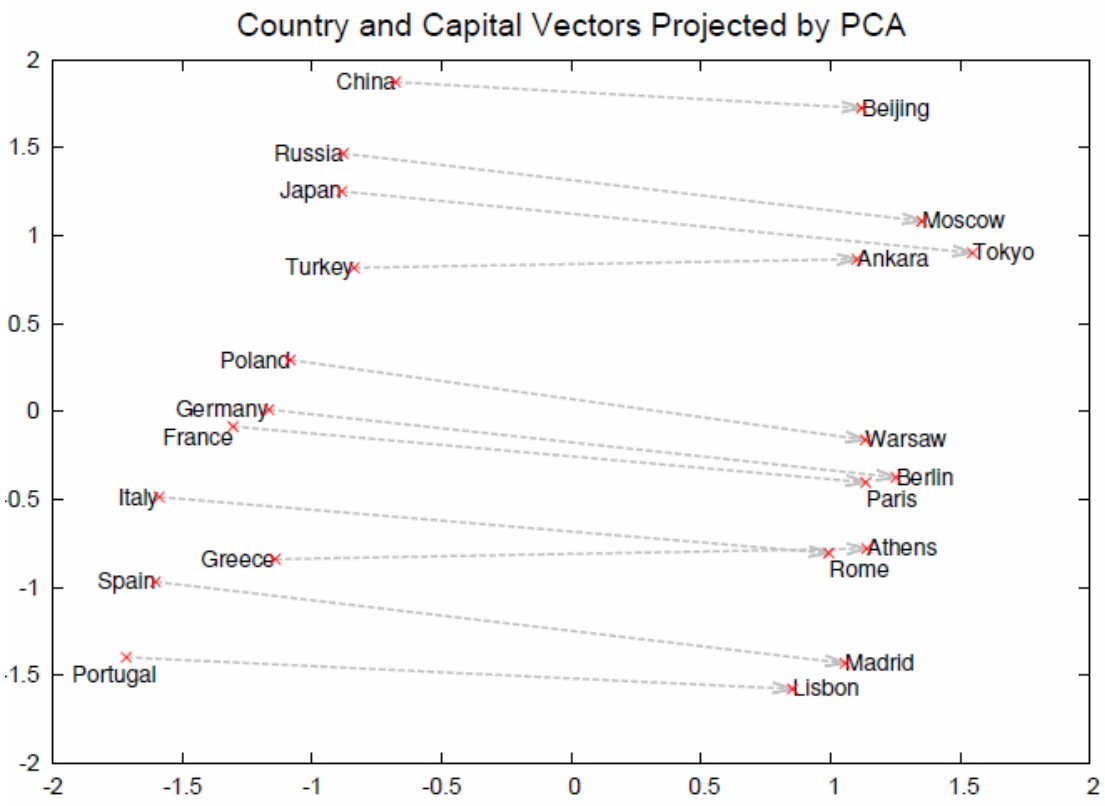
- 度量单词间语义相似性

目标词:	dog	book	cricket	boat	gold
	cat	books	badminton	ship	silver
	dogs	project	rugby	truck	blue
	puppy	review	lacrosse	plane	diamond

词向量的性质

- 捕捉单词间语义关系

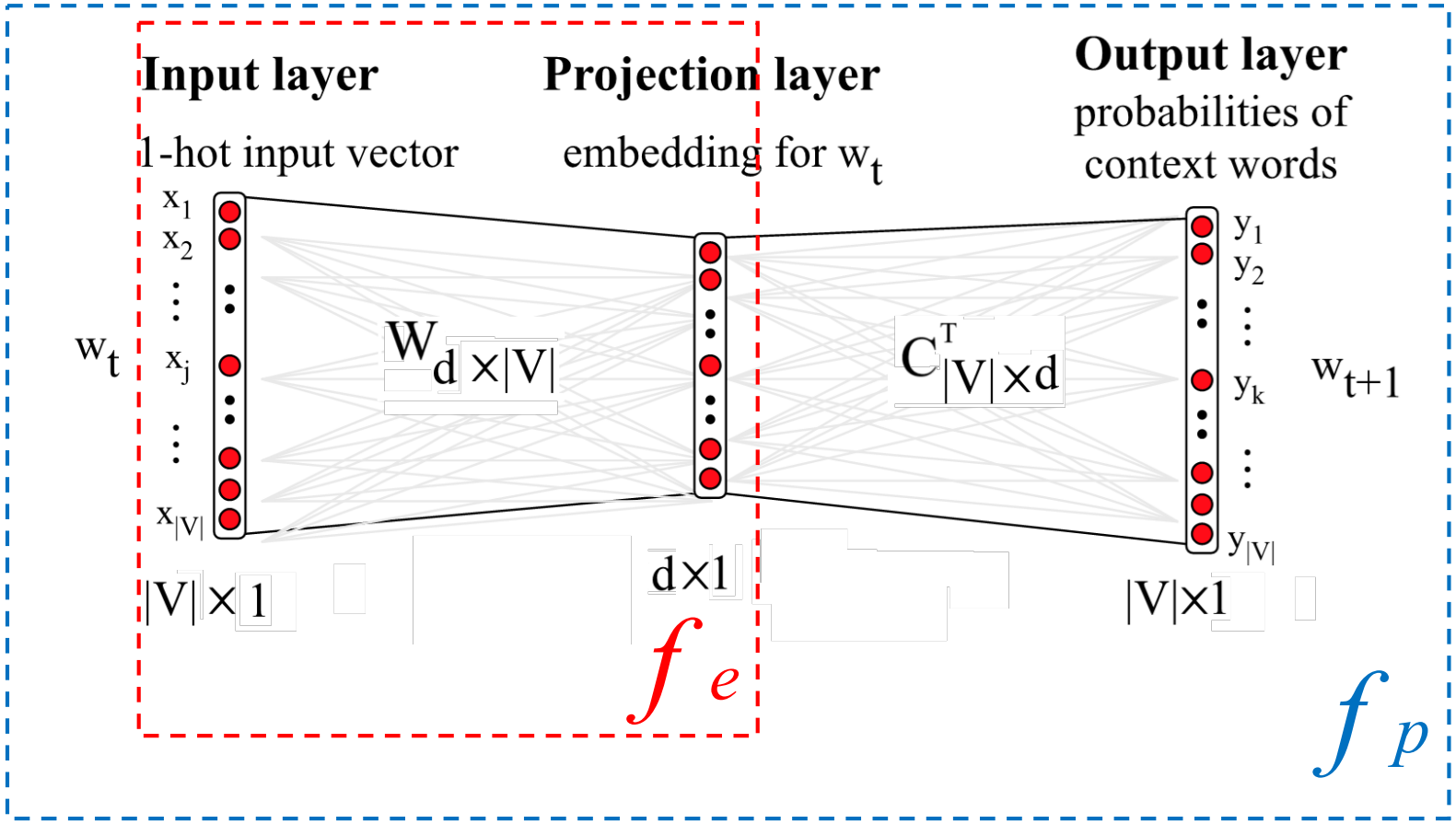
vector(*'king'*) - vector(*'man'*) + vector(*'woman'*) ≈
vector('queen')



[Image credits: Mikolov et al (2013) "Distributed Representations of Words and Phrases and their Compositionality", *NIPS*]

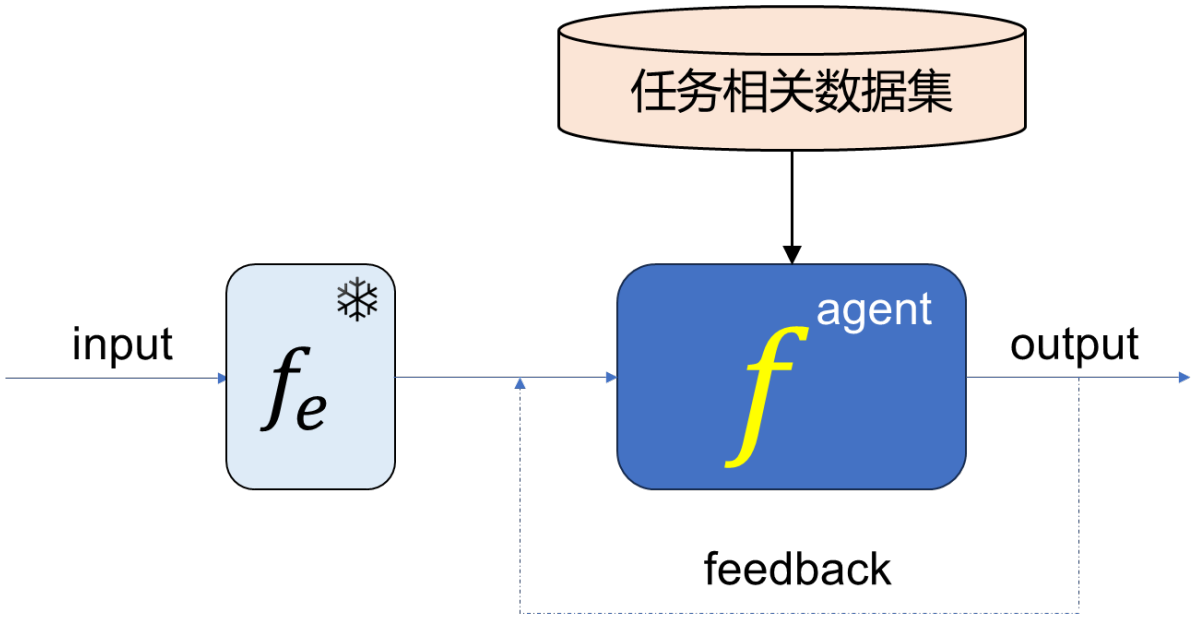
词向量的应用范式

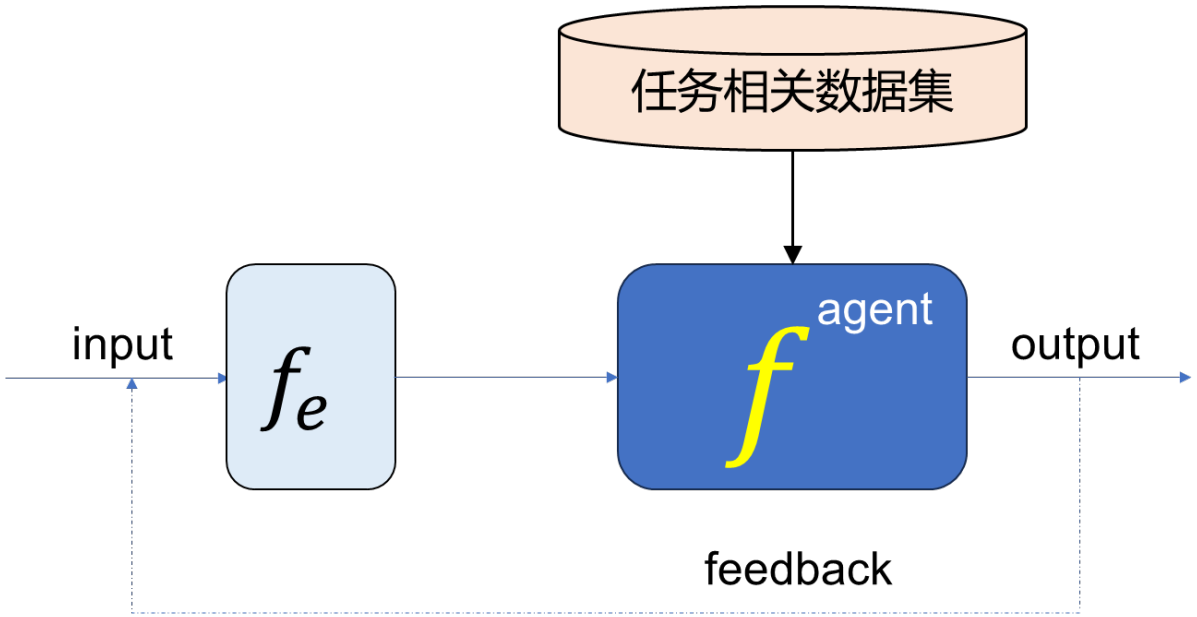
- 在下游任务中作为输入单词的表征



词向量的应用范式

- **固定表征微调或联合微调**

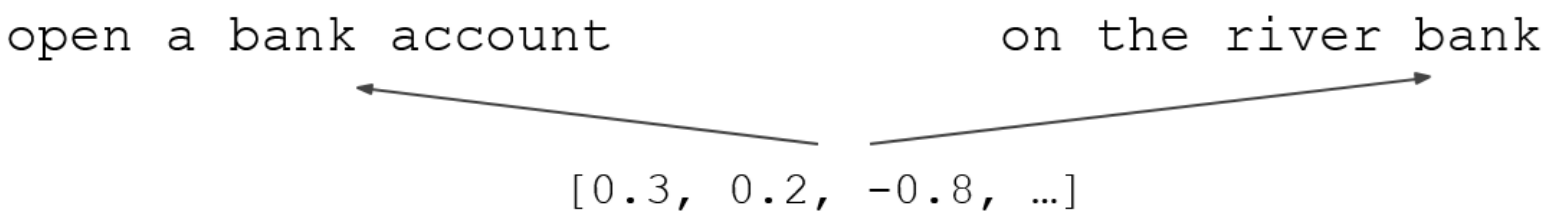




目录

word2vec词向量

- **存在问题**：词向量是静态的，和上下文无关 (context free)

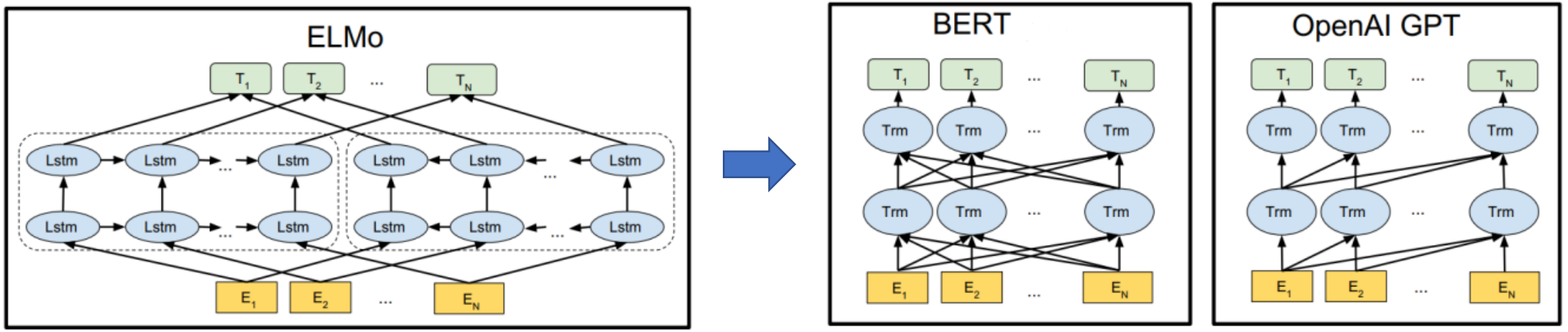


- **解决方法**：构建上下文相关的单词表征

基于 Transformer 的文本自监督预训练模型

- Transformer 为文本表征提供了新的选择
  - 在 Transformer 之前，LSTM 是自然语言处理中应用最广泛的模型
  - Transformer相对LSTM在建模长距离相关性、逐层并行计算方面具有优势

基于Transformer的文本自监督预训练模型



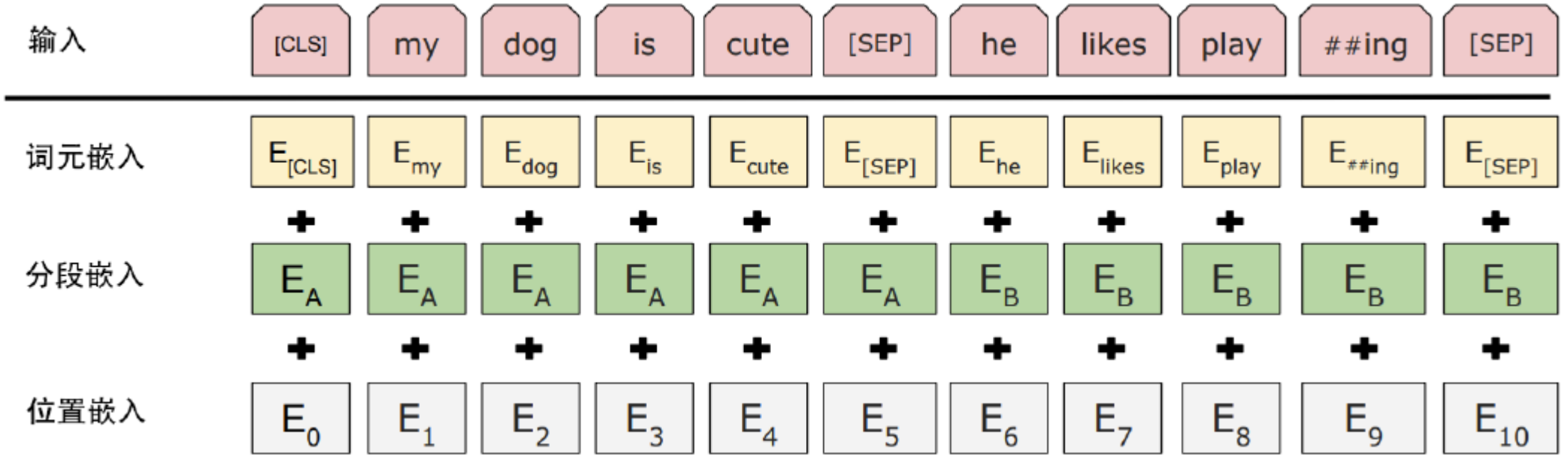
[Peters et al. 2018] [Devlin et al. 2018] [Radford et al. 2018]

BERT

- 基于Transformer的双向编码器
  - Bidirectional Encoder Representations from Transformers
- 使用子词(word-piece)作为模型输入
- 基于自监督任务进行学习，与下游任务无关
  - 掩码语言模型 (masked language model)
  - 下一句预测 (next sentence prediction)
- 不同的模型尺寸：$\text{BERT}_{\text{base}}$ $\text{BERT}_{\text{large}}$
- 上下文相关的动态单词表征

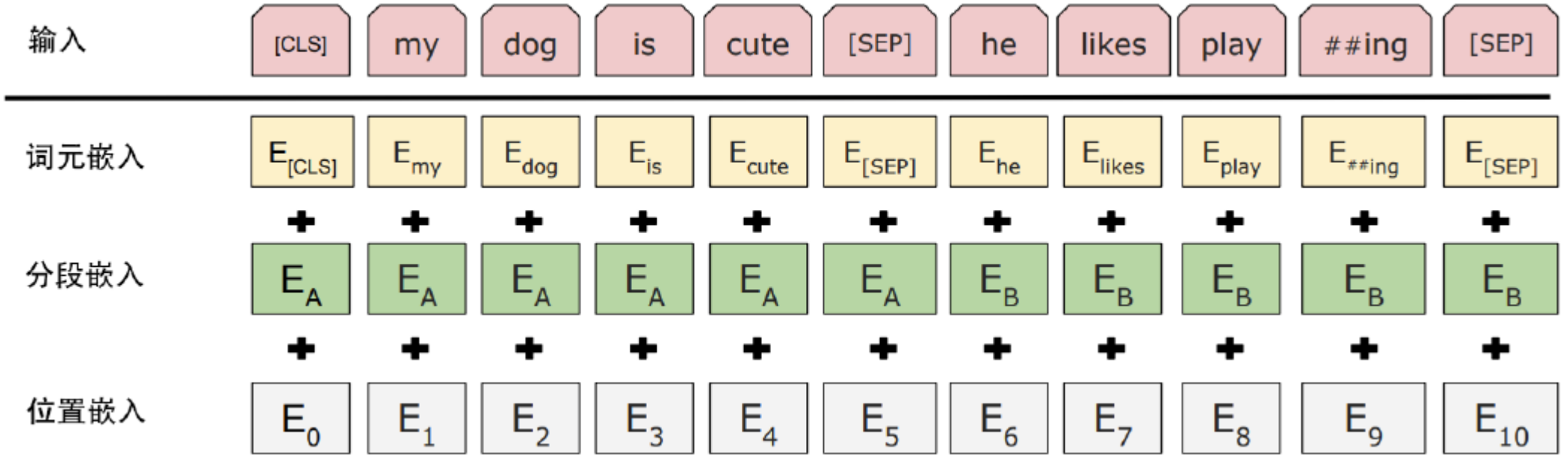
输入

- 训练样例由两个句子拼接组成，每个子词输入向量由三个向量相加
  - 词元嵌入(token embeddings)：表示每个词元的固定维度向量
  - 分段嵌入(segment embeddings)：用来指示当前两个词元来自哪个句子
  - 位置嵌入(position embeddings)：用于表示词元在完整序列中的位置



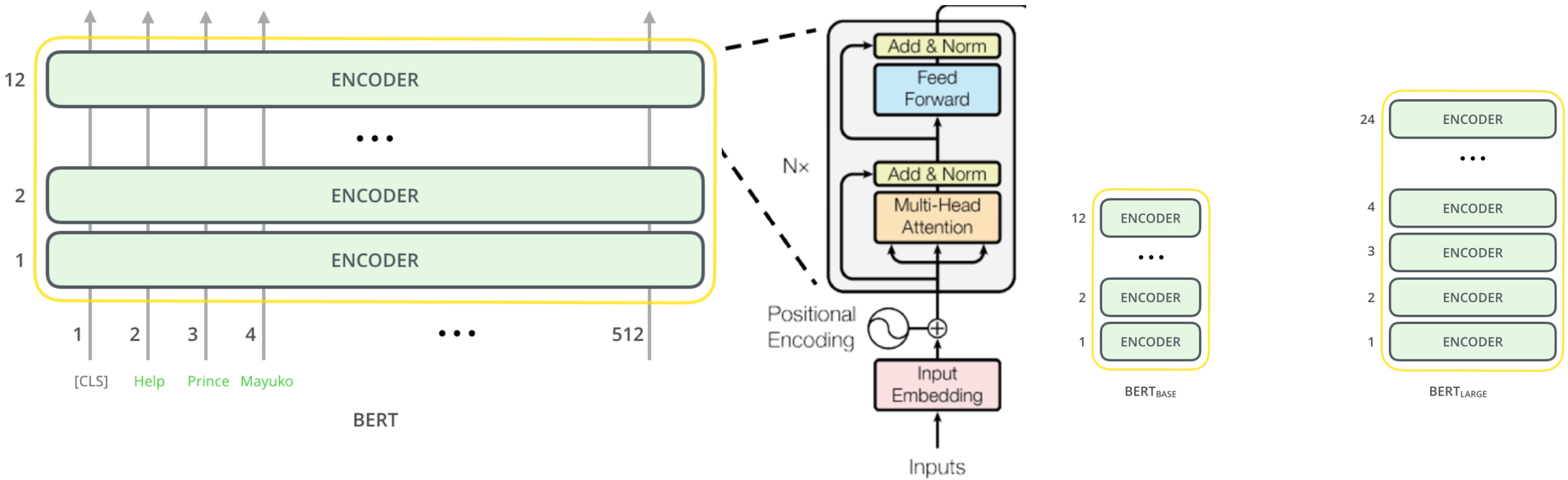
输入

- 向量维度：768 ($BERT_{base}$) 1024 ($BERT_{large}$)
- 特殊词元
  - [CLS] 序列开头作为句子级别的表征，提供整个句子的句义表征
  - [SEP]两个句子的结尾处，用来指示句子切换以及句子结束位置



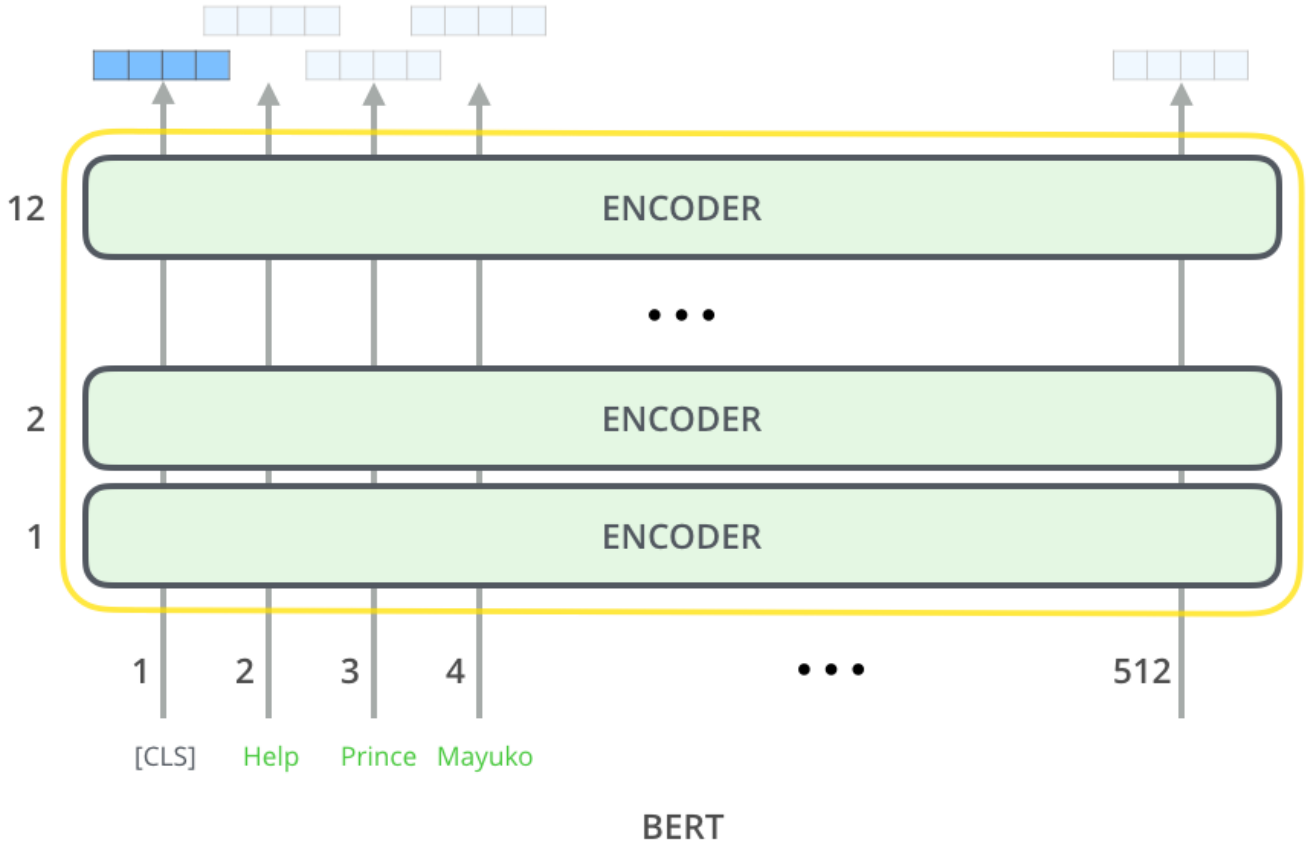
编码

- 一个多层 Transformer 编码器，有12 (base版本) 或 24 (large版本) 层

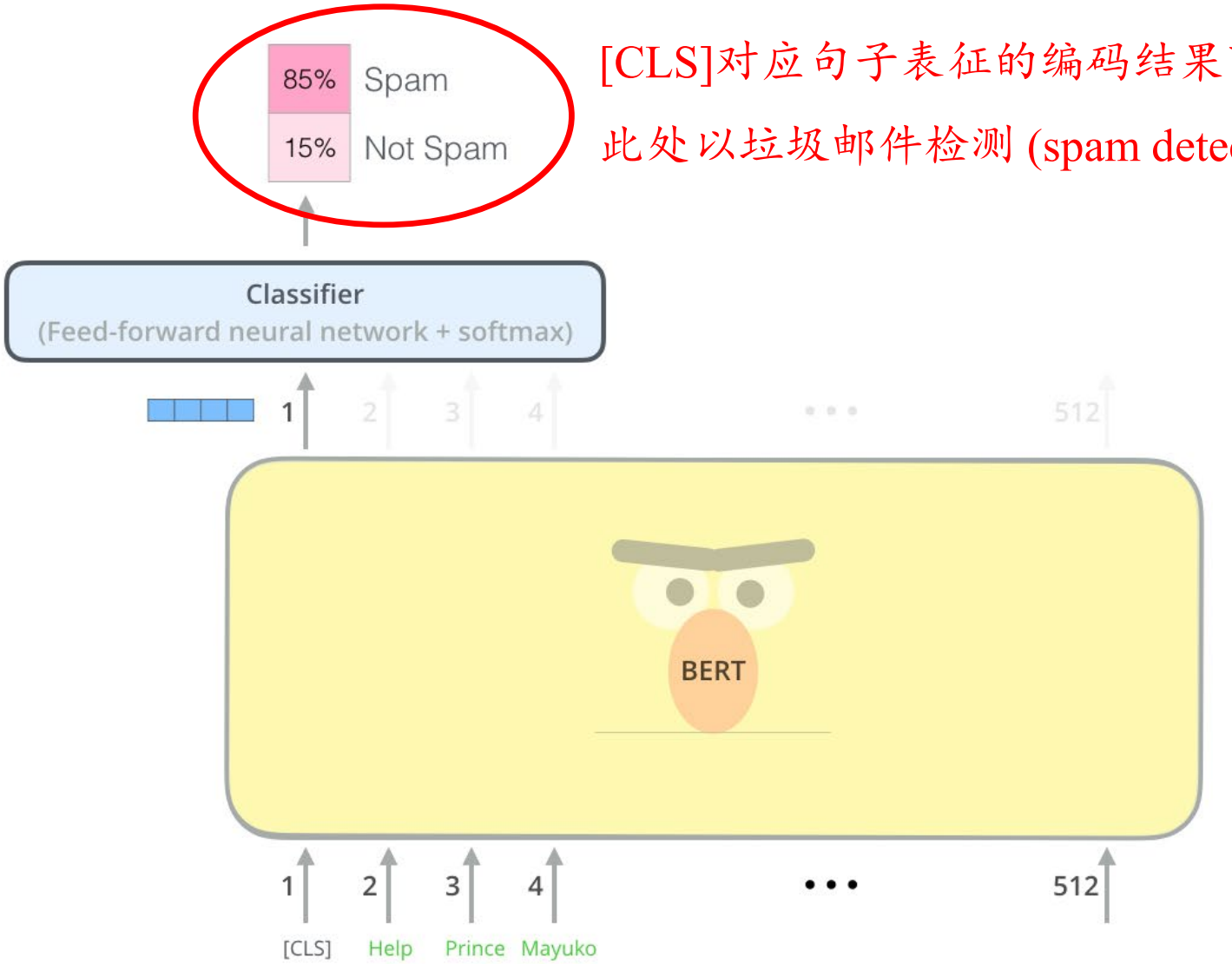


http://jalammar.github.io/illustrated-bert/

输出


12
ENCODER
• • •
2
ENCODER
1
ENCODER
1
2
3
4
• • •
512
[CLS]
Help
Prince
Mayuko
BERT

输出



[CLS]对应句子表征的编码结果可用于句子分类

此处以垃圾邮件检测 (spam detection)任务为例

预训练 – 自监督学习

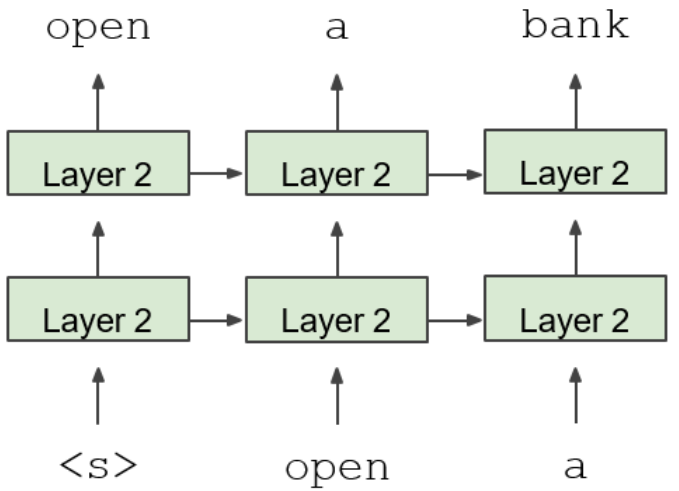
- **掩码语言模型 (masked language model, MLM)**
  - 随机掩码一些输入单元(token)并预测它们

    “ Apple is red” → “Apple [MASK] red” → “ Apple is red”
  - 最大化正确单元的概率 P(is | Apple [MASK] red)
  - **学习单词级别的信息**
- **下一句预测 (next sentence prediction, NSP)**
  - 对两个句子是否具有先后顺序关系进行二元判断

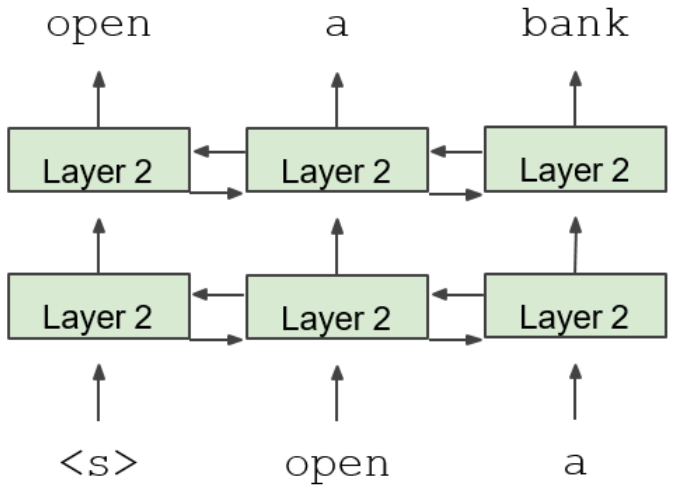
    S1: “Apple is red” S2: “No, there are also green apples” → yes (S2紧跟S1)
  - 最大化正确判断的概率P(yes | S1 S2)
  - **学习句子级别的信息**

预训练 – 掩码语言模型

- 利用掩码语言模型促进上下文信息的**双向学习**
  - 自回归语言模型只能从左到右或从右到左进行训练，单向建模单词间关系

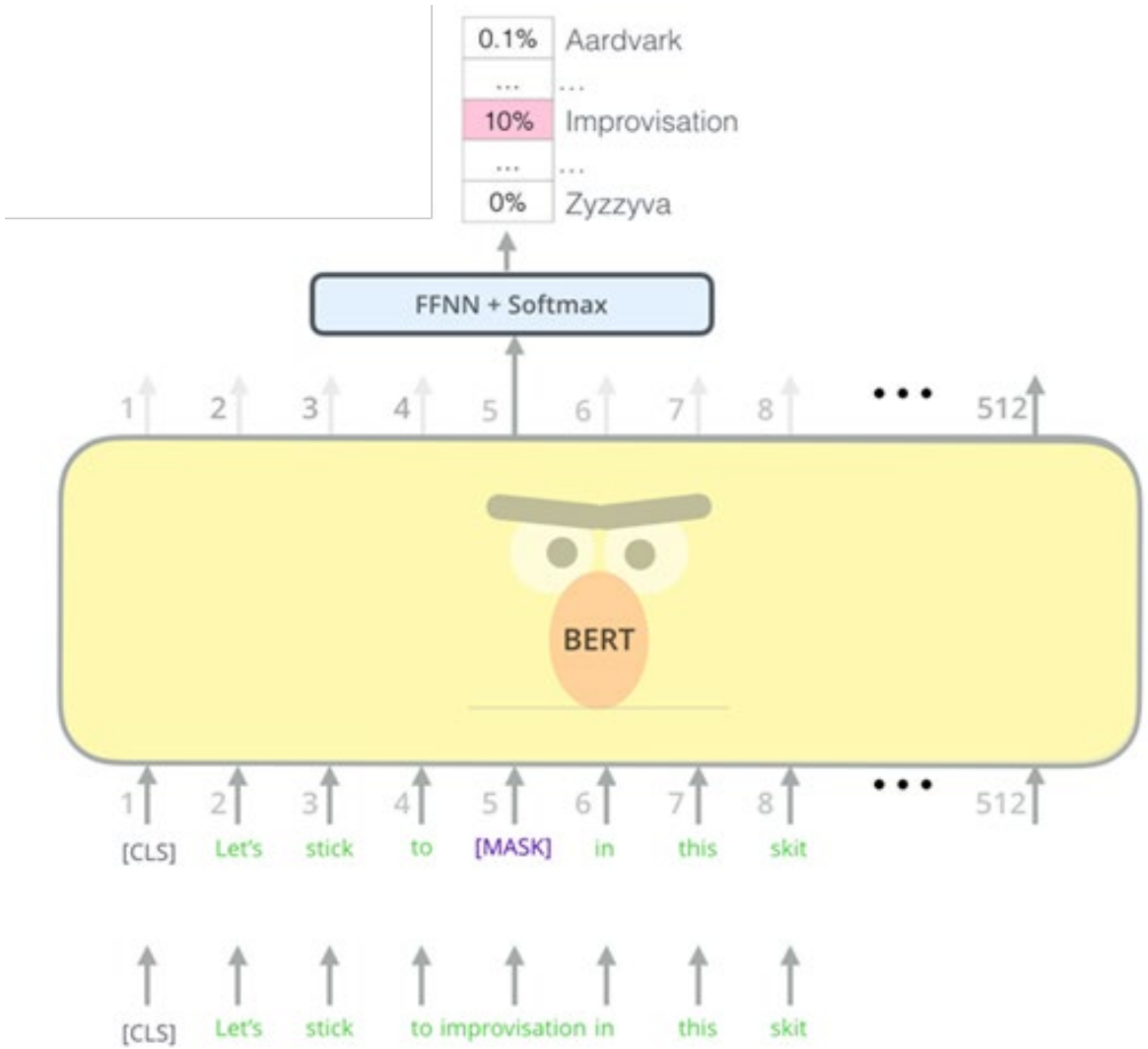


  - 直接进行双向编码会使每个单词间接 “看到自己”，模型无法有效训练



预训练 – 掩码语言模型

- 利用掩码语言模型促进上下文信息的**双向学习**
  - 被掩码词元位置的BERT编码输出结果经过前馈层和softmax激活
  - 输出该位置对应不同单词的概率
  - 训练时最大化真实单词的概率



预训练 – 掩码语言模型

- **存在问题**
  - 它造成了预训练(pre-training)阶段和下游任务微调(fine-tuning)的不一致
  - 因为在微调期间 [MASK] token不会出现

- **缓解策略**
  - BERT 并不总是用 [MASK] token替换被掩码的单词
  - 所有token的 15% 会被掩码
    - 其中 80% 的情况会被替换为指定的[MASK]
    - 10% 的情况会被替换为随机token
    - 10% 的情况保持不变

预训练 – 下一句预测

- 许多重要的下游任务如问答（Question Answering, QA）和自然语言推理（Natural Language Inference, NLI），都基于对两个句子之间关系的理解

前提（Premise） 一名穿着红色外套的女子正在公园里遛狗

假设（Hypothesis）

公园里有一名女子	→**蕴含（Entailment）**
这名女子独自在公园里	→**矛盾（Contradiction）**
这名女子喜欢户外活动	→**中性（Neutral）**

- 但当前的语言模型没有捕捉到这类信息
- NSP 就是为了对这类信息进行建模，思路类似学习句子嵌入
- [CLS]输出经前馈层和softmax激活，输出是否下一句的概率

预训练 – 实现细节

- **数据（英文）**
  - BooksCorpus (800M 单词)
  - Wikipedia (2,500M 单词)

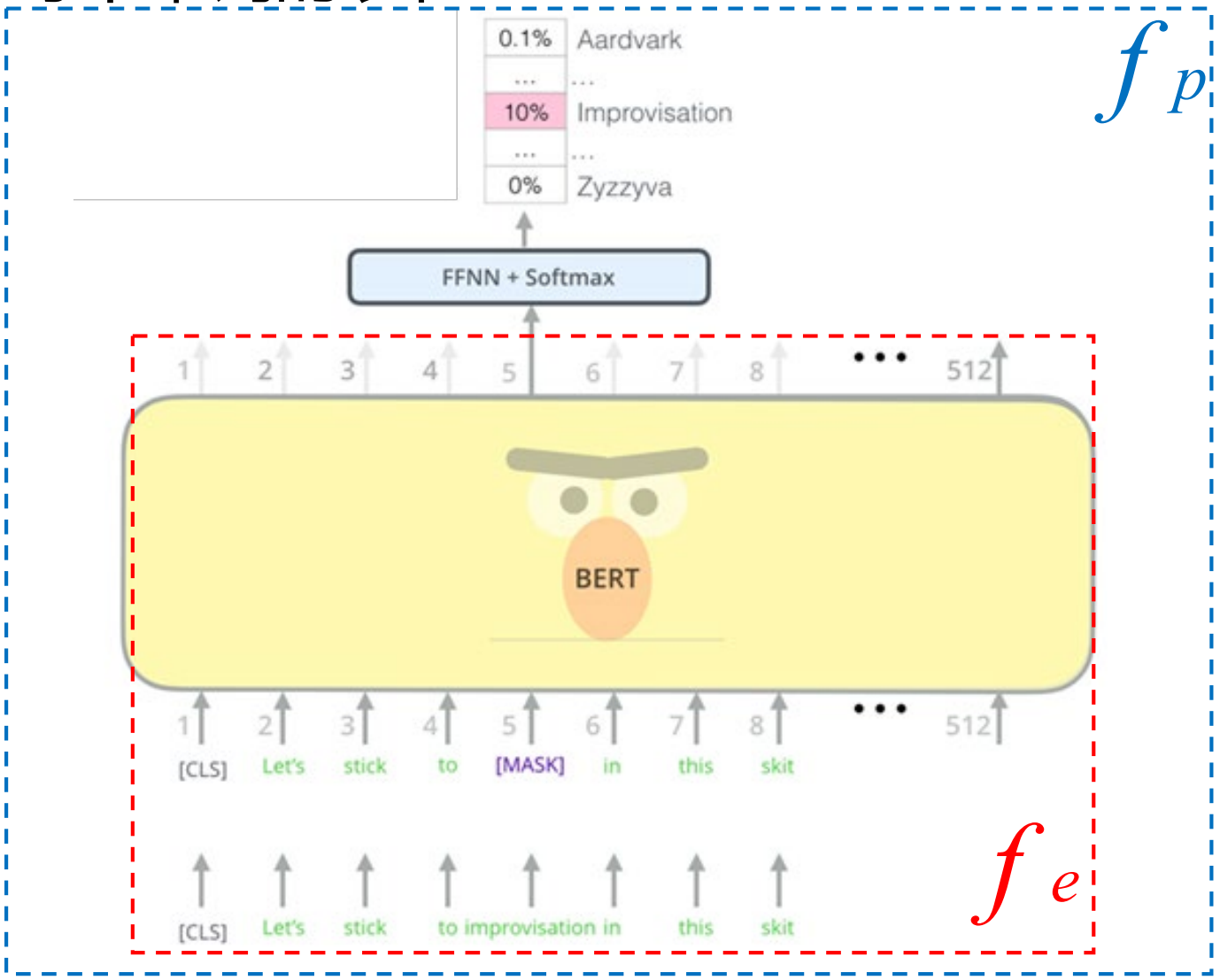
- **Base版本**
  - 12 层 Transformer，约 1.1 亿(110M)个参数

- **Large版本**
  - 24 层 Transformer，约 3.4 亿(340M)个参数

目录

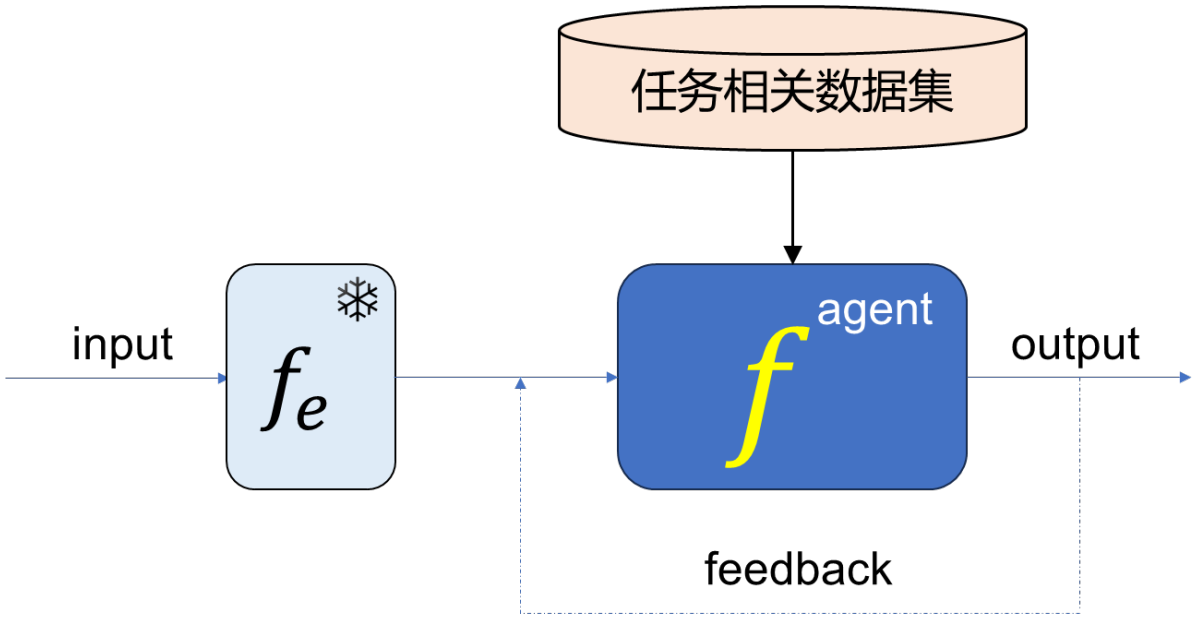
BERT的应用范式

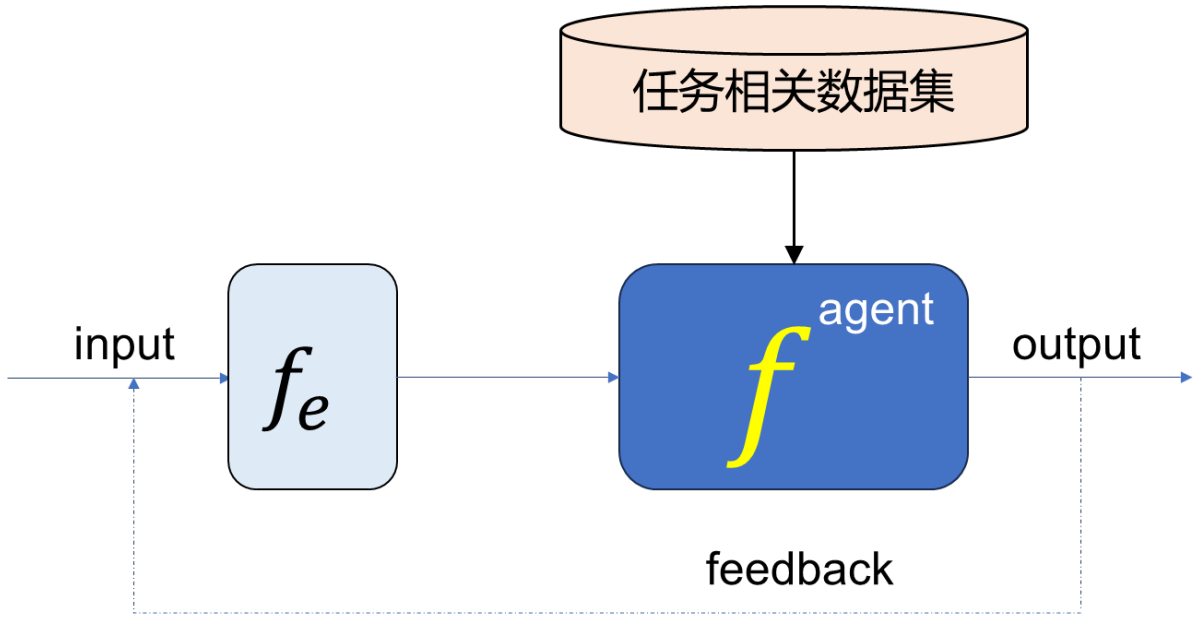
- 得到句子及句中每个单词的表征



BERT的应用范式

- **固定表征微调或联合微调**

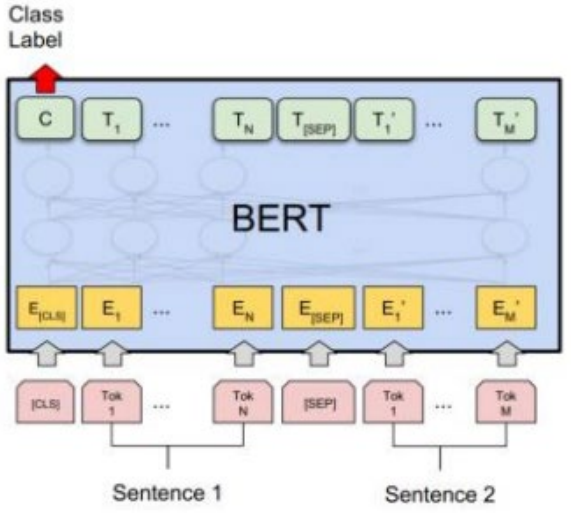




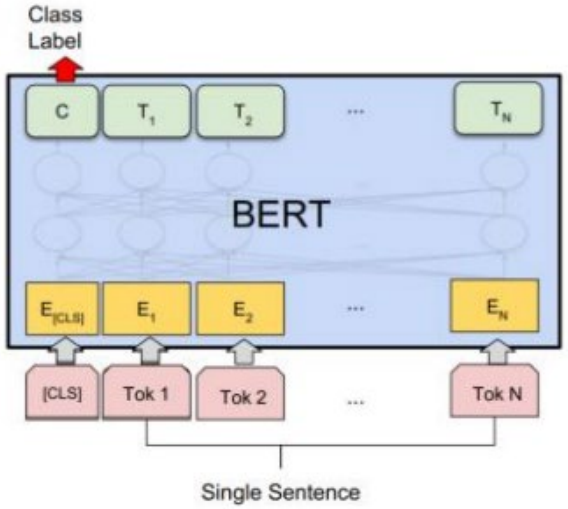
微调(fine-tune)

- 利用有监督数据标签，指导模型参数的微调更新

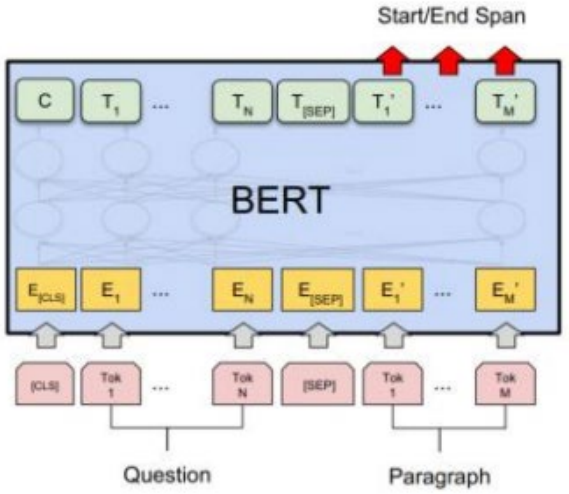
- 针对不同下游任务对于BERT有不同微调方式
  - 句对分类(sentence pair classification)
  - 单句分类(sentence classification)
  - 词元标记(token tagging)



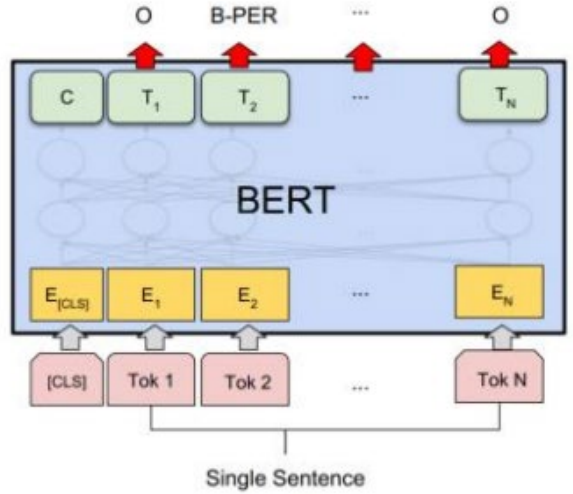
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



(b) Single Sentence Classification Tasks: SST-2, CoLA



(c) Question Answering Tasks: SQuAD v1.1



(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

性能评估（GLUE 基准测试）

- 句子对任务
  - MNLI, multi-genre natural language inference
  - QQP, Quora question pairs
  - QNLI, question natural language inference
  - STS-B, the semantic textural similarity benchmark
  - MRPC, Microsoft Research paraphrase corpus
  - RTE, recognizing textual entailment
  - WNLI, Winograd NLI/a small natural language inference dataset
- 单句分类
  - SST-2, the Stanford sentiment treebank
  - CoLA, the corpus of linguistic acceptability

性能评估（GLUE 基准测试）

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
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
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**91.1**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**81.9**

性能评估 (SQuAD)

- SQuAD机器阅读理解任务
  - Stanford Question Answering Dataset (SQuAD)
  - 包含 10 万个众包的问题 / 答案对

- 输入段落

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

- 输入问题

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

- 输出答案

Within a cloud

性能评估 (SQuAD)

- 仅使用 BERT 的效果优于其他复杂模型和集成模型
- $BERT_{large}$ 比$BERT_{base}$效果更好
- 添加额外数据有助于提升性能

System	Dev		Test	
	EM	F1	EM	F1
Leaderboard (Oct 8th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
#1 Single - nlnet	-	-	83.5	90.1
#2 Single - QANet	-	-	82.5	89.3
Published				
BiDAF+ELMo (Single)	-	85.8	-	-
R.M. Reader (Single)	78.9	86.3	79.5	86.6
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$BERT_{BASE}$ (Single)	80.8	88.5	-	-
$BERT_{LARGE}$ (Single)	84.1	90.9	-	-
$BERT_{LARGE}$ (Ensemble)	85.8	91.8	-	-
$BERT_{LARGE}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$BERT_{LARGE}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

性能评估 (命名实体识别)

- 命名实体识别(Named Entity Recognition, NER)
  - 一个典型的单元标记任务

- CoNLL-2003 NER数据集
  - 包含 20 万个训练单词，每个单词已标注为
    - **人物 Person**
    - **组织 Organization**
    - **地点 Location**
    - **杂项 Miscellaneous**
    - **其他 Other (non-named entity)**

```
Jim    Hen    ##son  was  a  puppet  ##eer
I-PER  I-PER  X      O    O  O       X
```

性能评估 (命名实体识别)

- BERT 优于其他复杂模型结构

System	Dev F1	Test F1
ELMo+BiLSTM+CRF	95.7	92.2
CVT+Multi (Clark et al., 2018)	-	92.6
$\text{BERT}_{\text{BASE}}$	96.4	92.4
$\text{BERT}_{\text{LARGE}}$	**96.6**	**92.8**

- $\text{BERT}_{\text{large}}$与$\text{BERT}_{\text{base}}$性能差异不大
  - 预训练过程引入了 NER 训练数据范围之外的更多知识

为何 BERT 有效?

- **利用大量未标记的高质量数据**
  - 7000 本书籍 + 维基百科（约 33 亿单词）
- **Transformer 中的多头自注意力模块**
  - 对单词之间的关联进行建模
  - 在实例内可并行计算，因此效率较高
- **有效的自监督学习目标**
  - 掩码语言模型（学习单词关联）
  - 下一句预测（学习句子关系）

本节小结

- **Skip-gram**
  - 词向量工具包word2vec中的一种模型结构
  - 基于分布式语义假设，以预测临近单词为目标
  - 使用神经网络实现邻近词概率计算；采用负采样训练策略
- **BERT**
  - 基于Transformer的双向编码器
  - 掩码语言模型 & 下一句预测
  - 支持句对分类、单句分类、单元标记等多种下游任务的微调
- **Skip-gram vs. BERT**
  - 相似点：掩码预测自监督任务；得到单词向量表征为目标；相似应用范式
  - 差异点：静态单词表征 vs. 动态单词表征(考虑上下文影响)

课后思考

1. 如果两个单词是反义词（例如fast和slow），它们的word2vec词向量相似还是不相似？给出你的解释。可以用这个demo网页(https://turbomaze.github.io/word2vecjson/)验证你的想法，网页可以列出和一个单词的词向量相似度最高的10个单词，检查单词的反义词是否会出现在这10个单词中。

2. 在BERT模型预训练过程中，15%的token会被掩码，如果这个掩码比例过高或者过低可能会有什么样的问题？按照你的理解给出简要回答。