

# NLP预训练模型概述



■ 报告人: cooper

■ 时间: 2021/9/11

# 【目 录】

## CONTENTS

第一章 NLP简述

第二章 预训练模型发展

第三章 重点模型

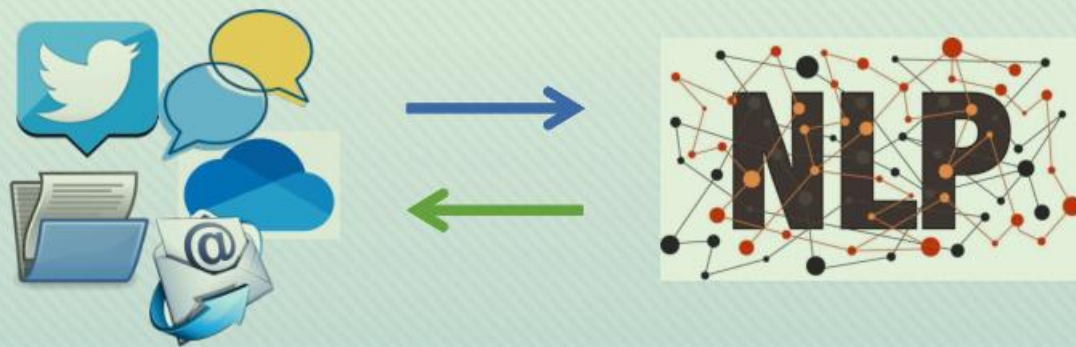
第四章 预训练模型trick

第五章 总结与讨论




# 第一章 NLP简述

- 语言是**思维的载体**，是人类交流思想、表达情感最自然、最方便的工具
  - 人类历史上大部分知识是以语言文字形式记载和流传的
- 自然语言指的是人类语言，特指**文本符号**，而非语音信号
- 自然语言处理（ Natural Language Processing , NLP ）
  - 用计算机来**理解**和**生成**自然语言的各种理论和方法



# 自然语言处理任务分级



## 应用系统 (NLP+)

- 教育, 医疗, 司法, 金融, 机器人等

## 应用任务

- 信息抽取, 情感分析, 机器翻译, 对话系统等

## 基础任务

- 分词, 词性标注, 句法分析, 语义分析等

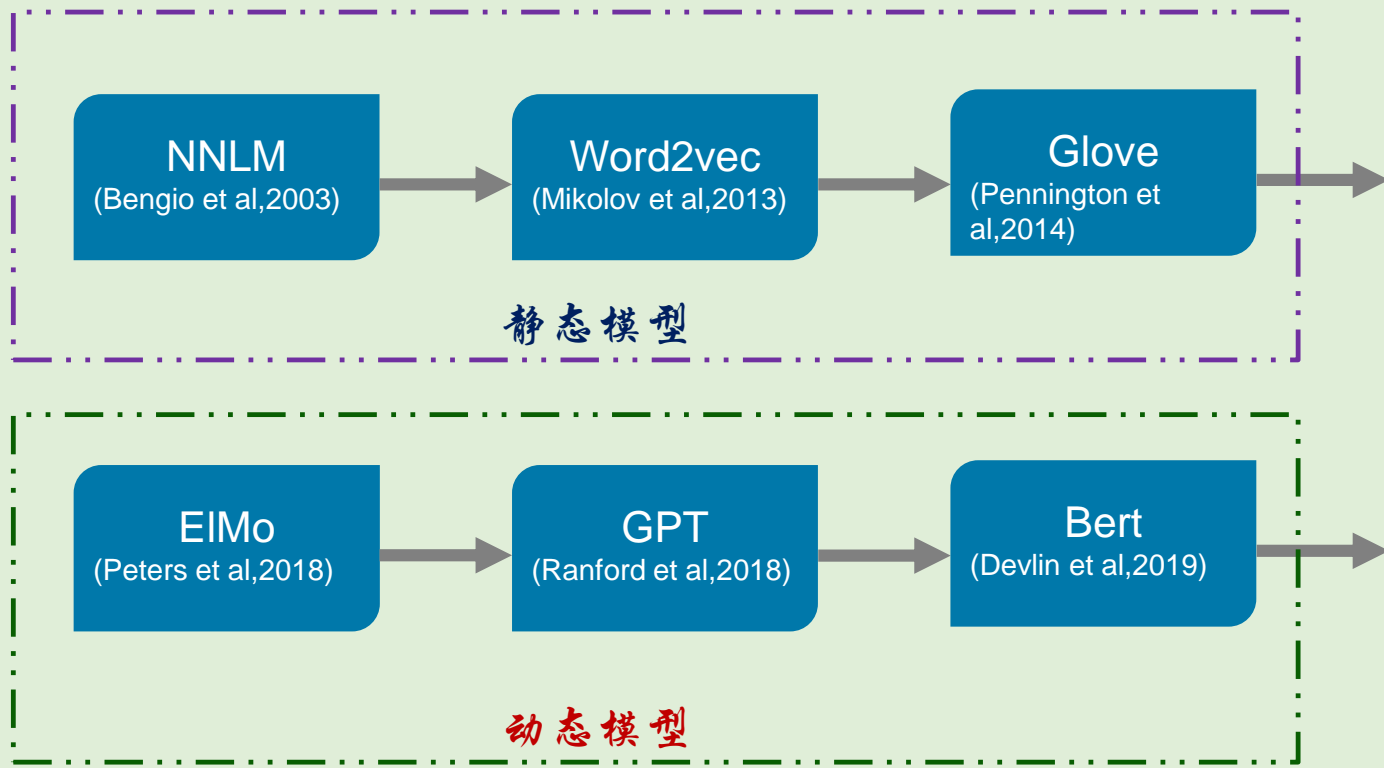
## 资源建设

- 语言学知识库建设, 语料库资源建设等



## 第二章 预训练模型发展

# 预训练模型发展





## 第三章 重点模型



□ <https://code.google.com/archive/p/word2vec/>

□ Mikolov et al., ICLR 2013

□ CBOW (Continuous Bag-of-Word)

□ 周围词向量加和预测中间的词

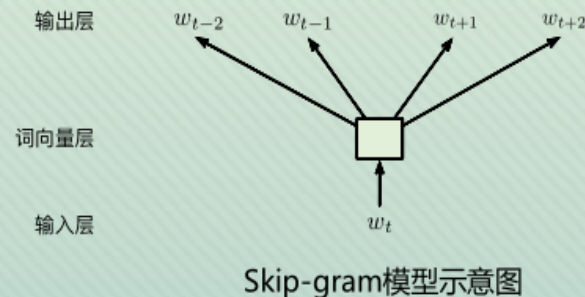
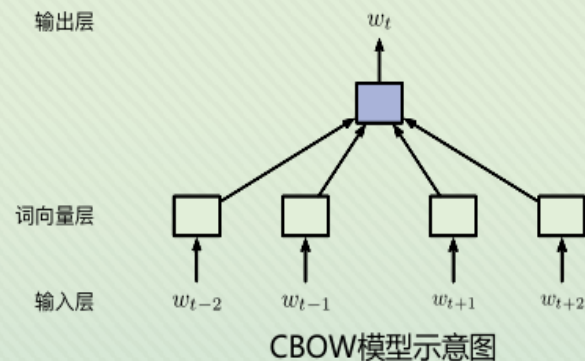
□ Skip-Gram

□ 中间词预测周围词

□ 训练速度快

□ 可利用大规模数据

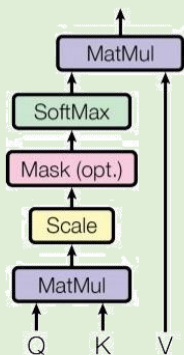
□ 弥补了模型能力的不足



# Transformer

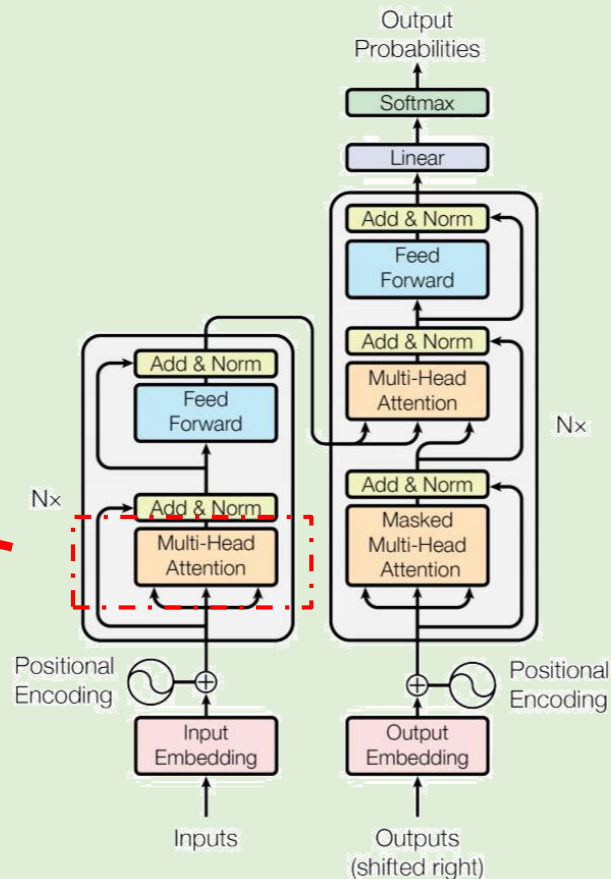
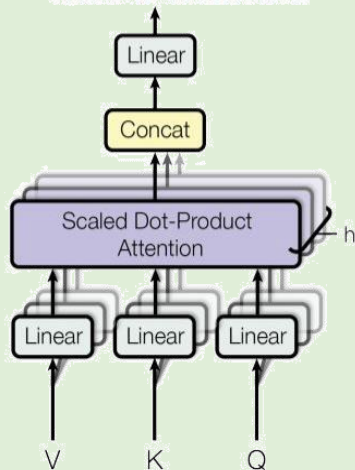
论文地址: <https://arxiv.org/abs/1706.03762>

Scaled Dot-Product Attention



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

Multi-Head Attention



## 整体流程

### Encoder

输入：Word Embedding + Position Embedding + Padding Embedding

计算：self-attention + feed-forward (双向)

### Decoder:

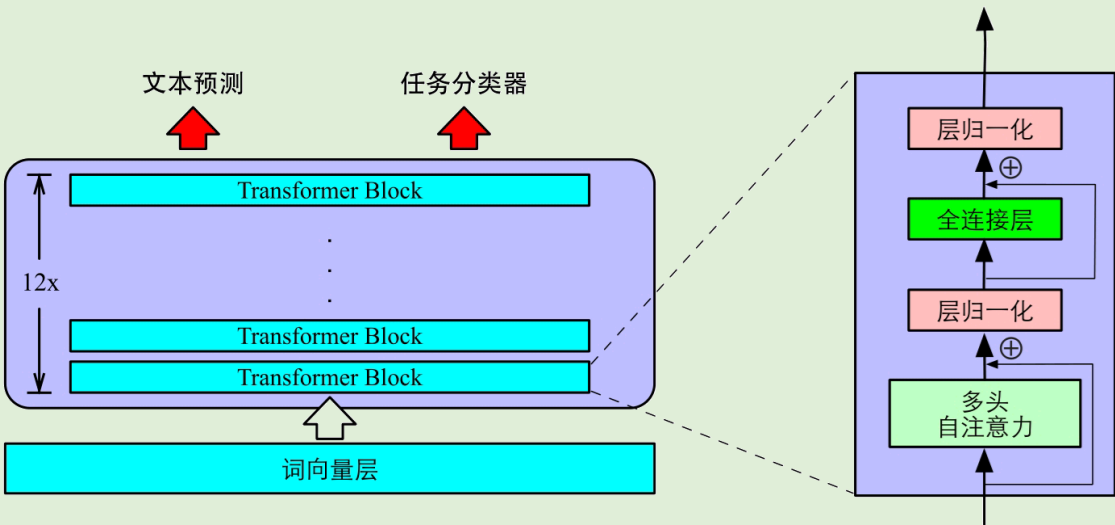
输入：前一时刻Decoder Embedding + Positional Embedding + Padding Embedding  
+ Mask Embedding

中间输入：Encoder Embedding(单向)

输出：预测的词

## 基于transformer的 decoder阶段的单向模型

自回归模型，即根据概率分布依次生成整个句子或者进行推理。



论文地址:<https://openai.com/blog/language-unsupervised/>

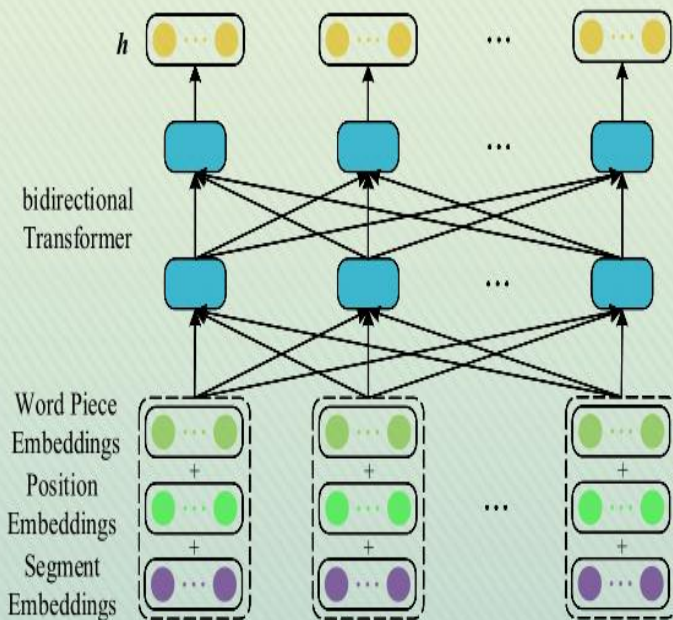
模型	发布时间	参数量	预训练数据量
GPT	2018 年 6 月	1.17 亿	约 5GB
GPT-2	2019 年 2 月	15 亿	40GB
GPT-3	2020 年 5 月	1,750 亿	45TB

论文地址:<https://arxiv.org/abs/1810.04805>

基于transformer的  
encode双向模式。

自编码的方式，即通过  
一个神经网络的输入转  
换成特征，然后再通过  
**decoder**把特征恢复成  
原始的信号。

## □ BERT: **Bidirectional** Encoder Representations from Transformers

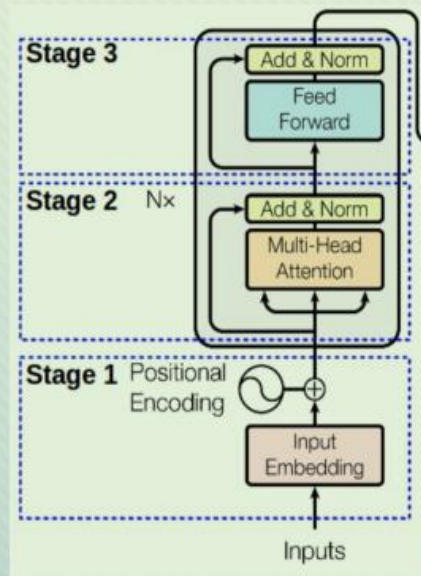




# Bert模型详解

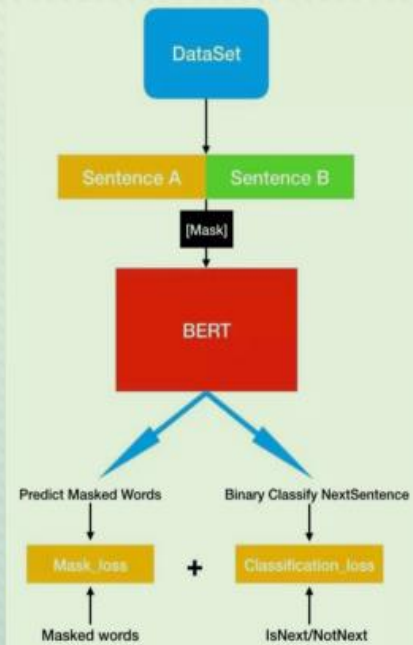
## 编码器

- 输入：Word Piece
- 编码器：Transformer



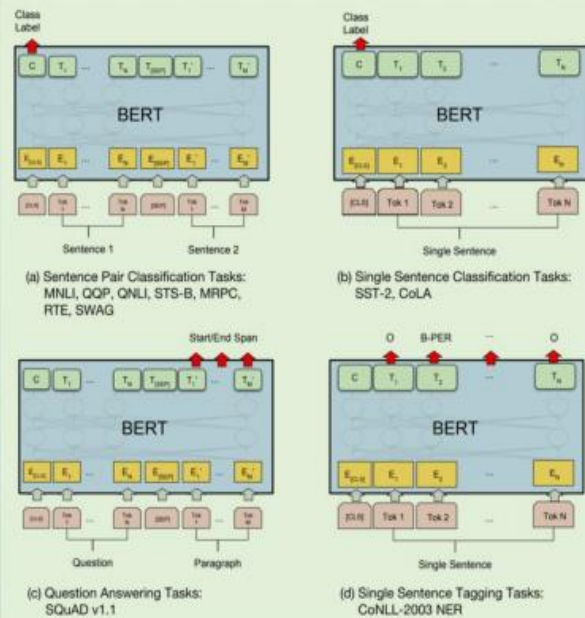
## 预训练任务

- 完形填空 + 下句预测 (NSP)



## 应用方式

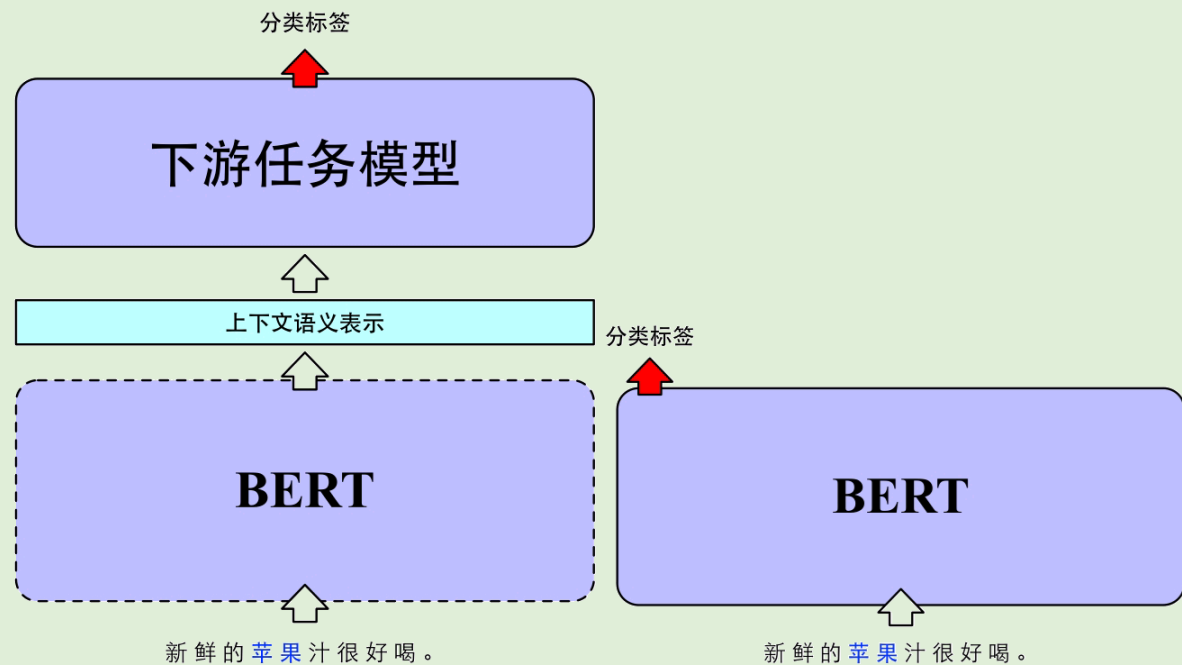
- 在目标任务上Fine-tune
- 四种任务类型



# Bert模型详解

## 两种方式

- ◆ 直接作为输入特征用到模型里面，然后进行训练。
- ◆ 直接对预训练模型进行微调。





## 第四章 预训练模型trick



# 模型的一些trick

1. 可以使用简单的模型，就使用简单的模型进行训练，比如我做文本分类，其实TextCNN的效果已经非常好了，不需要使用Bert等重量级的模型；
2. Bert针对长文本（长度大于512的），根据后面参考文献2，作者的研究，head+tail的组合方式相对而言效果最好。
3. 在中文的任务中，有很多的预训练模型可以起到非常好的效果，比如我在做我们公司的情感分析的时候，使用搜狗的预训练模型，可以达到比其他预训练模型更好地效果。可以查看参考文献4.
4. 调参的一些技巧可以查看参考文献3.我这边强调的是，数据确实最重要的原料，只要数据质量比较高的话，其实，选用什么模型以及模型的调优相对空间不大。



## 第五章 总结与讨论

- 1.随着理论与硬件，数据量的发展，预训练模型越来越大，越来越好用，基本上成为自然语言处理的一个**baseline**或者不二选择。
2. 形成新的训练范式：预训练+精调
- 3.预训练模型还在朝着参数量更大，训练数据更多，硬件更强的方向发展，整体上，利大于弊，很可能会出现大一统的一个模型。
- 4.作为一名打工人，可以解决业务问题的模型，就是最好的模型。不会为了使用模型而用，即没有那么多“奇技淫巧”，适合自己的就是最好的。

[1]史上最小白之Transformer详解

[<https://blog.csdn.net/Tink1995/article/details/105080033/>]

[2] Bert如何针对长文本处理[<https://arxiv.org/pdf/1905.05583.pdf>]

[3]调参技巧[<https://mp.weixin.qq.com/s/jlHQflrC0lfZWBebwP-p4g>]

[4] 中文词向量[<https://github.com/Embedding/Chinese-Word-Vectors>]

[5]自然语言处理，基于预训练模型的方法

[<https://book.douban.com/subject/35531447/>]

[6]预训练模型大综述[<https://arxiv.org/abs/2106.07139>]

# 感谢聆听

■ 报告人: cooper

■ 时间: 2021/9/11