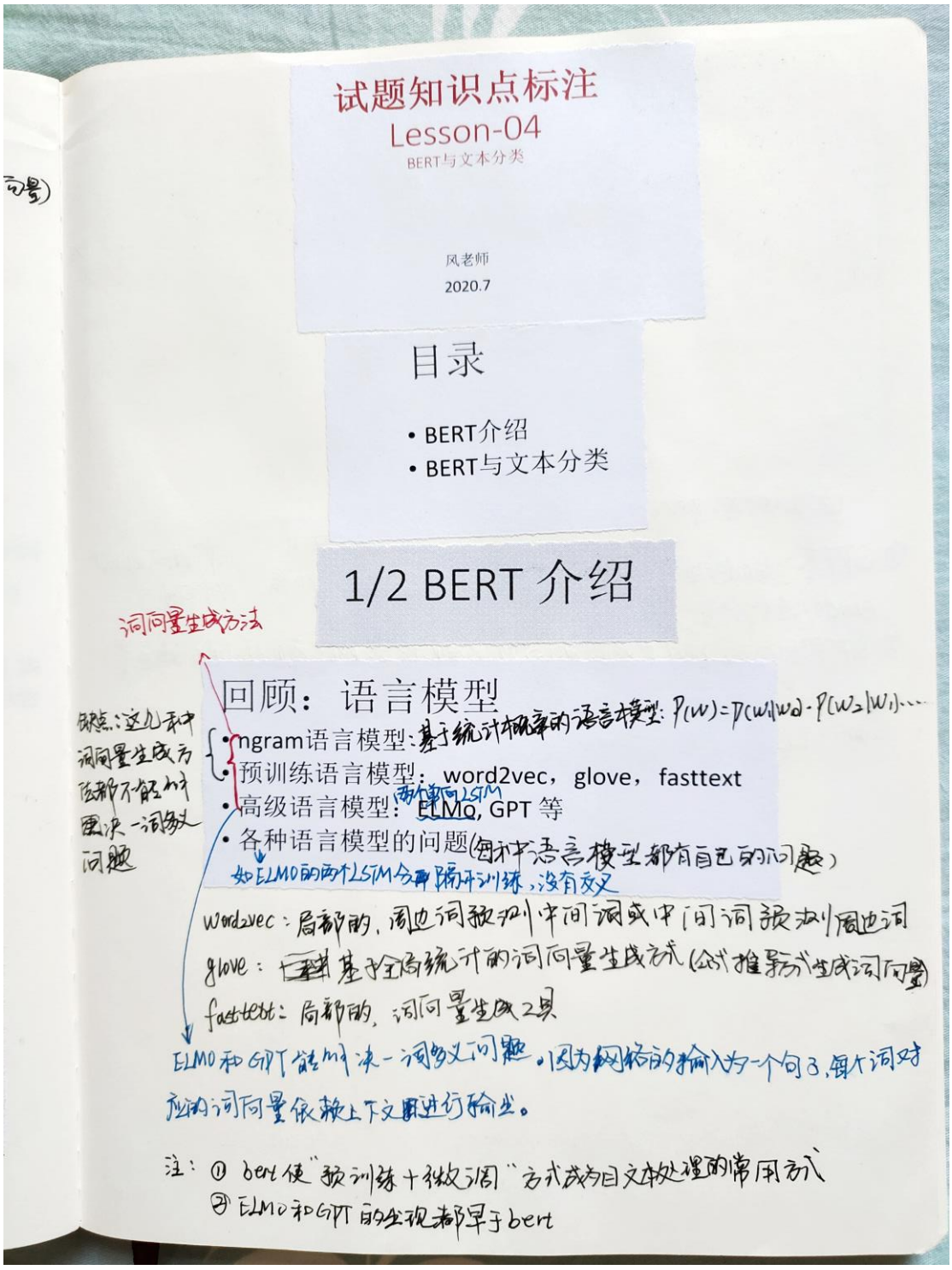
第十章 试题知识点多标签分类-BERT



回顾：transformer

- Encoder-Decoder结构
- Self-Attention：计算句子中词的权重
- Multi-head attention：增加模型对关注点的（表达更好的依赖）
- Positional Encoding
- 解码器Self-Attention输入层与编码器区别：只允许关注输出序列中较前的位置（mask遮罩）

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

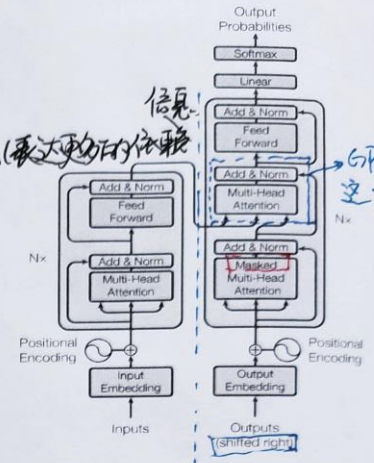


Figure 1: The Transformer - model architecture.

位置编码，保持句子的位置关系，形成位置向量，加入到embedding中。

① GPT基于decoder进行计算，bert基于encoder进行计算。

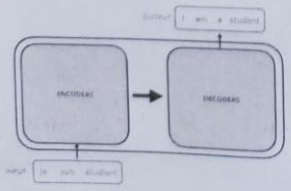
bert使用encoder的结构

GPT使用decoder的结构

② GPT是单向的，bert是双向的 => 你能看见我，我也能看见你（因为没有遮罩）

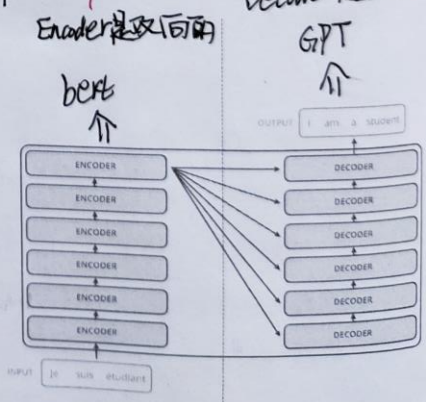
GPT是单向的：你可以看见我，我不能看见你（后面可以看见前面的信息，前面不能看见后面的）

Encoder-Decoder结构拆解



Encoder是双向的 bert

Decoder是单向的 GPT



BERT概述

基于transformer的双向预训练语言模型.

- BERT: 《Pre-traning of Deep Bidirectional Transformers for Language Understanding（Google AI Language）》
- BERT： Bidirectional Encoder Representations from Transformer
- GLUE基准80.04%（7.6%绝对提升），MultiNLI准确率86.7%（5.6%绝对提升）
- https://gluebenchmark.com/leaderboard

评估自然语言处理任务效果的网站.

BERT的贡献

bert的比较标准为GPT，GPT为单向的.

- 证明了双向模型对文本特征表示的重要性
- 证明了预训练模型能够消除很多繁重的任务相关的网络结构
- 在11个NLP任务上，提升了state of the art水平

增加各种中层.

采用 预训练+微调.

BERT特点

如"苹果手机"为"吃苹果"中"苹果"的词向量是相同的，[0.1, 0.2, 0.3, ...]为同一个值.

- 与其他词向量的关系：
  - Word2vec等词向量是词维度，训练好就确定了。
  - BERT句子维度的向量表示，依赖上下文构建结果。
    - 苹果 "苹果"的词向量不相同
    - bank
  - 都是无监督训练，不需要标签信息

bert和word2vec

不需要标签信息

通过无监督训练方式定义了一种有监督任务.

如用周围词预测中间词

000只∞

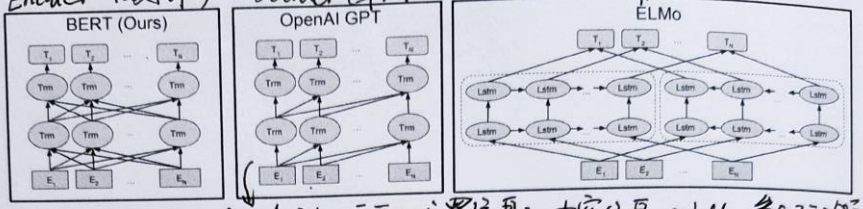
BERT 模型架构

能够互看见.

Encoder (双向)　Decoder (单向)

向量

为下游任务输出特征(feature)



Shifted right: 自身的位置信息和内容信息(embedding)参与了计算

Figure 3: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTMs to generate features for downstream tasks. Among the three, only BERT representations are jointly conditioned on both left and right context in all layers. In addition to the architecture differences, BERT and OpenAI GPT are fine-tuning approaches, while ELMo is a feature-based approach.

微调模型的权重

词向量不同. 两个"苹果"对应的词向量不同.

词向量是固定的. 如"苹果手机"与"吃苹果"中苹果的词向量是相同的

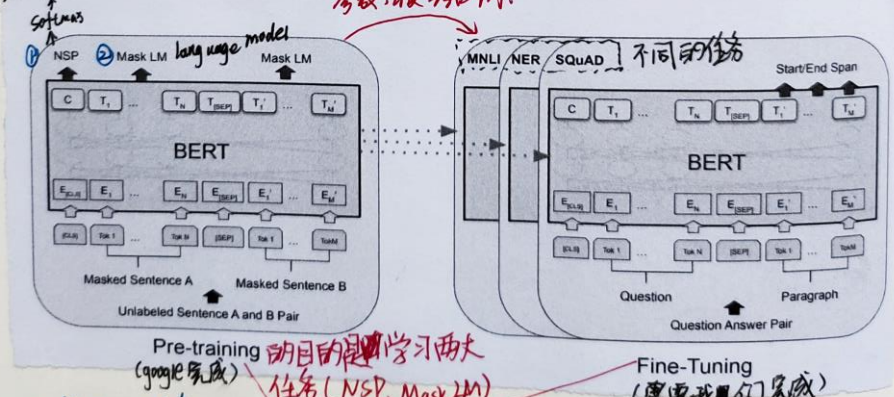
BERT 核心思想

预训练和微调是两个不同的任务, 有自己不同的目标

Next Sentence prediction 下一句预测

softmax

参数. 模型结构.



两个目的是同时学习两大任务(NSP, Mask LM)

预训练两大任务:

① 预测下一个句子 (句子B是不是句子A的下一个句子)

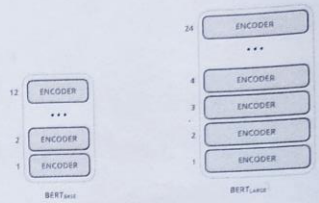
② 预测被 mask 的词.

微调中没有 mask

模型架构, 参数是一致的, 输入和输出不同

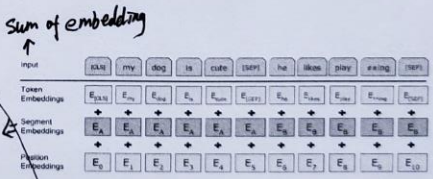
BERT

- 特点：跨越不同任务的统一架构
  - 预训练架构与最终下游架构之间的差异很小。
- 参数：
  - BERT_BASE（L = 12，H = 768，A = 12，总参数= 110M）
  - BERT_LARGE（L = 24，H = 1024，A = 16，总参数= 340M）
  - In all cases we set the feed-forward/filter size to be 4H, i.e., 3072 for the H = 768 and 4096 for the H = 1024.



BERT

- 输入&输出：
  - 单句或句对组合
  - WordPiece embeddings（英文）
  - 单字拆分（中文）
  - [CLS]：起始标记
  - [SEP]：句对分割标记



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

Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

BERT pre-train(Task 1): Masked LM

- 从左到右训练 or 从右到左训练
- BiLSTM等双向训练的问题
- masked LM：随机MASK 15% 的word piece（存在的问题？）
  - （1）[MASK]token替换（80%）
  - （2）随机token替换（10%）
  - （3）不变（10%）

预训练第一个任务

北京是中国的首都

ELMO暴露未来信息的问题

句子中词先后组合的关系

（预测单前词）

预训练的任务是预测被mask的词。（句子级别的任务。类似word2vec

mask是遮罩自己，shifted right是遮罩后面的。

BERT pre-train(Task 2)： Next Sentence Prediction (NSP)

第2个任务

下一句预测：句子与句子的关系

- 目的：问答，推理等句子对之间的关系
- 训练数据构建：句子A + 50%真实句子B + 50%随机句子
  - 50%：B是跟随A的实际下一个句子（标记为IsNext）
  - 50%：来自语料库的随机句子（标记为NotNext）
- 对比：
  - BERT：传输所有参数初始化最终任务模型参数
  - 其他：句子嵌入被转移到下游任务

将预训练后的embedding模型的结构和参数都给下游任务（微调）

- The final model achieves 97%-98% accuracy on NSP.
- The vector C is not a meaningful sentence representation without fine-tuning, since it was trained with NSP.

将预训练模型输出的feature给给下游任务

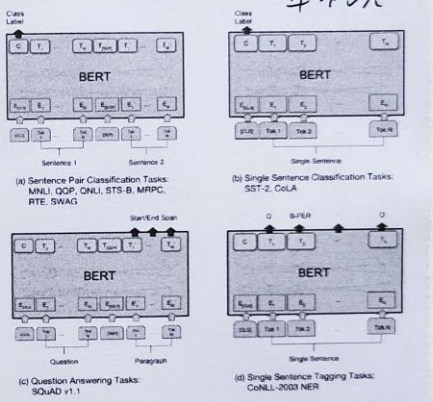
预训练数据：

- 英文：
  - BooksCorpus（800M words）
  - 英语维基百科（2,500M字）提取文本段落，忽略列表，表格和标题。
- 汉语
  - 中文维基百科

Fine-tuning：需要微调的原因是因为预训练任务和下游任务不一致。

- Transformer中的自动注意机制允许BERT通过交换适当的输入和输出来模拟许多下游任务
- 将任务特定的输入和输出插入到BERT中，并对端到端的所有参数进行微调。
- 为什么要微调？
- 参数：
  - Batch size: 16, 32
  - Learning rate (Adam): 5e-5, 3e-5, 2e-5
  - Number of epochs: 2, 3, 4

单句分类



(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

微调是一个有监督学习过程。

Fine-tuning的效果也feature好。

BERT & GPT：对比

- Encoder VS. Decoder
- 更多的数据:BooksCorpus and Wikipedia VS. BooksCorpus
- 在预训练中采用 SEP CLS（GPT： Fine-tuning）
- 每个batch size词用的更多
- (5e-5, 4e-5, 3e-5, and 2e-5) VS. 5e-5

在生成式问题中，单向模型比双向的好。原生bert不适合生成式问题

BERT 每一层学到了什么：

- ACL 2019：What does BERT learn about the structure of language?
- https://hal.inria.fr/hal-02131630/document
- 低层网络捕捉了短语级别的结构信息
- 表层信息特征在底层网络（3，4），句法信息特征在中间层网络（6~9），语义信息特征在高层网络。（9~12）
- 主谓一致表现在中间层网络（8，9）

BERT 变体：

- ROBERTA bert的mask是固定的.
  - 静态mask->动态mask
  - 去除句对NSP任务，改为输入连续多个句子
  - 更多数据 更大batch size 更长时间 训练时间更长

$A_{100} + B_{200} + 00000_{212}$ 改为 $A+B+C+D+\cdots$ (512)

句对占用的空间是有限的，多个句子可以占满整个空间 (512)

- ALBERT
  - 减少参数（瘦身）：原：V×H → V×E，E×H
    - 在词表 V 到隐层 H 的中间，插入一个小维度 E
    - 共享所有层的参数：Attention FFN 所有层的Attention参数是一致的
  - SOP 替换 NSP：负样本换成了同一篇文章中的两个逆序的句子
  - BERT对MASK 15% 的词来预测。ALBERT 预测的是 n-gram 片段，包含更完整的语义信息 词组
  - 增加训练数据长度：90%取512 BERT90% 128
  - 对应BERT large：H:1024 ->4096 L:24->12 窄而深->宽而浅
- 百度 ERNIE 引入先验信息

ALBERT训练中 90% 采用512的数据量，10%采用 128.

FFN:
Feed Forward Network

SOP: Sentence-order prediction 句子顺序预测

句子A. 句子B
正样本: A+B
负样本: A+Random
} => { A+B / B+A

2/2 BERT与文本分类

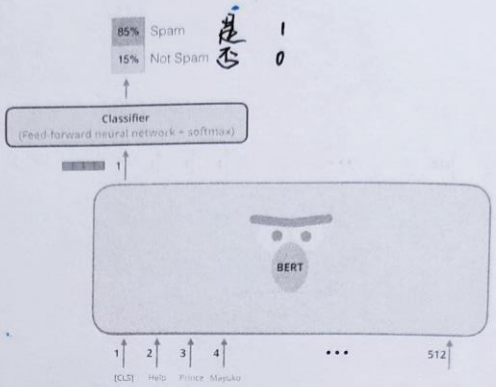
BERT可以解决的问题

- 序列标注
- 分类任务
- 句子关系A与句子B是否相似.
- 生成任务（表现不好，因为bert的训练中没有顺序生成的概念，没有前后预测的事情）

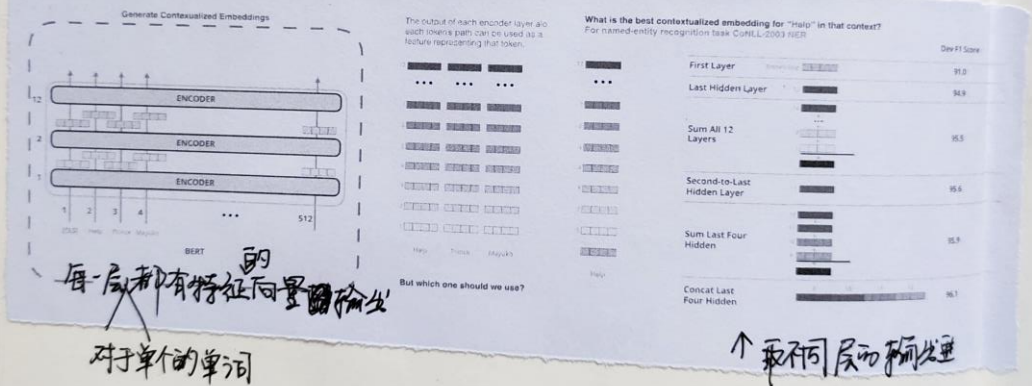
对于生成式问题，建议采用transformer的seq2seq框架。

Bert文本分类方法

从结构上看
bert进行文本分类的方式



从架构上看



每一层都有特征向量的输出

对于单个的单词

↑取不同层的输出进行组合测试.

文本分类方法1：特征抽取+其他模型

特征抽取流程①

feature-based

对词向量加权求和得到

Step #1: Use DistilBERT to embed all the sentences

Sentence embeddings

	Sentence	label
0	a stirring , funny and finally transporting re imagining of beauty and the beast and 1930s	1
1	apparently reassembled from the cutting room floor of any given daytime soap	0
..	...	...
1,999	the movie is undone by a filmmaking methodology that 's just experimental enough	1

Bert

DistilBERT

Already (pre-)trained

	Sentence Embeddings				label
	0	1	..	767	
0	-0.215	-0.1402	..	0.201	1
1	-0.172	-0.144	..	0.371	0
..	..	..	..	..	..
1,999	0.124	0.014	..	0.274	1

特征抽取

接其他模型 如LogReg 等其他模型

文本分类方法1：特征抽取+其他模型

特征抽取流程②

train/test流程.

Step #2: Test/Train Split for model #2, logistic regression

	Sentence Embeddings				label
	0	1	..	767	
0	-0.215	-0.1402	..	0.201	1
1					
2					
3					
4					
5					
6					
7					
8					
9					
..					
1,999	0.124	0.014	..	0.274	1

Training set
75% of examples

Testing set
25% of examples

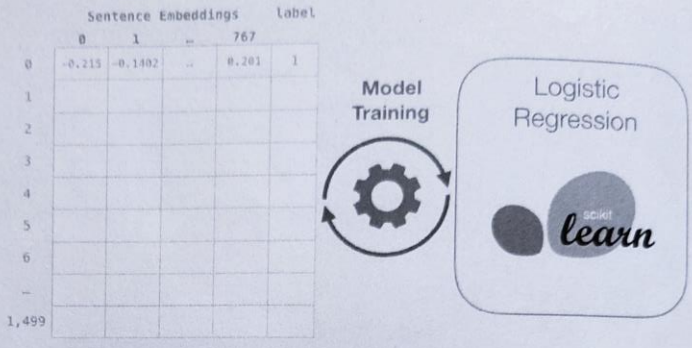
	Sentence Embeddings				label
	0	1	..	767	
0	-0.215	-0.1402	..	0.201	1
1					
2					
3					
4					
5					
6					
..					
1,499					

	Sentence Embeddings				label
	0	1	..	767	
1,500					
..					
1,999	0.124	0.014	..	0.224	1

文本分类方法1：特征抽取+其他模型

特征抽取流程③

Step #3: Train the logistic regression model using the training set



文本分类方法2：bert fine-tuning

微调流程

- 修改run_classifier.py
  - 构建自己的processer 继承DataProcessor
  - processors 增加新构建的DataProcessor
  - Run!

注：bert的作用是预训练语言模型，通过海量数据进行学习，

解决了标注样本少的问题。

附：

（1）attention 综述：https://zhuanlan.zhihu.com/p/62136754
（2）bert 简介：https://zhuanlan.zhihu.com/p/92849070
（3）自认语言处理任务评价网站：https://gluebenchmark.com/leaderboard
（4）bert 源码 github：https://github.com/google-research/bert
（5）bert-as-service 包：https://github.com/hanxiao/bert-as-service
（6）bert 论文：https://arxiv.org/pdf/1810.04805.pdf
（7）bert 解读博客：https://blog.csdn.net/weixin_42001089/article/details/97657149
（8）ai 竞赛开放平台：https://www.flyai.com/
（9）图神经网络工具：https://github.com/thunlp/OpenKE