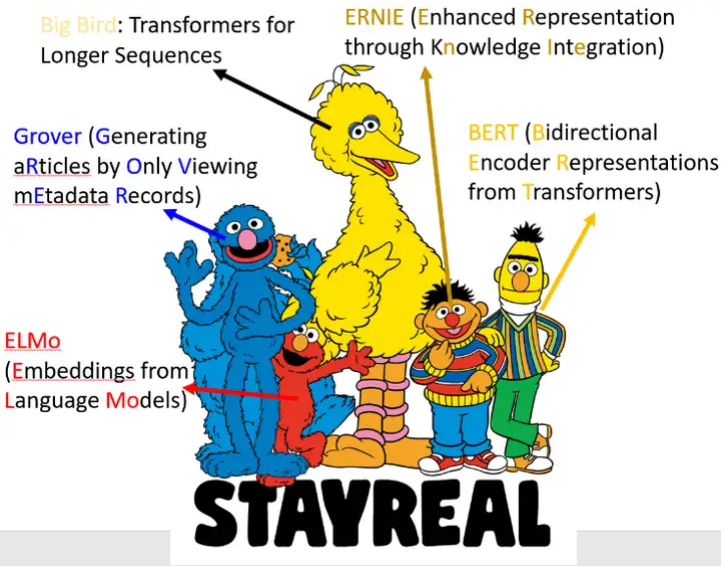
12 Self-Supervised Learning(BERT)

12.1 BERT家族

BERT family



参数量

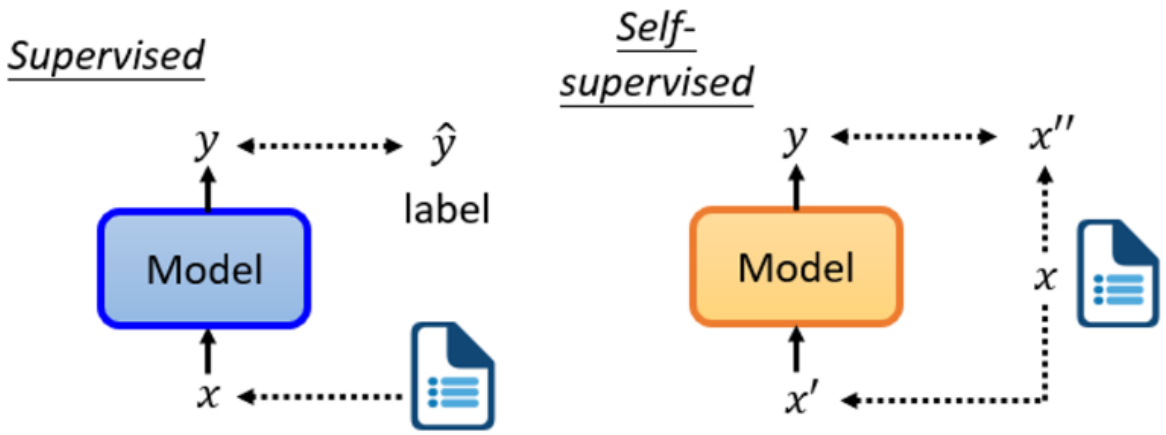
- ELMO：94M
- BERT：340M
- GPT-2：1542M
- Megatron：8B
- T5：11B
- Turing NLG：17B
- GPT-3：175B
- Switch Transformer：1.6T

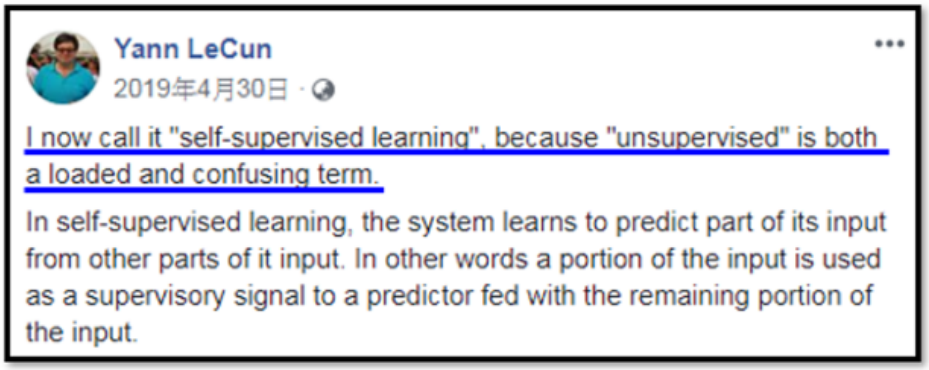
12.2 Self-supervised Learning简介

⇒Unsupervised Learning的一种

“自监督学习”数据本身没有标签，所以属于无监督学习；但是训练过程中实际上“有标签”，标签是“自己生成的”。

想办法把训练数据分为“两部分”，一部分作为作为“输入数据、另一部分作为“标注”。





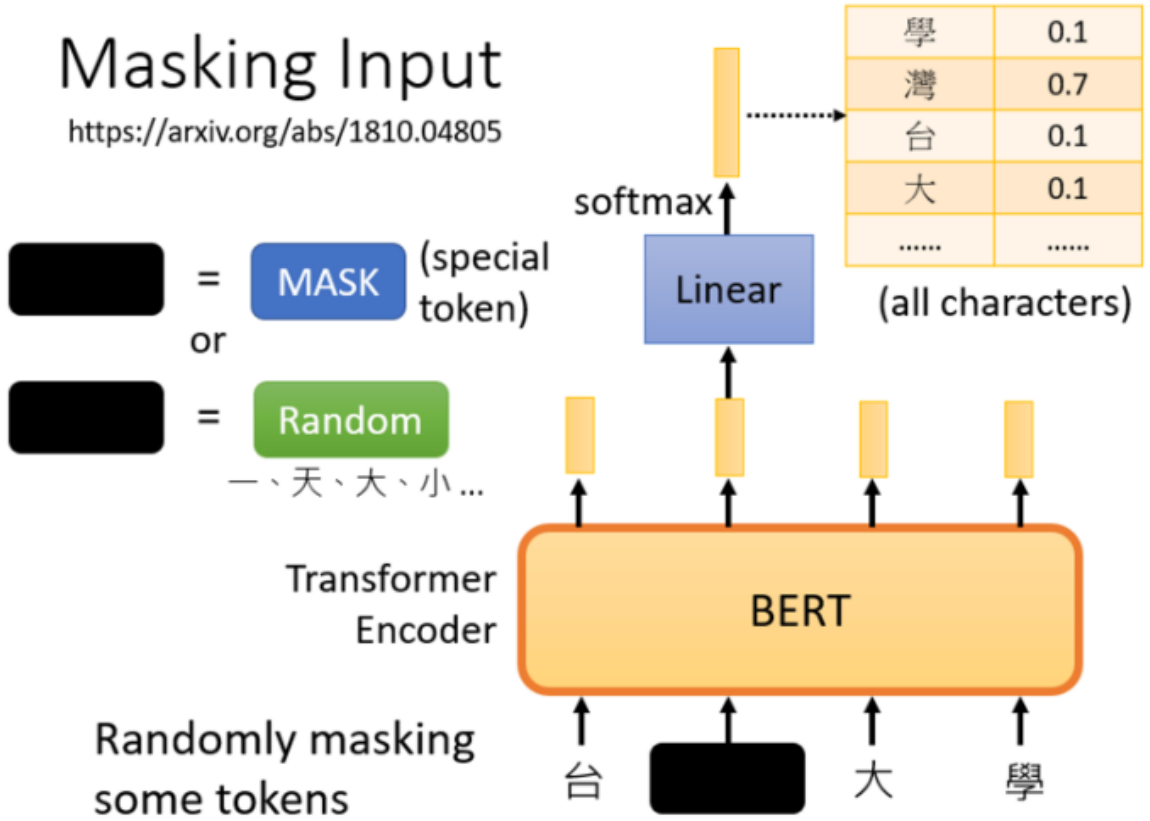
12.3 BERT简介

作为transformer，理论上BERT的输入长度没有限制。但是为了避免过大的计算代价，在实践中并不能输入太长的序列。 事实上，在训练中，会将文章截成片段输入BERT进行训练，而不是使用整篇文章，避免距离过长的问题。

BERT是一个transformer的Encoder，BERT可以输入一行向量，然后输出另一行向量，输出的长度与输入的长度相同。BERT一般用于自然语言处理，一般来说，它的输入是一串文本。当然，也可以输入语音、图像等"序列"。

12.3.1 Masking Input

随机盖住一些输入的文字，被mask的部分是随机决定的



MASK的方法

- 第一种方法是，用一个**特殊的符号替换句子中的一个词**，我们用 **"MASK "标**记来表示这个特殊符号，你可以把它看作一个新字，这个字完全是一个新词，它不在你的字典里，这意味着mask了原文。
- 另外一种方法，**随机**把某一个字**换成另一个字。**中文的 "湾"字被放在这里，然后你可以选择另一个中文字来替换它，它可以变成 "一 "字，变成 "天 "字，变成 "大 "字，或者变成 "小 "字，我们只是用随机选择的某个字来替换它

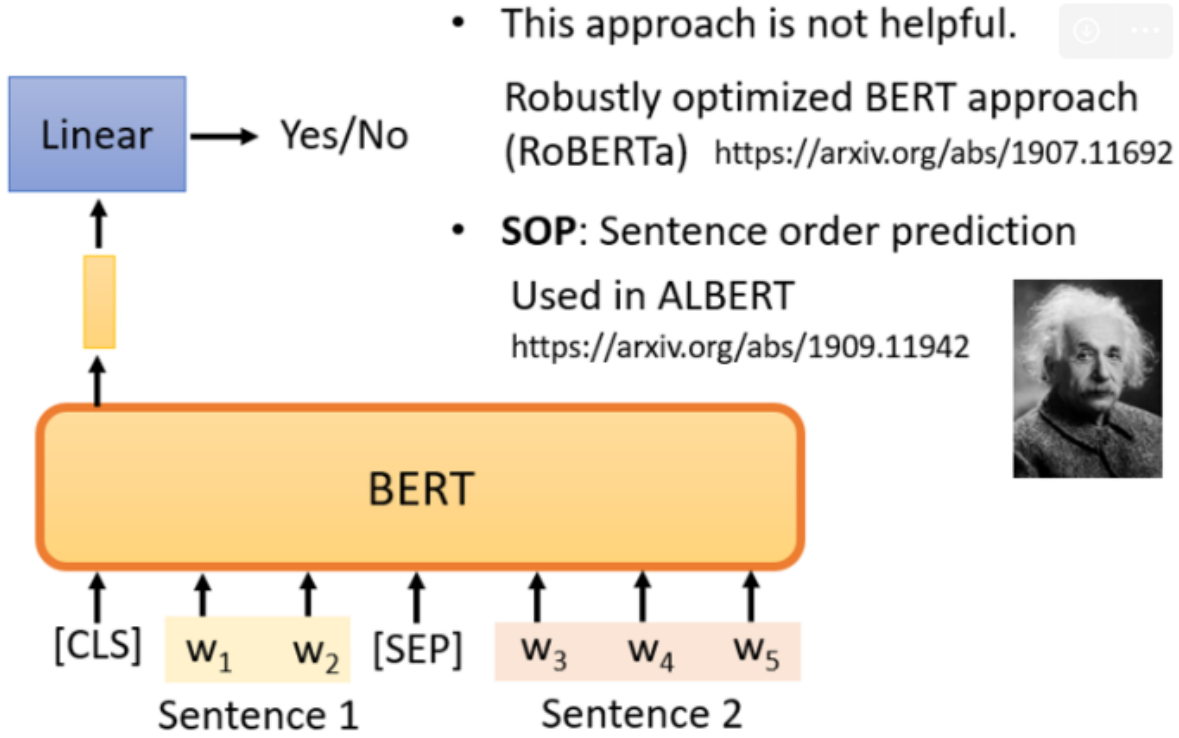
两种方法**都可以使用，使用哪种方法也是随机决定的。**

训练方法

1. 向BERT输入一个句子，先随机决定哪一部分的汉字将被mask。
2. 输入一个序列，我们把BERT的相应输出看作是另一个序列
3. 在输入序列中寻找mask部分的相应输出，将这个向量通过一个Linear transform（矩阵相乘），并做Softmax得到一个分布。
4. 用一个one-hot vector来表示MASK的字符，并使输出和one-hot vector之间的交叉熵损失最小。

本质上，就是在解决一个**分类问题。**BERT要做的是**预测什么被盖住。**

12.3.2 Next Sentence Prediction（不太有用）



从数据库中拿出两个句子，两个句子之间添加一个特殊标记[SEP]，在句子的开头添加一个特殊标记[cls]。这样，BERT就可以知道，这两个句子是不同的句子。

只看CLS的输出，我们将把它乘以一个Linear transform,做一个二分类问题，输出yes/no，预测两句是否前后连续。

没有用

Robustly Optimized BERT Approach(RoBERTa)

12.3.3 Sentence order prediction，SOP(句子顺序预测)⇒ALBERT

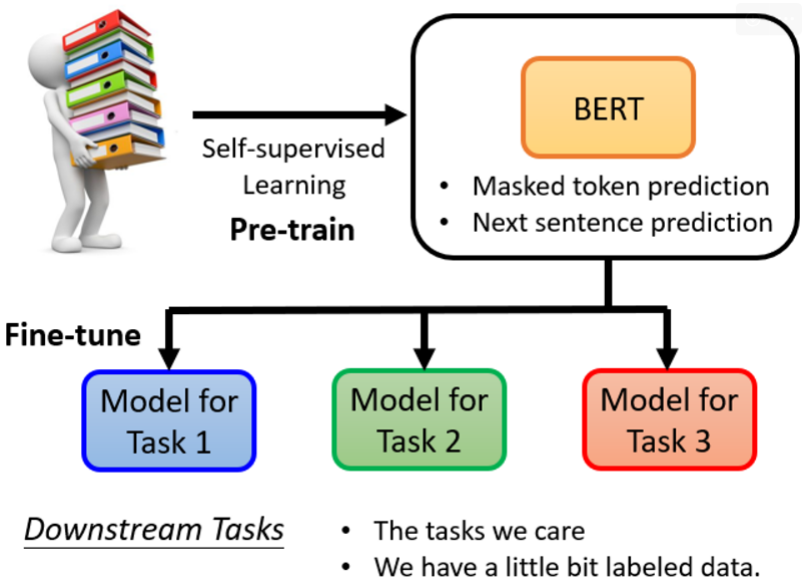
挑选的两个句子是相连的。可能有两种可能性供BERT猜测：

- 句子1在句子2后面相连，
- 句子2在句子1后面相连。

12.3.4 BERT的实际用途⇒下游任务（Downstream Tasks）

预训练与微调

- 预训练：产生BERT的过程
- 微调：利用一些特别的信息，使BERT能够完成某种任务



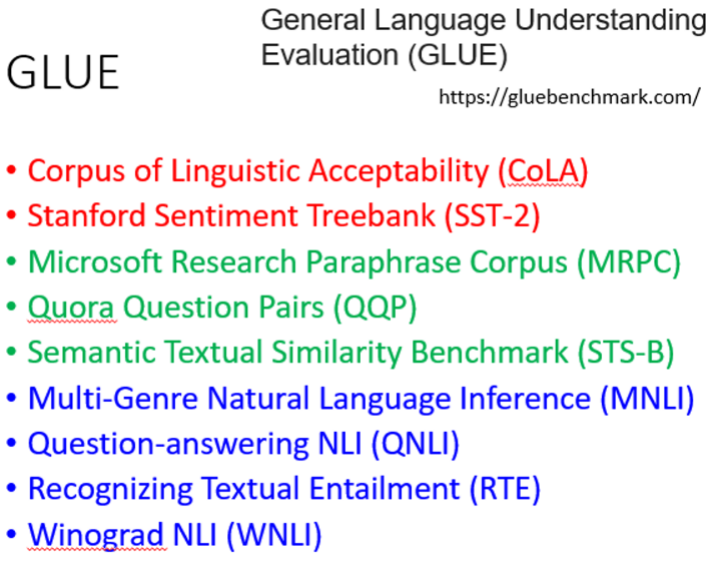
BERT只学习了两个“填空”任务。

- 一个是掩盖一些字符，然后要求它填补缺失的字符。
- 预测两个句子是否有顺序关系。

但是，BERT可以被应用在其他的任务【真正想要应用的任务】上，可能与“填空”并无关系甚至完全不同。【胚胎干细胞】当我们想让BERT学习做这些任务时，只**需要一些标记的信息，就能够“激发潜能”**。

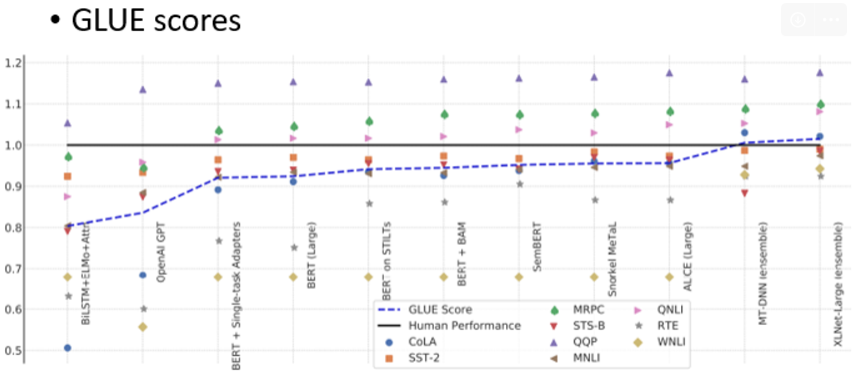
对BERT的评价任务集——GLUE（General Language Understanding Evaluation）

为了测试Self-supervised学习的能力，通常，你会在一个任务集上测试它的准确性，取其平均值得到总分。



性能衡量

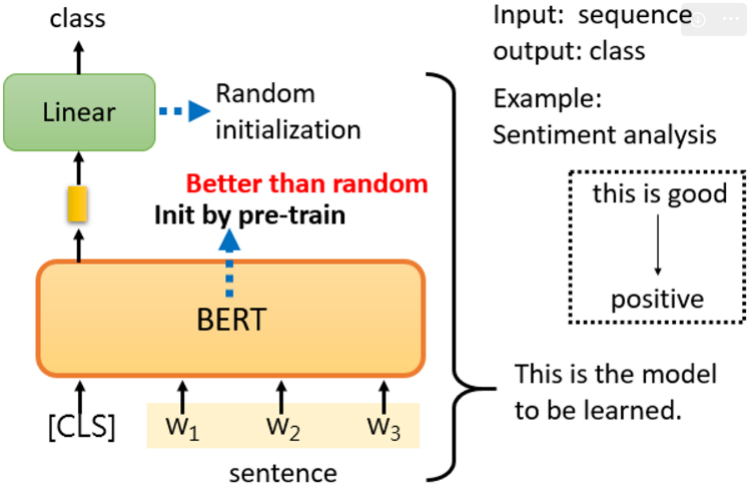
人类的准确度是1，如果他们比人类好，这些点的值就会大于1。



Source of image: https://arxiv.org/abs/1905.00537

12.3.5 How to use BERT——[CLS]+Fine Tune

Case 1：Sentiment analysis：给机器一个句子，让它判断这个句子是正面的还是负面的。



给它一个句子，把CLS标记放在这个句子的前面，只看CLS的部分。CLS在这里输出一个向量，我们对它进行Linear transform+Softmax，得到类别。

对下游任务，需要标注资料。

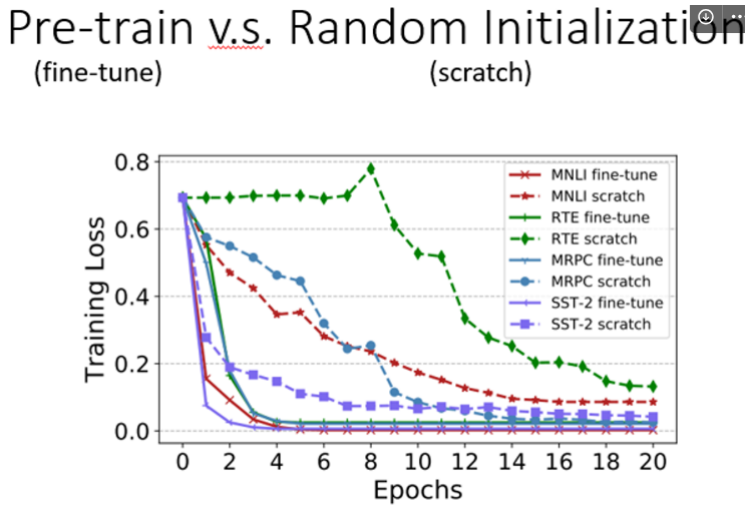
在训练的时候，Linear transform和BERT模型都是利用Gradient descent来更新参数的。

- Linear transform的参数是**随机初始化**的
- 而BERT的参数是由**学会填空的BERT初始化**的。⇒将获得比随机初始化BERT更好的性能。

对比预训练与随机初始化

- "fine-tune"是指模型被用于预训练，这是网络的BERT部分。该部分的参数是由学习到的BERT的参数来初始化的，以填补空白。
- scratch表示整个模型，包括BERT和Encoder部分都是随机初始化的。

scratch与用学习填空的BERT初始化的网络相比，损失**下降得比较慢**，最后，用随机初始化参数的网络的**损失高**于用学习填空的BERT初始化的参数。

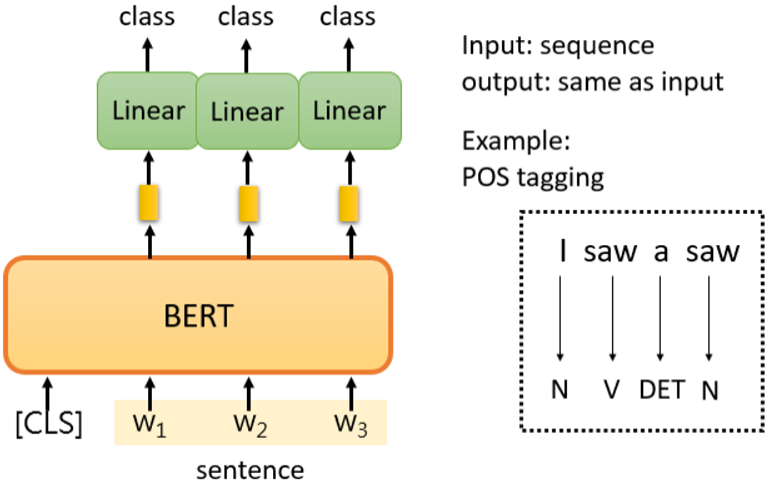


使用BERT的整个过程是连续应用Pre-Train+Fine-Tune，它可以被视为一种**半监督方法(semi-supervised learning)**

- 当你进行Self-supervised学习时，你使用了大量的**无标记数据⇒unsupervised learning**
- Downstream Tasks 需要少量的**标记数据**。

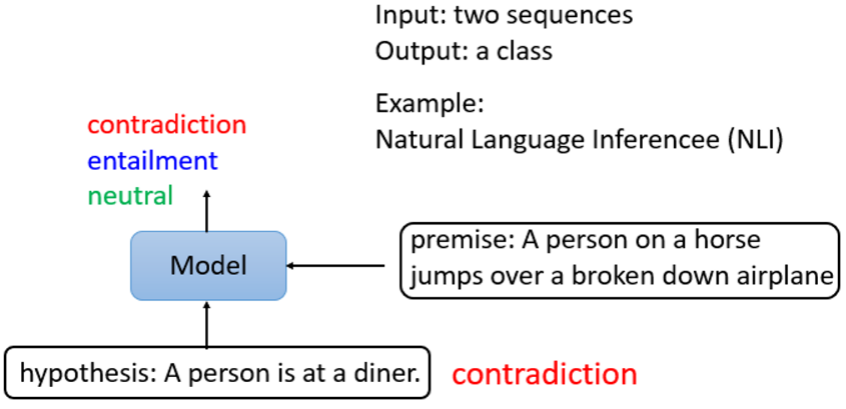
Case 2： POS tagging

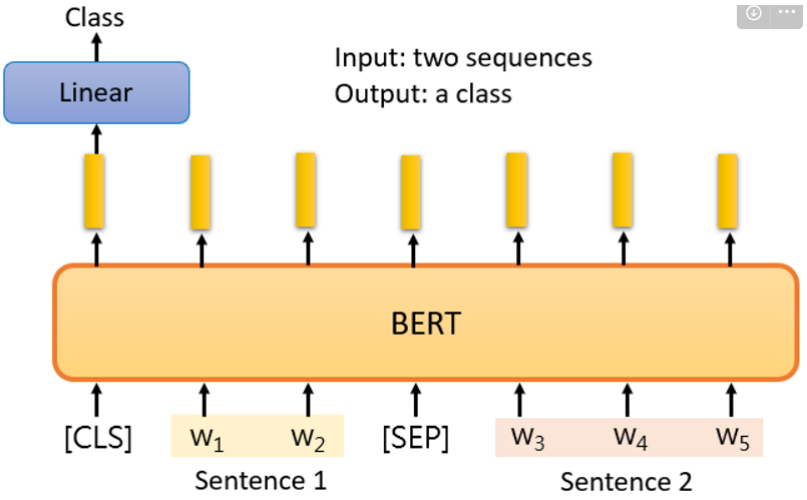
BERT部分，即网络的Encoder部分，其参数不是随机初始化的。在预训练过程中，它已经找到了不错的参数。



Case 3： Natural Language Inference(NLI)

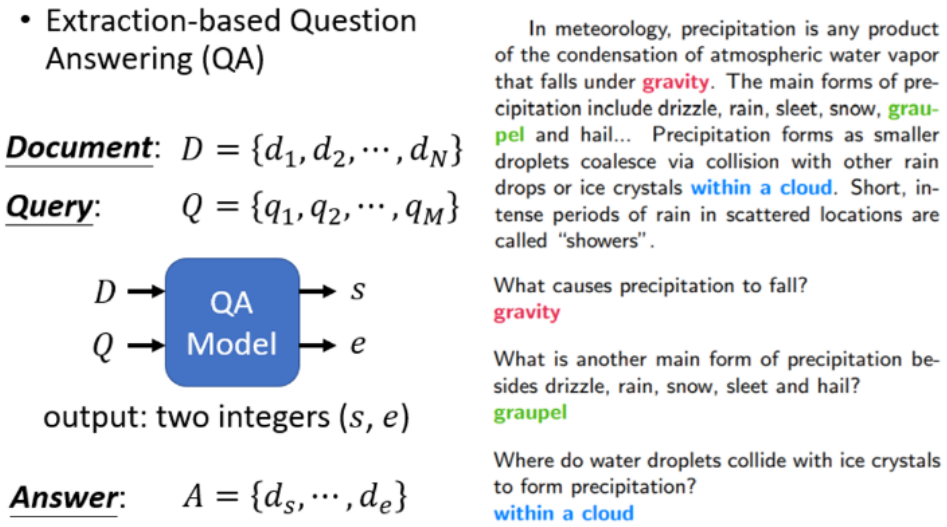
给出前提和假设，机器要做的是判断，是否有可能从前提中推断出假设。⇒预测"赞成、反对"





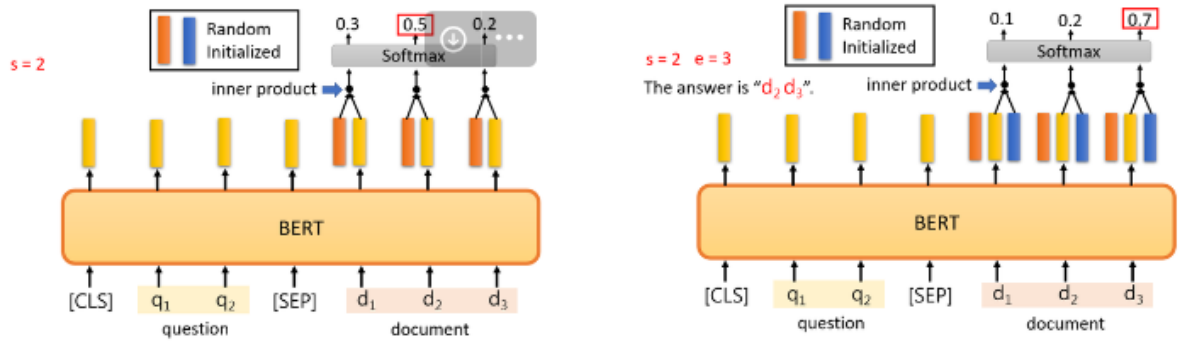
你只要给它两个句子，我们在这两个句子之间放一个特殊的标记SEP，并在最开始放CLS标记。最终考察CLS标记对应的输出向量，将其放入Linear transform的输入得到分类。

Case 4：Extraction-based Question Answering (QA)→答案必须出现在文中



入序列包含一篇文章和一个问题，文章和问题都是一个序列。对于中文来说，每个d代表一个汉字，每个q代表一个汉字。你把d和q放入QA模型中，我们希望它输出两个正整数s和e。根据这两个正整数，我们可以直接从文章中截取一段，它就是答案。这个片段就是正确的答案。

模型细节：



对于BERT来说，你必须向它展示一个问题，一篇文章，以及在问题和文章之间的一个特殊标记，然后我们在开头放一个CLS标记。

在这个任务中，你唯一需要**随机初始化从头训练两个向量，分别对应与答案的开始与结束**，用橙色向量和蓝色向量来表示，这两个向量的长度与BERT的输出相同。

- 首先,计算这个**橙色向量**和那些与document相对应的**输出向量（黄色向量）的内积**,计算内积,通过softmax函数，找到数值最大的位置，即为答案的开始位置。

  【这个内积**和attention很相似**，你可以把橙色部分看成是query，黄色部分看成是key，这是一个attention，那么我们应该尝试找到分数最大的位置】
- 类似地，利用蓝色向量可以找到答案的结尾位置。

12.3.6BERT缺陷

- 数据量大
- 训练过程困难

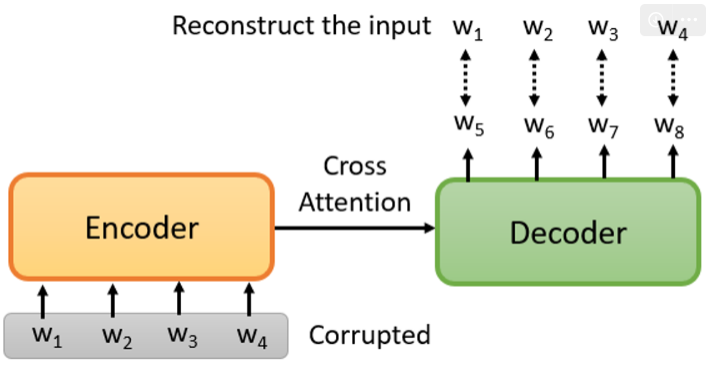
12.3.7 BERT Embryology (胚胎学)

自己训练BERT后，可以观察到BERT什么时候学会填什么词汇，它是如何提高填空能力的?

https://arxiv.org/abs/2010.02480

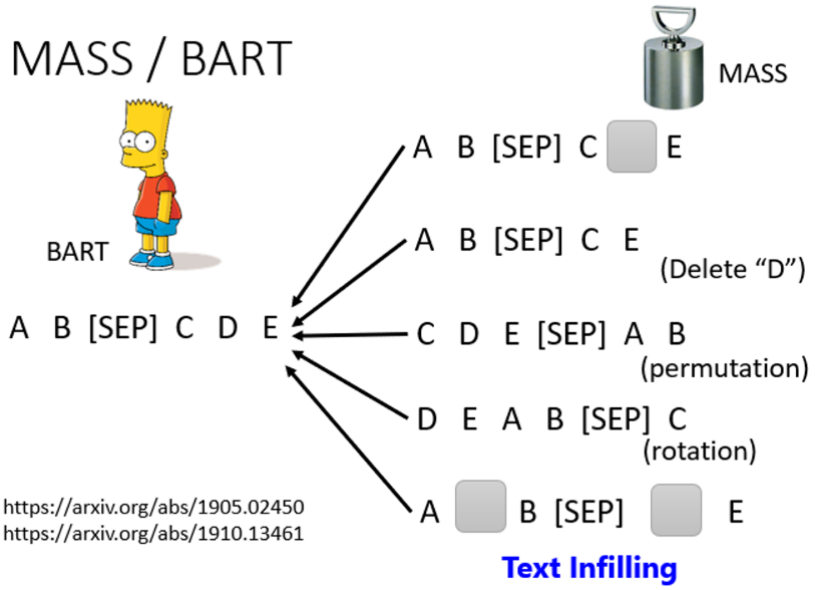
12.4 Pre-training a seq2seq model

输入是一串句子，输出是一串句子，中间用Cross Attention连接起来，然后你故意在Encoder的输入上做一些干扰来破坏它。Encoder看到的是被破坏的结果，Decoder应该**输出句子被破坏前的结果**，训练这个模型实际上是预训练一个**Seq2Seq模型**。



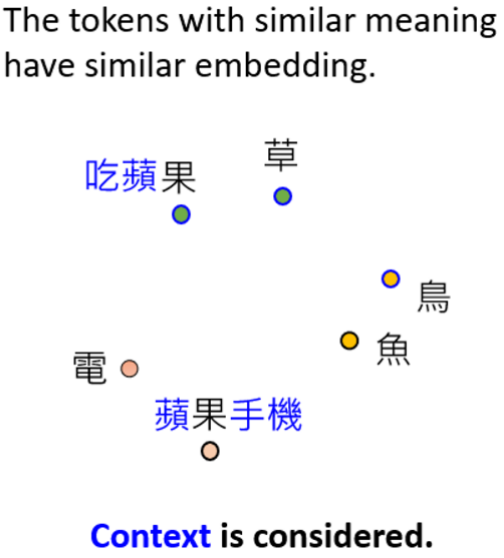
如何“破坏”句子？MASS/BART/T5

- MASS⇒MASK
- BART⇒删除一些词，打乱词的顺序，旋转词的顺序。或者插入一个MASK，再去掉一些词。
- T5(Transfer Text-To-Text Transformer)⇒在C4语料库（Colossal Clean Crawled Corpus）上尝试了各种组合



12.5 BERT提出的价值

12.5.1 Embedding

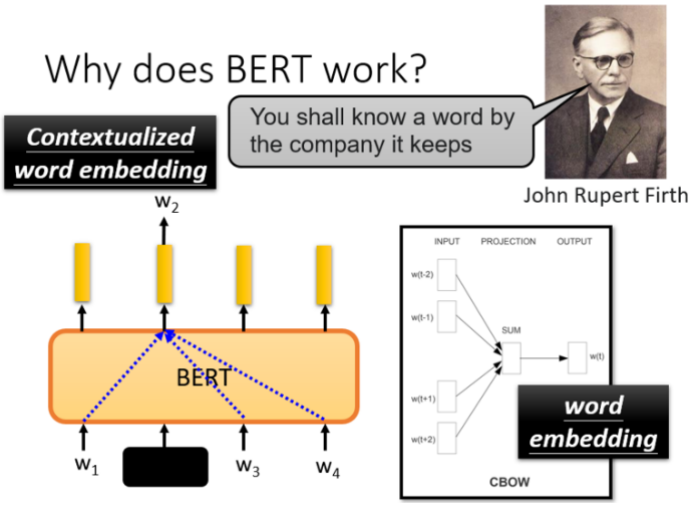


当输入一串文本时，每个文本都有一个对应的向量，称之为**embedding**。这些向量代表了**输入词**的**含义**。

更具体地说，如果你把这些词所对应的向量画出来，或者计算它们之间的**距离。*⇒*意思比较相似的词**，它们的**向量比较接近。**

12.5.2 Embedding in BERT

训练BERT时，我们给它w1、w2、w3和w4，我们覆盖w2，并告诉它预测w2，而它就是从上下文中提取信息来预测w2。所以这个向量是其**上下文信息的精华**，可以用来预测w2是什么。

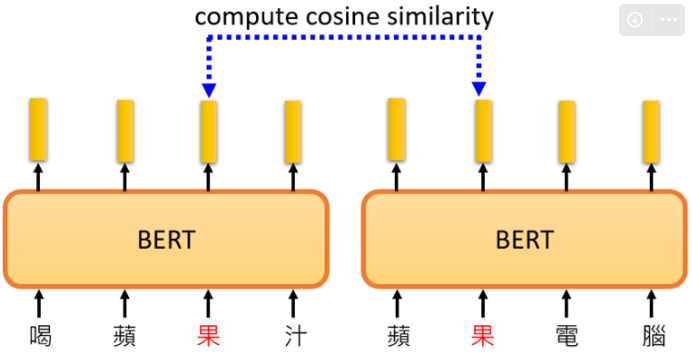


BERT的这些向量是输出向量代表了该词的含义，可以认为BERT在填空的过程中已经学会了每个汉字的意思。

相关技术：CBOW

CBOW所做的，与BERT完全一样。做一个空白，并要求它预测空白处的内容。这个CBOW，这个word embedding技术，可以给每个词汇一个向量代表这个词汇的意义。由于算力原因，CBOW是一个非常简单的模型，只使用了两个变换。今天的**BERT**，就相当于一个**深度版本的CBOW**，

contextualized embedding



语言存在“歧义”，同一token在不同的上下文中会有不同的含义。**BERT在Encoder中存在Self-Attention，会考虑上下文，根据不同的语境，从同一个词汇产生不同的embedding。**

> 计算相似度：BERT知道，前五个 "苹果 "是指可食用的苹果，所以它们比较接近。最后五个 "苹果 "指的是苹果公司，所以它们比较接近。所以**BERT知道，上下两堆 "苹果 "的含义不同。**

奇怪的应用——DNA、蛋白质、音乐的分类

- Applying BERT to **protein, DNA, music classification**

	Protein			DNA				Music
	localization	stability	fluorescence	H3	H4	H3K9ac	Splice	composer
specific	69.0	76.0	63.0	87.3	87.3	79.1	94.1	-
BERT	64.8	74.5	63.7	83.0	86.2	78.3	97.5	55.2
re-emb	63.3	75.4	37.3	78.5	83.7	76.3	95.6	55.2
rand	58.6	65.8	27.5	75.6	66.5	72.8	95	36

把一个DNA序列/蛋白质/音乐预处理成一个无意义的token序列，并使用BERT进行分类，也能得到比较好的结果。

也许它的力量并不完全来自于对实际文章的理解。 还有其他原因。例如，也许BERT只是一套更好的初始参数，与语义不一定有关，只是在训练大型模型时更好。

12.6 Multi-lingual BERT——Zero-shot reading comprehension

它是由很多语言来训练的，比如中文、英文、德文、法文等等，用填空题来训练BERT，这就是Multi-lingual BERT的训练方式。

google训练了一个Multi-lingual BERT，它能够做这104种语言的填空题。神奇的地方来了，如果你用英文问答数据训练它，它就会自动学习如何做中文问答。

Why?

Cross-lingual Alignment?

一个简单的解释是：也许对于multi-lingual的BERT来说，**不同的语言并没有那么大的差异。**无论你用中文还是英文显示，对于具有相同含义的单词，它们的embedding都很接近。汉语中的 "跳 "与英语中的 "jump "接近，汉语中的 "鱼 "与英语中的 "fish "接近，汉语中的 "游 "与英语中的 "swim "接近，也许在学习过程中它已经自动学会了。

验证：MRR（Mean Reciprocal Rank）的值越高，同样意思不同语言的词汇向量越接近，也就是不同embedding之间的Alignment就越好。

数据量是一个非常关键的因素。基于大量数据与大算力支撑，才能够吧不同的语言排列在一起。

语言之间的关系/差距

当训练多语言的BERT时，如果给它英语，它可以用英语填空，如果给它中文，它可以用中文填空，它**不会混在一起，这说明它知道语言的信息也是不同的。**

将某一语言的所有Embedding求平均作为“语言向量”，两个向量相减可以视为“语言的差距向量”。将某一语言的向量加上这一“差距向量”，能够实现“无监督翻译”。

12.7 GPT（类似于Transformer Encoder）

12.7.1 训练任务：Predict Next Token

使用MASK-attention，不断预测“下一个token”。

可以用GPT生成文章。

12.7.2 How to use GPT?⇒给出前半段，补上后半段

In-context Learning(no GD)

结果分析

目前看起来状况是,**有些任务它还真的学会了**,举例来说2这个加减法,你给它一个数字加另外一个数字,它真的可以得到,正确的两个数字加起来的结果,但是有些任务,它可能怎么学都学不会,譬如说一些跟**逻辑推理**有关的任务,它的结果就**非常非常地惨。**

12.8 Self-supervesed Learning Beyond Text

CV

Image - SimCLR

Image - BYOL

Speech

训练：

语音也可以做填空题,就把一段声音讯号盖起来,叫机器去猜;

语音也可以预测接下来会出现的内容

Speech GLUE - SUPERB

李宏毅——语音的基准语料库。