BERT 的演进和应用

Tobias Lee AINLP 今天

作者：Tobias Lee
知乎专栏：NLPer 的成长之路
原文链接，可点击文末"阅读原文"直达：
https://zhuanlan.zhihu.com/p/72805778

Pre-train language model 风头正盛，以 BERT 为代表的模型也在各个任务上屠榜，有一统天下的趋势。知乎上也有不少文章对 BERT 的原理、应用做分析和总结的，例如张俊林老师的一系列文章对 BERT 和 Transformer 的解读就很有深度。但**看别人写和自己读文章梳理一遍的效果是天差地别的**，因此，我也尝试着把最近读的一些关于 Pre-train Language Model 的文章做一次整理。

PLM 的演进

陆续有不少工作在原先的 Pre-train Language Model 的结构上做修改，如果读者对于 GPT、BERT 还不了解，可以看之前的张俊林老师的文章。这里主要是对其变种做一个梳理和对比：

XL-Net：把 Pre-train Model 划分为两类

1. AR Autoregressive: 从某个方向递归地建模 language model，缺点是不能建模双向的 context information，GPT 的做法就是这样。
2. AE Autoencoding： recover sentence from corrupted input，比如根据 masked input 来预测完整的句子，如 bert 所做。有个缺点就是， MASK token 会有一个训练阶段和 fine-tune 阶段的 mismatch，还有个更为严重问题根据 masked sequence 与测试，被 mask 的 token 之间的是相互独立的。例如，原句为 "New York is a city"，masked 之后"[Masked] [Masked] is a city"，那么预测时候是预测 `P(New | is a city)` 以及 `P(York | is a city)`，这样就无法捕获 `New` 和 `York` 之间的关联。

为了解决 AE 的这个问题，作者枚举输入序列的全排列（permutation），例如输入序列 `1 2 3 4`，那么可能的全排列就有 `2 1 4 3` 又或是 `3 2 4 1`，作者希望根据排列中的序列中靠前的 Token 来预测后面的 Token，拿之前的那个例子，假如说排列是 `is a city New York`，那么预测的就是 `P(New | is a city)` 以及 `P(York|is a city New)` （利用排列中靠前的 token 预测后面）：

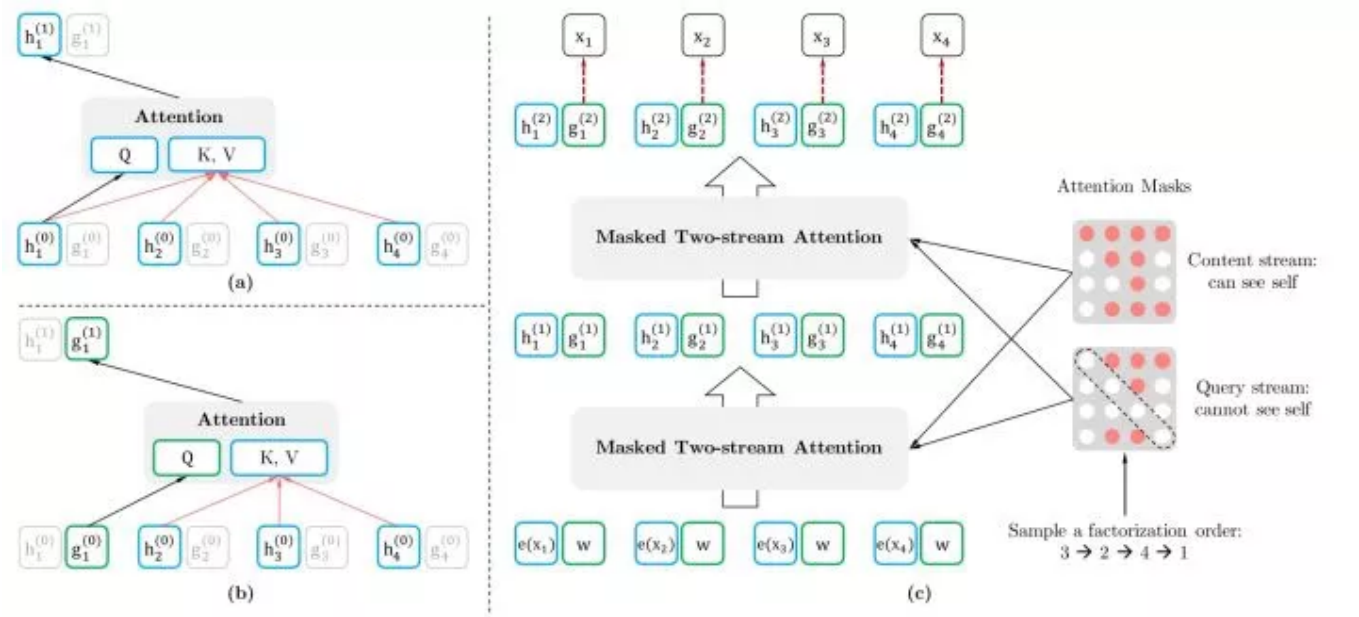


Figure 2: (a): Content stream attention, which is the same as the standard self-attention. (b): Query stream attention, which does not have access information about the content x_z. (c): Overview of the permutation language modeling training with two-stream attention.

具体的实现，其实就是在 attention 的 mask 上做手脚。但在这之前，还有一个问题要解决，就是可能会有多个排列，利用相同的前缀去预测不同的词，所以需要在预测的时候把位置信息加进去。加进位置信息的情况分两种：

1. 预测当前这个词 的时候，要知道这个词在句子中的位置 ，但是不能知道它的内容信息，如上图中的 b 所示
2. 利用这个词预测排列中后面的词时候，要知道他的内容信息，如上图中的 a 所示。

这两种方式，就被称为 two-stream attention（1 是 query stream， 2 是 content stream），实现上就是两个 mask 矩阵，如上图中 (c) 的右边，上方的 content stream，主对角线是不做 mask 的，意味着能看到自己；相反地，对于 query stream，就不能看到自己，而只能看到前面的 token 的信息。

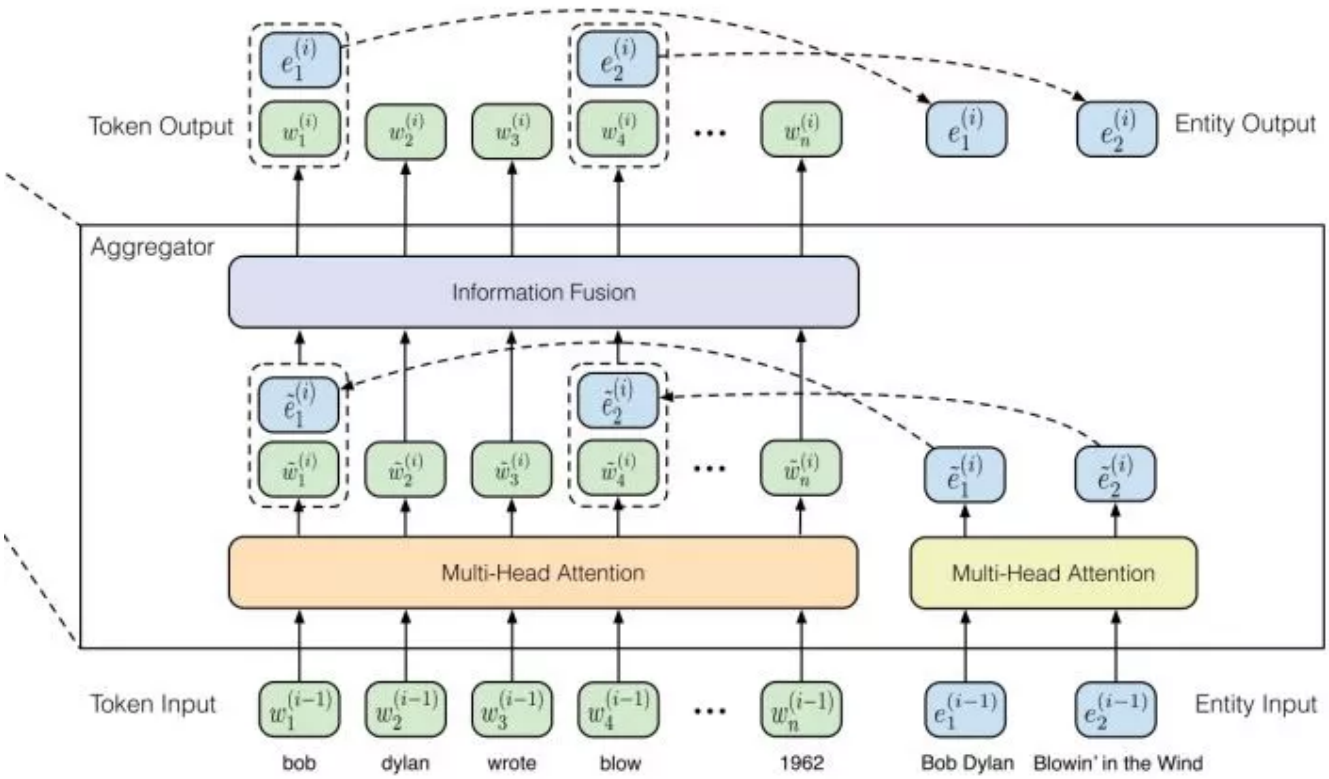
文章的核心贡献就在于**提出 permutation language modeling objective，不再引入额外的 MASK 标记，利用 two-stream attention 替换掉 BERT 中常规的 Attention，并且在很多任务上刷新了 sota**。、想法很好，但是想到长度为 n 的序列有 n! 种情况，即使作者只选择后面一部分的 token 进行 prediction，这个预训练过程的花费也是巨大的，机器之心给出的估计是 6万刀，好吧穷也是为什么限制我们发 paper 的原因吧。

XLM：把预训练的想法拓展到跨语言，做法也很简单，构建一个 shared sub-word （文章中使用的是 BPE）vocabulary 来对多种语言使用同一个词表，然后通过 Masked LM 做 Pre-train；对于有并行语料的场景，可以直接把 source 和 target 拼接起来然后中间加分隔符，不区分 source 和 target 地进行 mask，还是做 language modeling。作者发现这样的做法能够带来以下的好处

1. a better initialization of sentence encoders for zero-shot cross-lingual classification
2. a better initialization of supervised and unsupervised neural machine translation systems
3. language models for low-resource languages
4. unsupervised cross-lingual word embeddings

前面两个都比较直观，因为本身 BERT 的一个用法就是在 large scale 上的预料上 pre-train 然后到下游任务上 fine-tune。对于第三点，作者是通过利用多个语言来辅助少资源的语言建模任务，例如利用英语和印地语来帮助建模尼泊尔语，实验发现，相比英语，同为梵语后代的印地语能够更大幅地降低 PPL ，并且二者的结合能够带来更好的效果；对于第四点，因为训练的时候用的是 shared vocab，对于不同于语言的 word，只要在词表里查找训练之后的 embedding 即可。

ERNIE（THU）：把知识图谱的信息整合进 pre-train 的过程中，具体地，对于文本中的实体，在知识库中找到对应的 entity，利用 transE 来进行表示之后，在 BERT 的 text encoder 之上再加一层 knowledge encoder：



ERNIE-THU

就是利用经过 multi-head attention 得到 的 token representation 和 entity representation 过一个 FFW 得到 combined representation，然后再根据此过 FFW 得到 token 和 entity 的表示，是一个融合 -> 拆分的过程。为了适应这样的结构，文章也添加了一个 entity alignment 任务，根据给定的 toke 预测对应的 entity。模型在关系分类上的结果也证明了其确实有效。文章**主要的 contribution 就在于想到把 KG 融合到预训练 language model 的过程中去，剩下的实验和设计就非常水到渠成了，也是非常有意思的一篇工作**。

ERNIE（Baidu）：名字和清华的撞车了，但是 motivation 上还是不一样的。THU 的主要 argue 外部知识的重要性，重点在于融合；而百度这边的，则是在 mask 的 level 上对中文文本做了调整，英文 mask 单词是很直接的想法，而中文的处理一般是以词为单位，因此会有 phrase mask 以及 entity mask，例如人名、地名的 mask。通过添加两种 mask 机制，来让 language model 隐式地学习到 entity 之间的关联。比较有趣的一点是，作者还在 Pre-train 中加入了 Dialogue 语料，结果显示效果也会有一些提高。这指明了一个方向，在预训练阶段，除了修改模型、目标函数以外，选择高质量、特定 domain 的语料也是可行的方向。

PLM 的应用

PLM 的应用，也是最近顶会产出 Paper 很多并且刷新很多 state-of-the-art 的一个方向。除了已经在原 Paper 中展现的对于简单的分类任务、MRC 等，这方面的工作现在我觉得比较有意思的是：

1. 把一些问题改造成能够用 BERT 去解决的形式，然后在具体的数据集上 Fine-tune
2. 利用 BERT 做一些数据集的扩充和增广

改造任务

BERTSum：利用 BERT来做摘要生成任务，抽取式的摘要。对于每个句子，前面设置一个 CLS 在此基础之上判断是否选取这个句子；进一步地为了整合 Document-Level 的信息，再得到句子表示之后（即 CLS token），可以再做一次 self-attention 或者是过一层 RNN。此外，除了 BERT 原有的 Positional Encoding，文章为了区别句子（某些词属于某个句子），额外增加了一个 Segment Encoding，对句子进行交错编码。

BertSUM 实验结果

不过从结果上来看，再加一层对于模型的提升不是特别大（对比 BERT + Classifier 和 BERT + Transformer），也似乎说明**BERT 本身其实就能够考虑到比较远距离的语义关联信息**。

ASBA：Aspect-level Sentiment Analysis 是即情感分类问题的一个细粒度版本，给定句子判定某个方面的情感极性，具体地又有两种形式：

1. 给定 target entity t ，以及 aspect a，询问对于特定对象 在 在方面的情感极性
2. 给定某个 aspect a ，询问这个层面的情感极性

后者可以看做是前者的简化版本，因此文章的讨论也主要是基于第一种形式。作者通过构建辅助句子（Auxiliary Sentence）来组成问答对，从而利用 BERT 中的句子对分类范式来解决 APSA 问题。比如，对于评论：

> 杭州的房价很贵，而安吉的放假很便宜并且气候很适宜

可以构建以下几种形式的辅助句子：

1. 你觉得杭州的房价怎么样？（t = 杭州 ， = 房价 ）
2. 杭州 - 房价
3. 杭州的房价高 / 杭州房价低 / 杭州的房价不知道 （分别对应 negative / positive / None）
4. 杭州- 房价-高 / 杭州-房价-低 / 杭州-房价-不知道

前两周，将句子和评论拼接起来，在 label 中给出结果，BERT 预测的是这个 label；后面两种形式拿 BERT 的 NSP 任务来套，对于三种结果每个都和评论计算一个 score，取最高的作为分类结果即可。

数据扩充

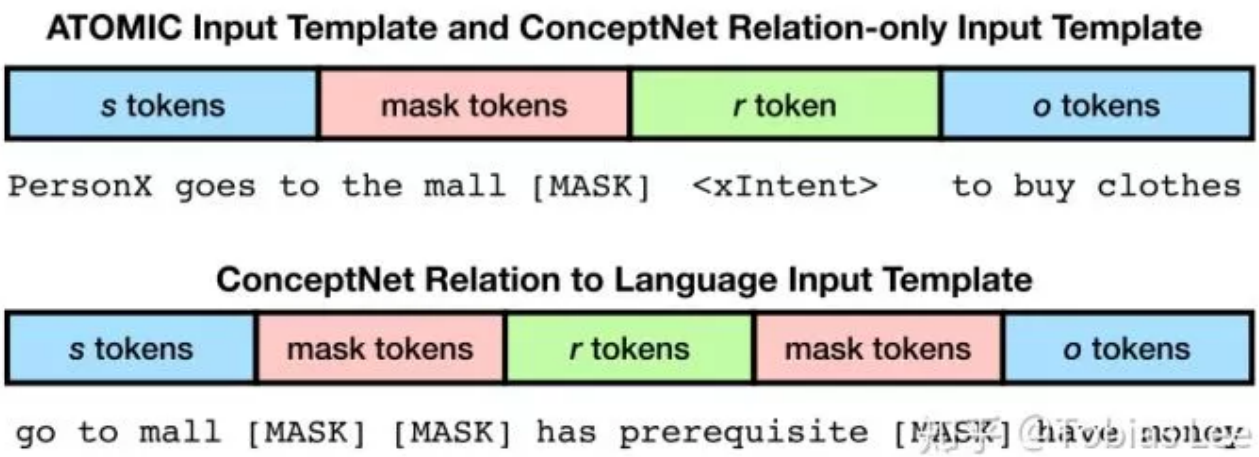
BERT 在各个数据集上屠榜之后，甚至超过人类表现之后，一个很自然的问题，还有没有能难倒 BERT 的数据集？另外，一个很重要的事实是，**DL 技术的进步是随着数据集的发展而不断向前的**，李飞飞老师做的 ImageNet 带来了神经网络的繁荣，而像 NLP 领域的 WMT 机器翻译数据集也是推动机器翻译技术不断进步的原因之一。道理也很简单，对于深度学习这样的实验科学，必然需要 benchmark 来做 Evaluation，只有找到了靶子，才能更好地练习射箭。因此，找到能够难倒 BERT 的数据集，不能说咱们有了 BERT 就一把梭，要找到够难的数据集，倒逼技术进步。这方面，还是有不少有趣的工作的

HellaSwag：Yejin Choi 组的工作，其核心想法就是上面说的那段，**数据集应该和模型一起进化**。SWAG 是 18 年提出的一个推理数据集（给定上文，判断一个句子是否是对应的结尾），人类能够达到 88% 的准确率，BERT之前的 state-of-the-art 是 60% 不到，而 BERT 则能达到 86% 的准确率。很自然地，会问，为什么 BERT 效果这么好？实验证明，BERT 并不具备很强的常识推理能力，而是通过 fine-tune 阶段习得的数据分布的归纳偏好(dataset-specific distribution biases），实现了接近人类的性能。下一个问题就是，如何难倒 BERT 呢？解铃还须系铃人，文章使用 adversarial filtering 技术，随机将数据集分成训练集和测试集，然后在训练集上训练分类器，利用 Pre-train Language Model 来生成假的 candidate，并且不断替换能够被分类器轻松识别的候选句子，直到在这些具有对抗性的候选答案上的准确率收敛为止。文章有意思的是对 BERT 在 SWAG 取得较好性能的探究，首先是对 fine-tune 数据集的 size 做了探究，发现只要十几个样本 BERT 就能达到 76% 的准确率，当然这并**不能得出是来对 data set 的 fit 所致**，文章还做了一个实验，发现即使是不给上文，也能达到 75% 的准确率，**说明 fit 故事结尾就能够学习到很多的 bias**，此外，即使是打乱结尾的句子词序，带来的性能降低也不足 10%，因此得出了 BERT 在 SWAG 上的出色表现来自于对于 surface 的学习，学习到合理 ending 的某些 Realization Pattern 的结论。

COMET：同样出自 Yejin Choi 组，idea 也很有意思，利用 Pre-train language model，来进行常识 knowledge triplet 的生成。方法也很简单，对于 KB 的三元组， 分别对应主语、关系和对象，像 ConceptNet 里的 "taking a nap" 就可以写成：

> (s = take a nap, r = Causes, o = have energy) 小憩一下能够恢复能量

我们的任务就是给定 s 和 r ，来预测出 ***o*** 。有了这样的想法之后，就要把任务改造一下来适应 Pre-train Model：



就是把 r ， o， s 看成句子，然后用 MASK 隔开，训练的时候利用 MLE，预测的时候就可以把交给 Model 来进行预测。得到的结果非常有趣：

Seed	Relation	Completion	Plausible
piece	PartOf	machine	✓
bread	IsA	food	✓
oldsmobile	IsA	car	✓
happiness	IsA	feel	✓
math	IsA	subject	✓
mango	IsA	fruit	✓
maine	IsA	state	✓
planet	AtLocation	space	✓
dust	AtLocation	fridge	
puzzle	AtLocation	your mind	🤔
college	AtLocation	town	✓
dental chair	AtLocation	dentist	✓
finger	AtLocation	your finger	
sing	Causes	you feel good	✓
doctor	CapableOf	save life	✓
post office	CapableOf	receive letter	✓
dove	SymbolOf	purity	✓
sun	HasProperty	big	✓
bird bone	HasProperty	fragile	✓
earth	HasA	many plant	✓
yard	UsedFor	play game	✓
get pay	HasPrerequisite	work	✓
print on printer	HasPrerequisite	get printer	✓
play game	HasPrerequisite	have game	✓
live	HasLastSubevent	die	✓
swim	HasSubevent	get wet	✓
sit down	MotivatedByGoal	you be tire	✓
all paper	ReceivesAction	recycle	✓
chair	MadeOf	wood	✓
earth	DefinedAs	planet	✓

Table 7: **Randomly selected and novel** generations from the ConceptNet development set. Novel genera-

像 `puzzle at your mind` 这样的信息可以说是非常有新意的。这篇文章除了能够在 ConceptNet 上做知识组的 completion 以外，也告诉我们，pre-train language model 还是能

学到一些常识信息的，但这是不是还是 Surface Realization 呢？有待探究。

所以，相比较拿 BERT 这类模型去套一些已有的任务，如何另辟蹊径地找一些类似数据增强、数据集生成的任务，也许更有意义。

PLM 的分析

对于 BERT 为什么表现出色，除了根据任务来探究这个问题以外，也有不少工作在一些基础的语言任务上做研究：

Syntactic Ability：探究 BERT 的句法能力，Yoav Goldberg 的一篇类似实验报告的文章，写的比较随意，通过主谓一致任务来探究 BERT 的对于句法结构的捕获能力。文章主要的发现有：

1. 一般我们会认为像 RNN 这样循环的结构对于句法（尤其是主谓一致任务非常重要），但是结果表明，purely attention-based model 也能够做的很好，至少和 LSTM 的表现是在同一个 level 上的
2. BERT-base 的表现会比 BERT-large 好，这是不是意味着对于句法任务而言， model capacity 并不是一个主要的因素。自然而然地，我们想到，那么多大的 capacity 能够刚好 cover 住这个任务?

Attention：BERT 主要组件是 Transformer，而 Transformer 的就是 Self-Attention + Multi-head Attention，Attention 权重是可以可视化出来看一看的，虽然最近有一些工作认为 Attention 不能解释模型行为的一个，但看看无妨：

比较有趣的现象是对 [SEP] 的 attention weight 大多很大，文章认为这可以看作是一个**空操作**，当不知道 attention 谁的时候就会 attend 到 [SEP] 上，为了证明这一点，文章可视化了 loss 对 attention 权重梯度，发现这些权重的梯度的大小很小，意味着其对于最终结果不会有太大的影响。另外，文章还发现在 self-supervised 的过程中，能够学习到一些句法知识，而这是通过 attention weight 来实现的，文章把 BERT 中 head 的 attention weight 拿出来对 pre-

trained word embedding（Glove） 做一个加权，来预测一个词是否是另外一个词的 head，结果能取得 77 的 UAS，说明 attention weight 之中也包含了很多信息。

总结

如张俊林老师所说，BERT 大有一统 NLP 局面的趋势，但是学术上依旧有很多坑可以填，虽然我们可能没有那么多的机器，但是，除了比手速套应用发文章，搞清楚 BERT 的原理、适用的场合，提出 idea 并且用实验去验证，这或许是更有意义的做法，至于实际的效用，就交给工业界去 judge 就行；反过来，工业界，也不是每家每户都有那么多卡的能用上 TPU 的，那部署的时候怎么用这么大的模型，是不是要蒸馏、压缩一下，也是很值得考虑的问题。**总而言之，NLP 不会因为 BERT 而失去活力，反倒是，焕发新春**。

Reference

1. Assessing BERT's Syntactic Abilities
2. Cross-lingual Language Model Pretraining
3. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction
4. ERNIE: Enhanced Language Representation with Informative Entities
5. ERNIE: Enhanced Representation through Knowledge Integration
6. HellaSwag: Can a Machine Really Finish Your Sentence?
7. What Does BERT Look At? An Analysis of BERT's Attention
8. XLNet: Generalized Autoregressive Pretraining for Language Understanding

---

推荐阅读

BERT系列文章汇总导读

BERT 瘦身之路：Distillation，Quantization，Pruning

BERT论文笔记

关于AINLP

AINLP 是一个有趣有AI的自然语言处理社区，专注于 AI、NLP、机器学习、深度学习、推荐算法等相关技术的分享，主题包括文本摘要、智能问答、聊天机器人、机器翻译、自动生成、知识图谱、预训练模型、推荐系统、计算广告、招聘信息、求职经验分享等，欢迎关注！加技术交流群请添加AINLP君微信(id：AINLP2)，备注工作/研究方向+加群目的。