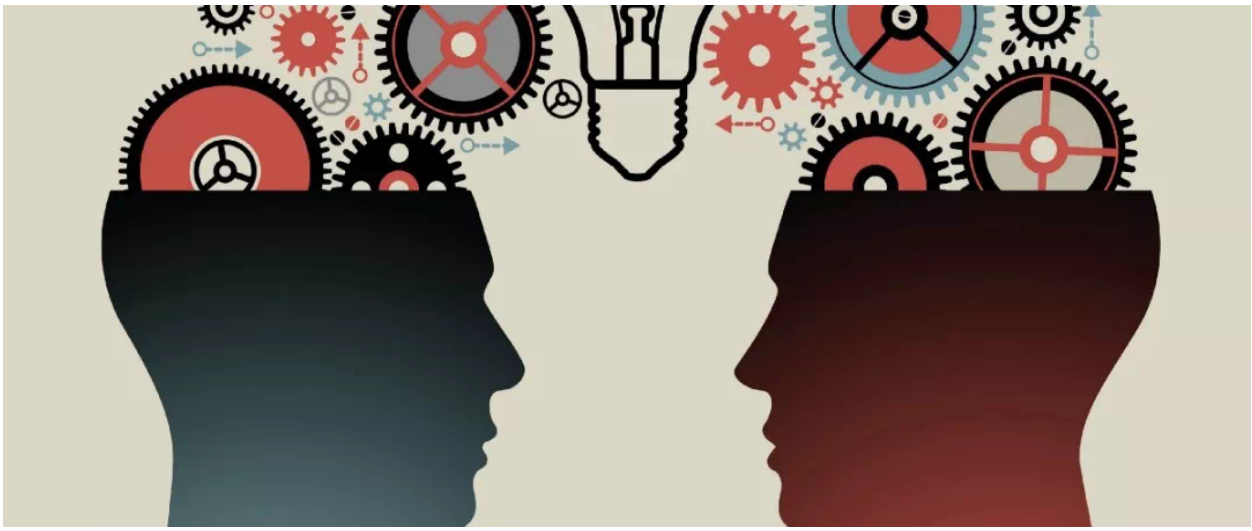
泰岳语义工厂 原创
2019/06/17 16:59

泰岳语义工厂
作者

BERT时代与后时代的NLP（一）



本文转载自 知乎专栏 智能对话机器人技术

2018年是NLP的收获大年，模型预训练技术终于被批量成功应用于多项NLP任务。之前搞NLP的人一直羡慕搞CV的人，在ImageNet上训练好的模型，居然拿到各种任务里用都非常有效。现在情形有点逆转了。搞CV的人开始羡慕搞NLP的人了。CV界用的还是在有监督数据上训练出来的模型，而NLP那帮家伙居然直接搞出了在无监督数据上的通用预训练模型！要知道NLP中最不缺的就是无监督的文本数据，几乎就是要多少有多少。还有个好消息是目前NLP中通用预训练模型的效果还远没达到极限。目前发现只要使用更多的无监督数据训练模型，模型效果就会更优。这种简单粗暴的优化方法对大公司来说实在再经济不过。而且，算法本身的效果也在快速迭代中。NLP的未来真是一片光明啊～

BERT发布之后，点燃了NLP各界的欢腾，各路神仙开始加班加点各显神通，很多相关工作被发表出来。本文会介绍其中的一些代表性工作，但更重要的是希望理清这些背后的逻辑，为它们归归类。通过这些思考，我自己也对NLP以后的工作方向有些预测，供大家参考。

本文的内容主要包括以下几部分：

1. 我对迁移学习和模型预训练的一些思考，以及对未来工作方向的粗略预测
2. 各类代表性工作的具体介绍（熟悉的同学可忽略），又细分为以下几大类：

- 有监督数据预训练
- 自监督训练
- 无监督数据预训练
- 多个有监督数据同时训练：多任务学习

3. 一些我们的实践经验、别人和自己的观点、以及总结和感想

第一和第三部分内容相对少，原创密度大点，大家要是赶时间的话看这两部分就够了。第二部分的内容都是具体技术，有很多很好的文章都介绍过。放在本文当中一是为了文章的完备性，另一个是里面提到的一些知识点在其他地方没怎么提到。第三部分也会涉及到我们（爱因互动）自己在一些任务上的实验工作，期望这些结果能坚定大家在自己的工作中把模型预训练技术用起来。

终于可以开始了。

一、迁移学习与模型预训练：何去何从

迁移学习分类

把我们当前要处理的NLP任务叫做**T**（T称为**目标任务**），迁移学习技术做的事是利用另一个任务**S**（S称为**源任务**）来提升任务T的效果，也即把S的信息迁移到T中。至于怎么迁移信息就有很多方法了，可以直接利用S的数据，也可以利用在S上训练好的模型，等等。

依据目标任务T是否有标注数据，可以把迁移学习技术分为两大类，每个大类里又可以分为多个小类。

第一大类是T没有任何标注数据，比如现在很火的无监督翻译技术。但这类技术目前主要还是偏学术研究，离工业应用还有挺长距离的。工业应用中的绝大部分任务，我们总是能想办法标注一些数据的。而且，目前有监督模型效果要显著优于无监督模型。所以，面对完全没有标注数据的任务，最明智的做法是先借助于无监督技术（如聚类/降维）分析数据，然后做一些数据标注，把原始的无监督任务转变为有监督任务进行求解。基于这些原因，本文不再介绍这大类相关的工作。

第二大类是T有标注数据，或者说T是个有监督任务。这类迁移学习技术又可以依据源任务是否有监督，以及训练顺序两个维度，大致分为四小类：

- 源任务S是无监督的，且源数据和目标数据同时用于训练：此时主要就是自监督（self-supervised）学习技术，代表工作有之后会讲到的CVT。

- 源任务S是有监督的，且源数据和目标数据同时用于训练：此时主要就是多任务（multi-task）学习技术，代表工作有之后会讲到的MT-DNN。

- 源任务S是无监督的，且先使用源数据训练，再使用目标数据训练（序贯训练）：此时主要就是以BERT为代表的无监督模型预训练技术，代表工作有ELMo、ULMFiT、GPT/GPT-2、BERT、MASS、UNILM。

- 源任务S是有监督的，且先使用源数据训练，再使用目标数据训练（序贯训练）：此时主要就是有监督模型预训练技术，类似CV中在ImageNet上有监督训练模型，然后把此模型迁移到其他任务上去的范式。代表工作有之后会讲到的CoVe。

	无监督数据	有监督数据（其他任务）
同时训练	自监督学习	多任务学习
序贯训练	预训练无监督模型（BERT）	预训练有监督模型（ImageNet）

何去何从

现状分析

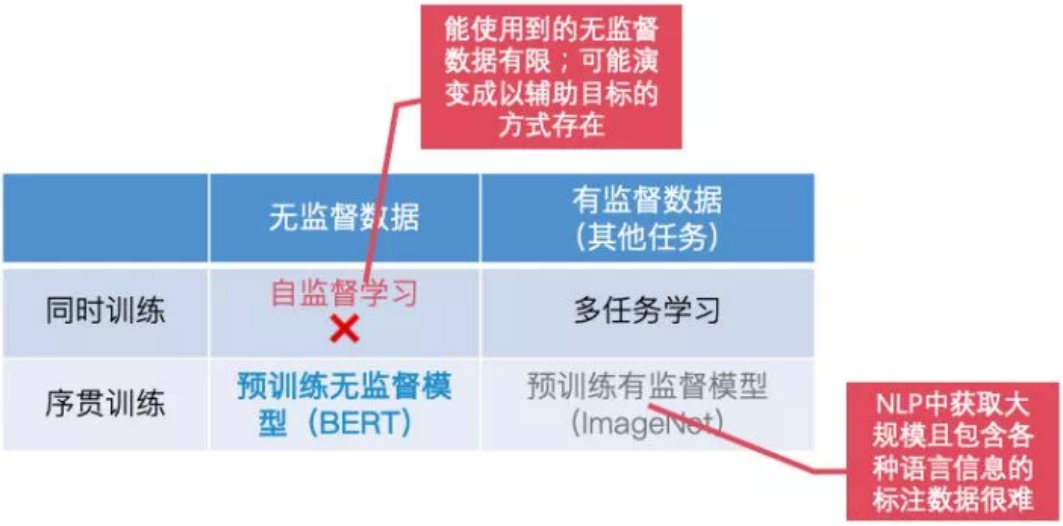
先说说上表中四个类别的各自命运。以BERT为代表的**无监督模型预训练技术**显然是最有前途的。之前也说了，NLP中最不缺的就是无监督数据。只要堆计算资源就能提升效果的话，再简单不过了。

而无监督预训练的成功，也就基本挤压掉了**自监督学习**提升段位的空间。这里说的自监督学习不是泛指，而是特指同时利用无监督数据和当前有监督数据一起训练模型的方式。既然是同时训练，就不太可能大规模地利用无监督数据（要不然就要为每个特定任务都训练很久，不现实），这样带来的效果就没法跟无监督预训练方式相比。但自监督学习还是有存在空间的，比如现在发现在做有监督任务训练时，把语言模型作为辅助损失函数加入到目标函数中，可以减轻精调或多任务学习时的灾难性遗忘（Catastrophic Forgetting）问题，提升训练的收敛速度。所以有可能在训练时加入一些同领域的无监督数据，不仅能减轻遗忘问题，还可能因为让模型保留下更多的领域信息而提升最终模型的泛化性。但这个方向迎来大的发展可能性不大。

而类似CV中使用大规模**有监督数据做模型预训练**这条路，看着也比较暗淡，它自己单独不太可能有很大前景。几个原因：1) 这条路已经尝试了很久，没有很显著的效果提升。2) NLP中获取大规模标注数据很难，而且还要求对应任务足够复杂以便学习出的模型包含各种语言知识。虽然机器翻译任务很有希望成为这种任务，但它也存在很多问题，比如小语种的翻译标注数据很少，翻译标注数据主要还是单句形式，从中没法学习到背景信息或多轮等信息。但从另一个方面看，NLP搞了这么久，其实还是积累了很多标注或者结构化数据，比如知识图谱。如何把这些

信息融合到具体任务中最近一直都是很活跃的研究方向，相信将来也会是。只是BERT出来后，这种做法的价值更像是打补丁，而不是搭地基了。

多任务学习作为代价较小的方法，前景还是很光明的。多个同领域甚至同数据上的不同任务同时训练，不仅能降低整体的训练时间，还能降低整体的预测时间（如果同时被使用），还能互相提升效果，何乐而不为。当然，多任务学习的目标一开始就不是搭地基。



上面说了这么多，其实想说的重点在下面。这些技术不一定非要单独使用啊，组合起来一起用，取长补短不是就皆大欢喜了嘛。

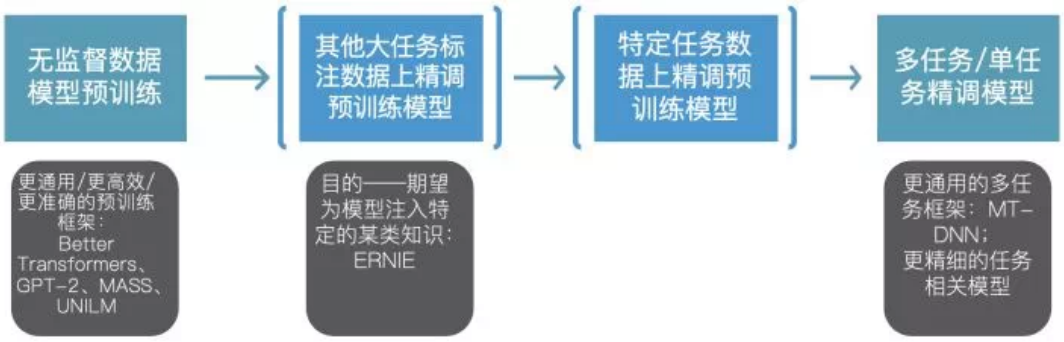
先回顾下现在的无监督模型预训练流程，如下图：

首先是利用大的无监督数据预训练通用模型，优化目标主要是语言模型（或其变种）。第二步，利用有监督数据精调上一步得到的通用模型。这么做的目的是期望精调以后的通用模型更强调这个特定任务所包含的语言信息。这一步是可选的（所以图中对应加了括号），有些模型框架下没有这个步骤，比如BERT里面就没有。第三步才是利用有监督数据中对应的标注数据训练特定任务对应的模型。

那这个流程接下来会怎么发展呢?

未来可期

上面我已经对四类方法做了分别的介绍，包括对它们各自前途的简单判断，也介绍了当下效果最好的模型预训练流程。相信未来NLP的很多工作都会围绕这个流程的优化展开。我判断这个流程会继续发展为下面这个样子：



详细说明下每个步骤：

1. 第一步还是利用大的无监督数据预训练通用模型。但这里面目前可以改进的点有很多，比如发展比Transformer更有效的特征抽取结构，现在的Evolved Transformer和Universal Transformer等都是这方面的探索。发展更有效更多样化的预训练模型目标函数。目前预训练模型的目标函数主要是(Masked) LM和Next Sentence Prediction (NSP)，还是挺单一的。面向文档级背景或多轮这种长文本信息，未来应该会发展出更好的目标函数。比如有可能会发展出针对多轮对话这种数据的目标函数。
   BERT主要面向的是NLU类型的任务，目前微软提出的MASS、UNILM从不同的角度把BERT框架推广到NLG类型的任务上了，细节我们之后会讲到。GPT-2利用更大的模型获得了更好的语言模型。更多更好的数据，更大的模型带来的改进有没有极限？目前还不知道，相信很多公司已经在做这方面的探索了。但这个游戏目前还是只有大公司能玩得起，训练通用大模型太耗钱了。提升训练效率，很自然的就是另一个很重要的优化方向。

2. 第二步是利用其他大任务的标注数据或已有结构化知识精调第一步获得的通用模型。步不一定以单独的形式存在，它也可以放到第一步中，在预训练通用模型时就把这些额外信息

注入进去，比如百度的ERNIE就是在预训练时就把实体信息注入进去了。既然人类在漫长的AI研究史上积累了大量各式各样的结构化数据，比如机器翻译标注数据，没理由不把它们用起来。相信未来会有很多知识融合（注入）这方面的工作。

3. 第三步和前面流程的第二步相同，即利用当前任务数据进一步精调上一步得到的通用模型。这么做的目的是期望精调后的模型更强调这个特定任务所包含的语言信息。ELMo的实验结论是，加入这一步往往能提升下一步的特定任务有监督训练的收敛速度，但仅在部分任务上最终模型获得了效果提升（在另一部分任务上持平）。
   另一种做法是把这一步与下一步的特定任务有监督训练放在一块进行，也即在特定任务有监督训练时把语言模型作为辅助目标函数加入到训练过程中，以期提升模型收敛速度，降低模型对已学到知识的遗忘速度，提升最终模型的效果。GPT的实验结论是，如果特定任务有监督训练的数据量比较大时，加入辅助语言模型能改善模型效果，但如果特定任务有监督训练的数据量比较小时，加入辅助语言模型反而会降低模型效果。但ULMFiT上的结论刚好相反。。所以就试吧。

4. 利用多任务或者单任务建模方式在有监督数据集上训练特定任务模型。多任务的很多研究相信都能移植到这个流程当中。我们之后会介绍的微软工作MT-DNN就是利用BERT来做多任务学习的底层共享模型。论文中的实验表明加入多任务学习机制后效果有显著提升。相信在这个方向还会有更多的探索工作出现。在单任务场景下，原来大家发展出的各种任务相关的模型，是否能在无监督预训练时代带来额外的收益，这也有待验证。

总结下，未来NLP的主要工作可能都会围绕这个流程展开。对流程前面步骤的优化带来的收益比后面步骤大，也更难。所以诸君请自己拿捏吧~。

入门 NLP