NLP.TM[28] | 浅谈NLP算法工程师的核心竞争力

原创 机智的叉烧 CS的陋室 2020-03-02 00:35

A Place Called You

Various Artists - 歌曲合辑

【NLP.TM】

本人有关自然语言处理和文本挖掘方面的学习和笔记，欢迎大家关注。

这篇文章来自我的一份知乎的回答，搬运过来给大家一起看看。

往期回顾：

- NLP.TM[22] | 如何修正NLP问题的bad case
- NLP.TM[24] | TextCNN的个人理解
- NLP.TM[25] | CS224N学习小结
- NLP.TM[26] | bert之我见-attention篇
- NLP.TM[27] | bert之我见-positional encoding

目前尚属新人，看到的比较少，但是工作了接近一年，大概知道自己和大佬们的差距在何处，这些其实就是自己不足的地方。来一份自己目前比较高赞的总结：

ML&DEV[8] | 算法在岗一年的经验总结

下面部分内容可能不局限在NLP上，而是整个算法圈子工程师的竞争力体现，当然还有针对NLP本身的。

现状

现状先说说现状吧，从我18年两次校招（额，比较久远了）来看，很多有竞争力的应聘者已经具备基本模型的理解和开发能力，说白了就是算得上“算法工程师”的选手，对认识的模型基本有一定的见解，开发能力虽不如专业的后端、服务端高手，但是学学也是能写几行实现代码，完成实验。所以说自己模型理解有多厉害，多会写代码其实都不能称为所谓的优势，毕竟这些方面不及格的人，其实在这行找到工作都非常困难。

问题的解决能力

问题的解决能力大部分入门的算法都容易把精力集中在模型上，无论是理论还是实践，但是却背离了算法本身所依托的背景。作为工程师而非科研人员，其核心价值在于解决问题而非研究模型，当一个问题能使用简单的规则策略去解决的时候，我们并不应该使用训练时间长、结果不可控、效果风险不明确的模型作为解决方案。

举例子，客服机器人大家都知道，对于算法而言可能就会开始看文本生成、对话机器人的知识了，但是实质上，一个“触发词-回答”的词典就能达到初版上线要求。命名实体识别在很多场景其实没有标注数据，模型根本无用武之地，语言模板规则、词典最大逆向匹配都是很好的方法，不必纠结于模型。

另一方面，解决问题的能力还要体现在逆境上，数据不全、数据质量不高、缺乏训练数据的时候，你要提出你的解决方案，尤其是在新项目下，这种场景很常见，能开天辟地的往往是高手，你具备这些能力才能够进一步进阶，除了研究模型本身，多想想类似“万一有XXX情况，这个方法还是否可行，不可行怎么办。

记住，用户对实现方法的高端低端是没有任何感知的，老板也是。

模型的优化能力

算法工程师的工作看着很简单，问题一来加模型调结构上线。

但问题是，如果模型效果不好，你该怎么办？换一个模型，调参？基本也是差不多的效果，大家看论文其实可以看到，在论文里模型的提升基本不会超过10个点对吧，所以换模型只能存在于微调中。那你就没辙了？这就是体现算法工程师优势的能力所在。

你可以：看看数据质量、数据量怎么样要不要提升。看看特征是否足够表征个体。标注是否正确。模型特性是否符合问题（例如RNN和CNN的适用场景）等等。

尤其是在NLP领域，整体非常黑盒，整个模型的流程相对固定，基本没有什么干预措施，所以你的选择会变得更少，如何能进一步提升结果，（新增人工特征、优化数据集、优化模型等）这个需要的是你的智慧和经验了。

顺风局谁都会打，逆风局如何化腐朽为神奇才是高手该体现的能力。

工程能力

这个概念有点模糊，简单举几个例子吧：

- 你的算法耗时多少，是否具备多线程的能力。（例如一般而言RNN系列模型的耗时都偏长）
- 算法如何部署，在线部分特征怎么构建和传入。
- 你写的算法复杂度多少，有没有优化空间。（例如用Trie树代替便利词表实现检索功能）模型的更新是热更新还是需要重启服务，生效时长等。

具体的难以三言两语说清，欢迎大家看看我这篇文章：

ML&DEV[6] | 浅谈算法工程师的工程能力

对公司，论文始终不赚钱，要赚钱始终是靠能跑的起来让用户用的产品，所以对于算法工程师而言，首先是工程师，然后才是算法。

知识更新迭代的能力

这一行知识更新很快，要始终保持前沿其实很难。虽说工业界不见得每个项目都是bert走天下，甚至不能说是“前沿”了。但是你也知道，这是目前公认大部分领域的最佳结果，那你就得会，甚至要懂，知道里面的positional encoding，知道里面的多层encoder-decoder，知道transformer的原理，面试上肯定的，但是工作中你也要知道这点，方便你分析结果，甚至做出针对问题的改进。

要紧跟时代发展，你还是要坚持看论文，理解论文，知道一些新的解决问题的思路，例如这个positional encoding就很能说明问题，这是一个可以代替RNN系列来处理序列信息的方法，这个能不能用在TextCNN上等等，这就会成为你工具百宝箱中的一个。

NLP工程上的地位和难点

回到NLP，其实NLP在现实应用上并非是一颗明亮的星，而是承担的一个辅助的作用，很少独立存在，为核心问题提供更多信息的一个方法，哪怕是对话机器人这种重度nlp的任务，也不是以nlp为唯一技术点的。在我看来，目前nlp主要承担语义理解方面的功能，文本分类、实体识别、语义相似度等是们目前最为常用的工作，他在很多大系统中有稳定的一席之地，如搜索（理解query含义）、推荐（理解文档内容）、对话（理解用户需求）等场景。

至于难点，我列举一下：

1. 很多场景下数据其实并不支持我们使用模型，数据量问题、数据质量等。
2. 工程性能上，可能会不允许我们使用太过重型的模型，bert之类的真不是谁都跑得起、训的起的。
3. 由于调研时间不定，很容易成为团队项目安排上时间拖后腿的角色，尤其是NLP这种黑盒性比较强的任务上。
4. 边缘case更加多又更突出，如“我想过过过儿过过的生活”，这种case非常常见。

然后一一来对应上面难点，提炼出我们需要的能力：

- 无模型或轻量模型下解决问题的能力。（12）
- 工程上举一反三的能力。（3）
- 任务的取舍能力与安排能力。（34）
- 这些就是我理解的，高于模型和技术本身，我们需要的能力。

以上。

我是叉烧，欢迎关注我！

叉烧，OPPO搜索算法工程师。19届北京科技大学数理学院统计学硕士（保研），17届北京科技大学信息与计算科学、金融工程本科双学位。论文7篇，1项国家自科参与人，国家级及以上会议4次，1次优秀论文，国家奖学金，北京市优秀毕业生。曾任去哪儿网大住宿事业部产品数据，美团点评出行事业部算法工程师。

微信 zgr950123
邮箱 chashaozgr@163.com
知乎 机智的叉烧

微信个人公众号
CS的陋室