初入NLP领域的一些小建议

李纪为
你光明，中国便不黑暗

ACL2019投稿刚刚落幕，投稿数超过了2800篇，可以说是历史以来最盛大的一届ACL。在深度学习的推动下，自然语言处理这个子领域也逐渐被推上人工智能大舞台的最前列。

最近在跟同学的邮件、或者知乎留言中的交流中，不少同学尤其是刚入（jin）门（keng）的同学，提到了深度学习背景下做NLP科研的很多迷茫。基本可以归纳为如下几点：如今一个模型，几十行TensorFlow或者PyTorch就可以解决掉，大家不厌其烦地刷数据集的benchmark，但是因为如今实现模型的门槛低一些，SOTA很难再刷的上去；就算好不容易刷上去了，因为模型千篇一律无非修修补补，文章投出去了因为novelty 受限，文章中不中看天；即便是文章中了，似乎并无太大新意，灌水中已然迷茫。

深度算法的风靡会让研究者过度关心这些算法本身，而层出不穷模型结构的调整和改进又让我们眼花撩乱。当侃侃而谈深度学习网络结构变成一个很cool的事情的时候，人们的虚荣心会使得不约而同地忽略了几个重要点。基于我自己多年来曾经走过的弯路，踩过的坑，这篇文章做一点点小的总结。希望会对刚刚进入NLP领域的同学有所帮助。

1、了解NLP的最基本知识：Jurafsky和Martin的Speech and Language Processing是领域内的经典教材，里面包含了NLP的基础知识、语言学扫盲知识、基本任务以及解决思路。阅读此书会接触到很多NLP的最基本任务和知识，比如tagging, 各种parsing，coreference, semantic role labeling等等等等。这对于全局地了解NLP领域有着极其重要的意义。书里面的知识并不需要烂熟于心，但是刷上一两遍，起码对于NLP任务有基本认识，下次遇到了知道去哪里找还是非常有意义的。另外Chris Manning 的 introduction to information retrieval 也是一本可以扫一下盲的书，当然我认为依然不需要记住所有细节，但轮廓需要了解。IR里面的很多基本算法跟NLP有不少的重合。说说我自己曾经走过的弯路。Stanford NLP的qualification考试的一部分就是选一些jurafsky 和 manning书里面的一些chapter来读，然后老师来问相关问题。开始我一直对里面的东西懒得看，所以qualification考试一拖再拖。但博士最后一年没办法拖的时候，才发现如果早知道这些东西，博士早年可以少走很多弯路。

为什么了解NLP基础知识的重要，我给大家举几个例子。
最近跟同学一起做语言模型 language modeling相关的事情，很多同学用LSTM或者transformers做language model随手就能实现，但是实现一个 bigram 或者 trigram的language model（LM）却因为里面的OOV的平滑问题卡了大半天（熟悉的同学可能知道，需要拉普拉斯平滑或者更sophisticated的Kneser-Ney平滑）。为什么bigram 或者 trigram的LM很重要呢？去做一个语言模型的问题，实现深度模型之前，第一步其实就要去写一个 bigram 或者 trigram的LM。为什么呢？因为这些N-gram模型实现简单，并且robust。通过这样简单的实现，可以告诉你这个数据集的LM模型的下限。这样我们心里会有数，神经网络模型至少不应该比这个模型差的。神经网络模型因为其超参数、梯度爆炸等问题，有时候我们不太容易决定是真的模型不行、参数没调好还是代码有bug。那么通过N-gram LM的给出的下限，我们就可以直观地知道神经网络是有bug还是没调好参数。

random替换其实本质上属于language modeling里面基于interpolation的平滑方式， 而基于interpolation的LM平滑，就躺在jurafsky那本书的第3.4.3节。

2. **了解早年经典的NLP模型以及论文**：相比简单粗暴的神经网络模型，早年的NLP算法确实比较繁琐复杂，但里面确实有很多早年学者在硬件条件艰苦情况下的智慧结晶。熟悉了这些模型，可以在现在神经网络里面融会贯通。去年在人民大学做seminar。Seminar有大概30-40位同学参加。Seminar中，我问了一个问题，有谁知道机器翻译中的IBM模型大概是干嘛的，举手的同学大概有五分之一。我再问，谁能来手写（或者大概手写）一下IBM model1，一个人都没有。仅仅从基于IBM模型的Hierarchical Phrase-based MT, 近几年就有很多篇引用量很高的文章是基于里面的思想的。例子数不胜数：

1) chris dyer 组的Incorporating structural alignment biases into an attentional neural translation model (NAACL16) 提出用双向attention做neural机器翻译的约束项，意思是如果在英语翻译法语生成的target中的一个法语词attend到了一个source中的英语词，那么反过来，法语翻译英文target中相同这个英语词应该也attend到source中的这个英语词。其实这个思想就是完完全全相似Percy Liang 曾经的成名作之一，早在NAACL06年 Alignment by Agreement，大家通过题目的意思就可以猜到文章的内容，正向翻译与反向翻译中的 对齐(alignment) 要 一致(agree)。如今做neural MT的同学，有多少同学读过Percy的这篇大作呢 （大家知道Percy最多的应该是Squad吧）。

2) 处理对话系统的无聊回复，用反向概率p(source|target)做reranking现在应该已经是标配。再比如Rico Sennrich的成名作之一将Monolingual data 跟seq2seq 模型结合。其实这连个思想在phrase-base MT 里面早就被广发的使用。Neural之前的MT，需要对一个大的N-best list用MERT做 reranking， 反向概率 p(source|target) 以及语言模型概率 p(target)是reranking中feature的标配。

3) Harvard NLP组, Sam Wiseman 和Alex 发表的EMNLP16 best paper runner-up, Sequence-to-Sequence Learning as Beam-Search Optimization, 基本上传承了Daume´ III and Daniel Marcu 2005年的 LaSO模型，将其思想adapt到neural里面。
如果再准本溯源，诞生于neural MT的attention，不就是IBM模型的神经网络版本嘛。

3. **了解机器学习的基本模型**：神经网络的简单暴力并且有效。但是从科研的角度讲，熟悉基本的机器学习算法是必修课。比如吴恩达的 machine learning就是必要之选。记得前段时间我面试一个小伙子，一看就是很聪明的同学，而且很短的时间就有一篇NAACL在投。我就问小伙子，EM算法是什么，小伙子说没有听说过EM，而且自己的科研也用不到EM。我认为这其实是一个挺大的误区。当我想起我自己，曾经就吃过很多类似的亏。因为早期数学基础偏弱，也没有决心恶补一下数学，所以早年每次看到跟variational inference相关的算法就头大，这种偏科持续了很久，限制了科研的广度。相比粗暴的神经网络，CRF等模型的inference确实相对复杂（当年我自己也看了很多次才彻底搞明白）。但搞懂这些，是一个NLP researcher的基本素养。Pattern Recognition and Machine Learning那本书，尤其是某些小节确实比较难（又暴露了数学基础差的事实），即便是只是为了过一遍，也需要很强的耐力才能看完，更不用说完全看懂了。我自己也曾经半途而废很多次，如今依然有很多章节是不太懂的。但是其中的很多基础chapter，我认为还是很值得一读的。其实可以组成那种两三个人的学习小组，不需要有太雄伟的目标，用个一年哪怕两年的时间，把几个重要的chapter 过一遍。

NLP相对是应用科学，并不是特别的数学。但是我们天天用的算法的基本数学逻辑我认为还是需要搞懂，比如dropout, 比如天天用到的优化(SGD, momentum, adaboost, adagrad)，比如各种batch, layer normalization。这样其实可以省去很多浪费的时间，磨刀不误砍柴工。这些年来，在帮同学调bug的过程中，我至少遇见过3-5个同学 training 的时候开dropout, test 的时候没有对每个cell用 (1-dropout)去 scale （大家不要笑，这是真的）。然后画出dropout曲线就是 dropout 值越大，结果越差。在讨论的时候，同学一脸茫然并且不清楚test时候需要scale。其实本质就是并不了解dropout背后的数学原理。

4. 多看NLP其他子领域的论文：NLP有很多子领域，MT，信息抽取，parsing，tagging，情感分析，MRC等等。多多熟悉其他子领域的进展是必要的。其实不同子领域所运用的模型不会相差太大。但是最开始看不熟悉领域的问题可能会有一点难，原因是对问题的formalization不是很了解。这可能就需要多花一些时间，多找懂的同学去问。其实了解不同问题的formalization也是对领域知识最好的扩充。

知乎

我就出现过竟然在讨论班上直接把faster-RCNN讲错了的情况，以为自己看懂了，然后就讲错了（至今昱先天天还在因为这个事情调侃我）。不过重要的是，NLP领域里面一些重要的文章其实或多或少借鉴了CV里面的思想，当然也同样出现CV借鉴NLP的情况。NLP神经网络可视化、可解释性的研究，时间上还是落后于CV里面对CNN的可视化。所以很多工作大量借鉴了CV里面的类似工作。NLP运用GAN其实也是借鉴CV的。其实两个领域很多是很相通的。比如，如果不考虑question query, vision里面detection中的 region proposal（在一个大的图片背景下找一个特定区域）, 大家想是不是跟MRC里面的 span extraction （在一大堆文字里面找一个span）有异曲同工之妙。更不用说image caption generation与sequence-to-sequence模型了，本质上几乎没什么太大的区别。强化学习在生成领域generation，发完了MT(Ranzato et al., ICLR2016)再发 image caption generation, 再回到summarization. Actor-critic 模型也是类似的，还是很多做generation diversity的文章。因为跨领域不好懂，所以第一次推荐看tutorial, 如果有 sudo code 的tutorial那就更好了。另外看看扫盲课的视频，比如Stanford CS231n也是个好办法。另外，一个NLP组里面有一个很懂CV的人也很重要（拜谢昱先）， and vise versa。
graph embedding近两年崛起于data mining领域。目测会在（或者已经在）NLP的不少任务得到广泛应用。想到几年前，deep walk借鉴了word2vec, 开始在data mining领域发迹，然后似乎又要轮转回NLP了。

当然啦如何写论文也是极其重要的一环，但不是这篇文章的主题，强烈推荐清华大学刘知远老师的相关文章

zhuanlan.zhihu.com/p/58...

先写到这儿，欢迎大家补充拍砖。

香侬科技 李纪为 2019年3月11日

编辑于 15:18

自然语言处理 AI技术 人工智能