如何打造高质量的NLP数据集

原创 夕小瑶 夕小瑶的卖萌屋 2019-08-04

来自专辑
卖萌屋@自然语言处理

今天发烧睡了一天T^T，睡醒后突然想起这个都快凉透的订阅号，刷了刷知乎，刷到了这个问题

> 知乎：如何打造高质量的机器学习数据集?
> https://www.zhihu.com/question/333074061/answer/773825458

于是就有了暖暖卖萌屋的冲动(￣▽￣)

无论是做研究还是解决业务问题，做数据集都是绕不开的问题。很多刚入行的同学觉得发布一个数据集是最容易灌水的了，燃鹅如果你真的做过就会发现，随意产生一个数据集很容易，但是若以解决实际问题或让大家能在上面磕盐玩耍为目的，来产生一个能用的、质量高的、难度适中的数据集一点都不容易，超级费时费脑子甚至费钱好不好(ノ °□°）ノ ⌒┻━┻

虽然并没有刻意的研究数据集该怎么做，不过因为项目或研究需要，已经被赶鸭子上架的做了近10个数据集了，不过只是在问答、对话和一些分类问题上做过，所以像私信箱里“如何构建知识图谱”这类问题就请放过小夕吧╮(￣▽￣””)╭

由于没有很刻意的研究过这个问题，所以就分享几个个人觉得比较重要的点吧，分别是

1. 什么是高质量
2. 基本工具
3. 数据与标签来源
4. 适可而止的预处理
5. 验证可用性，尽早构造数据集迭代闭环
6. 关于复杂NLP任务

什么是高质量

刚入坑的一些小伙伴可能会以为“高质量”=“超级干净”，于是为了追求“高质量”而疯狂的预处理，最后哭了╮(╯▽╰)╭。

做数据集一般有两种动机。一种是为了research，也就是为了造福广大研究人员以及推动领域的进步；

> 不得不说SQuAD的发布对NLP这一波研究热潮的推动作用还是蛮大的

另一种，就是为了使用数据驱动的方法来优化业务指标，或解决项目中实实在在存在的问题。

这两个看似不太相关的目的背后对“高质量”的定义确是非常相近的，那就是：解决问题！

只不过，对后一种目的来说，问题一般来源于线上系统

> 一般来说，在做数据集之前一般已经存在一套系统了（为了让系统冷启动，一般先开发一套规则驱动的系统），系统上线后自然会产生日志，分析其中的badcase便可以知道哪些问题是现有系统搞不定的，这些问题就可以考虑使用数据驱动的方法来解决，于是需要做数据集了。而解决这些问题就是你做数据集的第一目标啦。

而对于前一种目的来说，问题一般来源于学术界的研究现状

> 现阶段的NLP研究多为数据驱动的，甚至说数据集驱动的。虽然这不是一个好现象，不过也不得不承认很大程度上推动了NLP的发展和研究热潮。当现有的数据集无法cover领域痛点，或无法发挥数学工具潜力，或已经被解决掉的时候，就需要一个新的数据集，更确切的说是新的benchmark了。

换句话说，还有哪些问题是行业痛点问题？或可以进一步挖掘现阶段数学工具的潜力？或现有数学工具的现发展阶段还没法很好的解决该问题？这应该是做一个高质量数据集前首先要考虑的问题。

想想2015年的SNLI[1]、2016年的SQuAD[2]、2018年的GLUE[3], CoQA[4]，再到如今的SuperGLUE[5], MRQA(https://mrqa.github.io)，都是问题驱动的，当现有数据集不足以cover问题痛点或无法满足数学工具潜力，或上一个问题已经被解决的差不多的时候，就会有新的数据集冒出来解决下一个痛点问题。

在明确要解决的问题后，数据集的质量也就保障了一半，剩下的一半就要看这个数据集怎么做啦。这里面最关键的问题是数据与标签来源的选择，以及预处理程度的把握。除此之外，迭代闭环的构建以及对复杂NLP任务的处理也会对问题解决的效率和质量产生非常重要的影响。下面开始依次介绍(~￣▽￣)-☆

基本工具

所谓工欲善其事必先利其器，只要不是太着急，在做数据集之前先掌握一些好用的工具和tricks，可以大大减少无谓的重复和低效劳动，提高迭代效率。

- **github**
  写爬虫和清洗**最原始**数据之前先在github找一下
- **正则表达式**
  文本清洗利器，不解释
- **Hadoop/Spark**
  千万级以上的语料就别去为难你的小服务器了
- **vim**
  分析样本专用。数据集只有几万或一二十万的话，vim性能一般还是够用的，不过默认的vim配置是比较鸡肋和反人类的，需要事先熟悉和配置好。要是跟vim过不去，其他带正则搜索和高亮显示的性能别太差的编辑器也ok
- **awk,grep,cut,wc等命令行工具**
  分析样本专用。数据集大了，你的vim就罢工了，当然你要是跟这些命令过不去也可以在ipython里玩，只不过写代码效率更低，而且分析结果保存起来更麻烦一些，再就是别来open(file).readlines()这种神操作就好
- **ipython + screen/tmux**
  在分析一些重要的数据集统计特性如样本长度分布时，开个vim写python脚本会很低效，数据集一大的话反复IO更是让人无法忍受的。因此开个ipython把数据集或采样的一部分数据集load进内存里，再进行各种分析会高效的多。
  另外为了避免ssh断开后从头重来，可以把ipython挂在screen或者tmux窗口里。当然啦，load进来的数据比较多时，记得时不时的del一下无用的中间结果，以免把服务器内存撑爆。哦对，记得了解一些常用的magic命令如%save，可以很方便的对复杂操作进行备份。

数据与标签来源

对数据集质量产生第二关键影响的就是数据和标签来源的选择了。其中数据可以通过人工构造、撰写的方式来产生，也可以从互联网上爬取或对公开数据集进行二次加工得到；标签同样可以人工标注，也可以远程监督的方式来获取。

人工构造和标注

最容易想到的方式就是数据和标签都来源于人工啦(￣▽￣)可惜小夕并没有资金去众包平台上帮你们积累经验(。ﾍ ﾍ)对于很多相对简单的NLP任务，数据一般在互联网上总能找到合适的，但是也有一些任务的数据很难在互联网上接触到，一般情况下只能人工精心构造（比如自然语言推理，任务型对话中的大部分子任务，分词、NER、抽取等一些序列标注任务）。如果有小伙伴想系统的学习标注，小夕推荐一本之前在图书馆刷过一半的一本书，叫《**Natural Language Annotation**》，中文名貌似叫《自然语言标注：用于机器学习》。这本书写的挺赞的，还因此怼过一次不太会标注的PM小姐姐(//▽//)\（希望她不会看我知乎hhhh

还好对于大部分nlp任务而言，基本都能从互联网上找到合适的数据源，或在已有的公开数据集的基础上加以改造就可以产生。

爬

如果要自己爬，英文语料的话可以通过国外的twitter、quora、wiki、reddit等网站按需爬取甚至直接下载，官方提供的数据获取脚本满足不了需求的话可以在github上自己搜下，基本总能找到一些奇奇怪怪的第三方爬虫绕过限制（emmm怎么有种教别人犯罪的感觉）。如果目标数据是中文，当然国内也会有微博、贴吧、豆瓣、百度百科、知乎等网站坐等被爬啦。

当然啦，Twitter、微博、贴吧这类网站的缺点就是灌水内容太多，爬完记得去github找相应的预处理脚本瘦瘦身。注意别用那些太过浮夸的脚本，处理的太干净可能会有问题，后面会讲原因噢～

改

讲真，自己爬数据真是dirty work超级超级多，尤其是你要爬的数据量灰常大或者去爬一些不那么主流的网站的时候！所以小夕更加推荐的还是先从现有的数据集想办法啦，拿来现成的然后一顿改改改绝对可以省不少力！

其实很多数据集都是这样“偷懒”做成的，比如早期Socher把只有1万样本的情感分类数据集MR[16]用parser将MR里的句子给分解为短语、子句等，再分别标注，于是就变成了20多万样本量、多粒度的SST[17]┐(￣▽￣”)┌最近也恰好刷到一篇做文本风格控制的paper[18]，同样也是用了parser，将Yelp情感分类数据集[19]拆解后疯狂加工，变成了结构->文本的风格化文本生成数据集（parser真是个造数据集的好东西）。总之，玩过一次就知道，改比爬方便多啦┐(ㄏ▽ㄣ)┌

远程监督

在打标签方面，最容易想到的当然还是花钱众包，不用说了，下一个方法。

更加经济可用的方法就是远程监督了，这方面的可玩性就非常大啦，脑洞有多大，标注质量就会有多高！

做好远程监督的前提就是提一个靠谱的假设，比如“给定一个query-answer pair，如果answer string在搜索引擎召回的某document出现，那么该document可以回答该query”，于是有了机器阅读理解数据集TriviaQA[6]、searchQA[7]；再比如“一条Twitter中包含的emoji可以反映这条Twitter的（细粒度）情感”，于是有了情感分类数据集TwitterSentiment[8]和情感可控对话生成数据集Mojitalk[9]。

如果不放心的话，自己采样一些样本，粗略统计一下你提出的假设成立的样本占比，只要大部分情况下成立就是有希望的，而后再对假设增加一些细节性的约束（比如TriviaQA里的answer必须在doc中高频出现；mojitalk里的带多媒体信息的Twitter直接丢掉，多emoji时只看最高频的emoji等），在一个靠谱的假设下，经过几番小迭代往往就可以一个能用的数据集啦。

总之，玩好远程监督也就是要掌握逆向思维，忘掉“标注”这个词，把思维改成“**握着标签找数据**“。

好啦，先休息五秒，你懂滴(↓￣▽￣)↓

适可而止的预处理

其实在做数据集这个事情上，有“洁癖”并不是一件好事，尤其是当语料的lexical diversity & semantic richness比较强的时候，一条看似让数据集更干净的正则表达式很可能

1. 沙雕了一些跟类别标签相关的有效模式，导致一些本来成立的X->Y的映射关系因此消失了
2. 减少了模型对抗噪声的学习机会，**你无法消除所有噪声，但是却消除了很多模型识别噪声适应噪声的学习机会**

这方面小夕一把辛酸泪呀，曾经花了半下午时间写了几十条清洗规则，结果model更难收敛以及开发集表现更差了。最终发现数据量和模型都不是太小的情况下，遵从最少预处理原则一般就够了，除了一些常规操作（比如滤掉HTML标签、URL、脱敏、去重、截断等），小夕一般只对如下情况进行处理：

1. 导致了“标签泄漏”，这种情况容易发生在任务简单、标签典型的场合，数据源比较多时尤其容易踩坑。比如你任务的目标是让模型通过文本语义判断情感，那就不要对emoji、颜文字手下留情了，严格控制它们在数据集中的比例。
2. 导致了样本过长，比如连续100个相同的emoji、哈、啊等
3. 样本中出现了预留的功能词（比如BERT中的[UNK],[PAD],[CLS],[SEP]之类的）

当然，如果你的数据集是生成任务相关，记得滤掉黄反内容=,=。对于一些高频错别字，一堆点点点之类的让你觉得dirty的东西，没特殊需求的话就放过它们吧。。。（真想彻底消除它们的话就换数据源啊喂，不要妄想以一人之力对抗广大人民群众产生的辣鸡！！）

验证可用性，尽早构造数据集迭代闭环

无论是人工标注的还是远程监督标注的，数据集看起来做好了不代表就是可用的，如果标注的噪声太大或者标签边界太过模糊（大量标注错误，或标注规则写的太松、太模糊，导致人都分不清某几个类别之间的区别），很可能再复杂的模型都在这份数据集上无法收敛；反之，如果数据集中有“标签泄漏”（比如你用emoji远程监督构造了情感分类数据集，最后却忘了滤掉emoji）或标签与内容有非常直接的映射关系（类别太过具体或标注规则写的太死），那就会导致一个非常简单的模型都会轻易的把这个数据集刷到近乎满分，那这个模型学到的知识基本是没有什么实际意义的，换言之，这么简单直接的任务其实几条规则几行代码就搞定了，完全没必要做数据驱动的模型训练。

因此绝对不要抱着将数据集一次做成的心态，而是要尽早构造一个**“生成数据集->跑baseline->badcase study->更新策略->重新生成数据集”**的闭环。注意，baseline别选的太麻烦（那种对各种超参敏感的模型还是算了吧），最好是已被普遍验证有效的、有开源代码的、上手轻松的、基本不用调参就效果还可以的模型（比如BERT系列）。

这里要注意侧重点，在迭代的早期，让baseline能在你的数据集上正常收敛是第一目标，中期则是关注baseline在开发集上的表现，表现太好要留意标签泄漏或数据泄漏（X中出现了Y，或忘记去重），表现太差调调参，后期则是更多关注badcase了，看看badcase中更多的是样本问题（标注噪声）还是真的模型能力不够。

关于复杂NLP任务

当然啦，上面其实都说的比较宽泛，其实在不同的NLP问题上做数据集可能会很不一样。像一些简单NLP任务如文本分类等基于上面的基本原则就差不多了，但是一些复杂NLP任务如任务型对话、知识图谱相关，哪怕完全人工产生和标注都不好做的。

比如任务型对话相关的数据集，很难使用远程监督这种偷懒的方式来构造，样本和标签的产生可能都很难脱离人力标注。有兴趣的小伙伴可以参考MultiWOZ[10]这个数据集（cover了DST、act-to-text generation和context-to-text generation这三个任务型对话中的子任务）的paper，里面对machine-machine（如M2M[11]）、machine-human（如DSTC系列[12][13][14]）、human-human（如ATIS[15]，WOZ系列[10]）这三种协同构造任务型对话数据集的方式总结的很到位，会让你感受到产出一个高质量的任务完成型对话数据集是一个很有挑战的工作，自己从头摸索的话可能到头来只会收获一脸懵逼ヽ(￣▽￣””)ノ

所以面对一些比较复杂的NLP任务的时候，一定一定要记得先精读一下最新最权威的数据集的paper，这类数据集的构建经验可能整个微信和知乎也找不到几篇的噢ヽ(ﾟ▽ﾟ)ノ

参考文献

[1] Bowman S R, Angeli G, Potts C, et al. A large annotated corpus for learning natural language inference[J]. arXiv preprint arXiv:1508.05326, 2015.

[2] Rajpurkar P, Zhang J, Lopyrev K, et al. Squad: 100,000+ questions for machine comprehension of text[J]. arXiv preprint arXiv:1606.05250, 2016.

[3] Wang A, Singh A, Michael J, et al. Glue: A multi-task benchmark and analysis platform for natural language understanding[J]. arXiv preprint arXiv:1804.07461, 2018.

[4] Reddy S, Chen D, Manning C D. Coqa: A conversational question answering challenge[J]. Transactions of the Association for

Computational Linguistics, 2019, 7: 249-266.
[5] Wang A, Pruksachatkun Y, Nangia N, et al. Superglue: A stickier benchmark for general-purpose language understanding systems[J]. arXiv preprint arXiv:1905.00537, 2019.
[6] Joshi M, Choi E, Weld D S, et al. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension[J]. arXiv preprint arXiv:1705.03551, 2017.
[7] Dunn M, Sagun L, Higgins M, et al. Searchqa: A new q&a dataset augmented with context from a search engine[J]. arXiv preprint arXiv:1704.05179, 2017.
[8] Go A, Bhayani R, Huang L. Twitter sentiment classification using distant supervision[J]. CS224N Project Report, Stanford, 2009, 1(12): 2009.
[9] Zhou X, Wang W Y. Mojitalk: Generating emotional responses at scale[J]. arXiv preprint arXiv:1711.04090, 2017.
[10] Budzianowski P, Wen T H, Tseng B H, et al. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling[J]. arXiv preprint arXiv:1810.00278, 2018.
[11] P Shah, D Hakkani-Tur, G Tur, A Rastogi, A Bapna, N Nayak, and L Heck. 2018. Building a conversational agent overnight with dialogue self-play. arXiv preprint arXiv:1801.04871.
[12] Jason Williams, Antoine Raux, Deepak Ramachan- dran, and Alan Black. 2013. The dialog state track- ing challenge. In Proceedings of the SIGDIAL 2013 Conference, pages 404–413.
[13] M. Henderson, B. Thomson, and S. J. Young. 2014b. Word-based Dialog State Tracking with Recurrent Neural Networks. In Proceedings of SIGdial.
[14] Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014c. The third dialog state tracking challenge. In Spoken Language Technology Work- shop (SLT), 2014 IEEE, pages 324–329. IEEE.
[15] Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The atis spoken language sys- tems pilot corpus. In Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania
[16] B. Pang, L. Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with re- spect to rating scales. In Proceedings of ACL 2005.
[17] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng, C. Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Tree- bank. In Proceedings of EMNLP 2013.
[18] Oraby S, Harrison V, Ebrahimi A, et al. Curate and Generate: A Corpus and Method for Joint Control of Semantics and Style in Neural NLG[J]. arXiv preprint arXiv:1906.01334, 2019.
[19] Zhang X, Zhao J, LeCun Y. Character-level convolutional networks for text classification[C]//Advances in neural information processing systems. 2015: 649-657.