R&S[25] | 搜索中的意图识别

原创 机智的叉烧 CS的陋室 2020-05-07

China-X

徐梦圆 - Change

往期回顾：

- R&S[24] | 浅谈Query理解和分析
- R&S[23] | 搜索系统中的纠错问题
- R&S[22] | 搜索系统中的召回
- R&S[18] | SIGIR2018：深度学习匹配在搜索与推荐中的应用
- R&S[17] | 手把手搞推荐[6]：回顾整体建模过程

曾经谈过实体识别、纠错等内容，这次给大家完成query理解的最后一块拼图——意图识别。

为什么要做意图识别

意图识别算是query理解中比较上有的位置，对query进行意图识别，是指分析用户的核心搜索需求，例如是要找电影、找小说，还会想问百科知识，还有查快递、市政办公等需求，这些需求在底层的检索策略会有很大的不同，错误的识别几乎可以确定找不到能满足用户需求的内容，导致产生非常差的用户体验，因此精准的意图识别非常重要。

来举一个例子，用户输入唐人街探案，一般是电影、网剧、网评之类的需求，退一步可能是比较弱的新闻、明星之类的意图，而肯定不是汽车、体育、快递之类的意图。而在底层的检索策略，只有我们识别到了电影意图，我们才会去电影的数据库里面检索，里面就不会有汽车、体育、快递，换言之，没有识别到电影意图，压根就不可能出现唐人街探案的电影的卡片甚至咨询，而如果我们错误的识别到了汽车，那我们就只会在这个意图下出汽车的相关内容了。可想而知，错误的意图识别会带来多大的负面影响。

意图识别的具体内容

其实从上面的描述可以看到，意图识别是对query进行意图的分类，因此总体思路上采用文本分类的方式进行，这个相信大家能想的明白。而在架构上，也需要对意图识别的逻辑进行设计。

意图识别的架构设计

考虑搜索场景的下面这些原因，进行意图识别的架构设计：

- 用户输入内容模糊或者涵盖内容广，因此经常会出现多意图，如“苹果”（手机、水果）、“长城”（景点、电影、汽车）、口罩（医疗、购物）。
- 多个意图不可能同时开发上线，必须一个一个开发迭代。
- 开发一个意图时要求独立，尽可能不影响其他意图的计算。
- 虽然每个意图独立，但是要求意图之间尽可能可比（例如意图强弱打分），这个信息需要传导到排序层。
- 每一个意图的识别算法可能会不同，需要分别设计。

当然上面的所有很难兼顾，我们一般根据搜索系统当前的情况进行设计，按照《美团机器学习》的建议，采用多个二分类器并联的方法能够较好的保证框架拓展的完整性，一个新意图的上线对其他意图的影响降到最低，但是在可比性上可能会较欠缺。

意图识别的方法

虽说是文本分类，但是大家千万不要把这个分类限制在基于深度学习和机器学习的文本分类里面了，在搜索场景下，这些方法很可能会失效。大家拓宽思路，也多想想没有深度学习之前大家可以怎么做意图识别，手里多几把武器，刀子切菜，斧头砍柴，锤子锤钉子，能解决的问题就多了。

基于词典和规则的方法

基于词典和规则的方法在搜索中最为常见，好的词典一般能够解决超过80%甚至以上的问题，而准确率也能达到90%甚至以上，这应该是搜索领域最应该首先想到的方法。

- 搜索query绝大部分很短，没有上下文、特征词，识别非常困难，如“都挺好”（电视剧），用户就这么输入，没有额外信息，除了词典难以处理。
- 词汇含义变化很快，随着一些热点信息的变化，需要快速上下线，词典可以处理，如“少年的你”，识别到是新电影，马上更新词典，意图就可出。
- 语义信息有的时候反而是误导。“少年的你”是一部电影，“年轻的你”可就不是了。
- 部分的搜索要求有限保高准确，低召回可接受，词典就是为了这个场景量身打造的。

基于词典，就是对用户query内容和词典内容进行比对，比较常见的方式是序列标注问题里用的最大逆向匹配，通过这种方式找到词典里的实体词，匹配触发了自然就有意图了，而在词典匹配的时候，不是使用链表之类的来构造匹配结构，而是使用搜索树的结构，这种匹配的复杂度最低，速度也快，两者结合，其速度甚至比很多模型要快得多（基本上1ms以内就能完成），fasttext速度非常快，但是textcnn之类的其实就已经达到ms级别以上，bert甚至在10ms级别。

基于规则，其实和词典的区别并不大，但是处理query会比词典更为灵活，例如正则表达式，相比词典能处理更加复杂的内容，构造对应的模板，query命中这个模板就触发意图，这种当时依赖个人对query的理解，一般结合词典来使用。

基于ML和DL的方法

机器学习和深度学习就这么一无是处？当然不是，他也有存在的意义，把他搬出来往往是因为这些原因。

- 规则和词典的泛化能力不足，召回率达不到预期。
- 用户在这个意图下输入的内容比较复杂，规则和词典难以覆盖：如“劳动节去哪玩”可以是一个电影意图，这里没有指向意图的实体词。
- 可以依赖语义进行识别的内容，例如词典里没有“让子弹飞”，但是用户输入了“让子弹飞的票房怎么样，一些修饰词之类的可以帮助进行意图识别，此时语义，也就是深度学习的方法，可以协助完成意图识别。

有关文本分类，这块相信绝大部分人都了解的不少了，简单的就是fasttext，复杂一点可以升级为textcnn。但是注意几个点：

- rnn系列并不推荐，一个是耗时，另一方面query的内容很短并不适合抽取序列信息。
- bert之类的大家伙，使用时需要谨慎，耗时虽然要考虑，但是性价比、提升空间之类的也要好好分析，fasttext效果很差，很多时候bert也不见得好到哪里去。

大模型想用其实也有方法，可以考虑用蒸馏的方式来寻求替代方案，2015年（不知道有没有记错）Hinton提出了知识蒸馏的概念，首次应用在图像处理上，至今已经过了5年（其实我也比较好奇为啥工业界用的，媒体吹的都不是很多），而有一些论文在nlp上进行了尝试，效果尚可，现在科研界有很多尝试对bert进行轻量化的方案，蒸馏就是其中一个，大家可以多尝试一下。

其他定制化方法

因地制宜、因时制宜，因XXX制宜，都在告诉我们，要从实际出发，实事求是，通识通法是帮助我们解决大部分问题的尚方宝剑，但是在特殊情况下也需要我们采取特殊手段解决问题，在遇到这些问题的时候，我们也要懂得应对，类似的说法我在之前的文章也有提到过，这是作为算法工程师的其中一个凸显能力的重要素养。来举一个特殊的场景，供大家思考：

- 如何识别用户输入的是一段歌词、台词。
- 用户对热点事件的关注点不同，描述不同，如何匹配到特定的热点新闻上去。
- 图书的模糊搜索如何实现。

这些问题如何解决，取决于你对用户query的理解，多看日志，多看用户是怎么尝试描述他们想要的东西的，才能让你更好的解决这类问题。