Query意图方法（1）- 基于片段意图

原创 XG数据 WePlayData 2019-04-08

现有搜索系统主要基于关键词匹配的方式返回搜索结果。通常query 比较短且歧义大， 使得按关键词匹配可能返回一些不相关的文档，不能真实反映用户真实需求。 比如 query“乡村爱情 10”， doc“一段发生在美丽乡村里的爱情”，虽然 query和doc 完全字面匹配，但是从语义来看，doc并不和query相关。 因此理解query 的意图是搜索中一个重要的模块。 一方面从意图角度计算query和doc匹配度，缓解字面匹配的问题，另一方面可以帮助触发相应意图的 box。

现有 query 意图识别方法主要分为两种，一种是挖掘意图规则和模版，比如符合模版 “**在线电影观看”的 query 存在视频意图。 这种方法识别准确率高，但覆盖率不足，同时挖掘模版也是一个繁琐的过程，发现和制定模版需要较多的人工参与， 很难实现自动化。另一种方法把意图识别当作一个短文本分类，提取bag of words特征或者语义特征，然后利用贝叶斯模型或 CNN 模型对 query 按照预定义意图类别进行分类。这种方法有一定的泛化性，但是需要大量的标注数据样本，模型更新也比较困难。而且存在很多短 query，比如“周杰伦”，“回头太难”， 提取字面特征太过稀疏， 语义特征不够准确， 很难通过分类正确的识别出意图， 尤其是多意图。

本文基于现有意图识别方法的优缺点，提出一种基于query片段的意图识别方法，把 query 意图识别转化成意图片段的离线挖掘问题，减轻了在线计算复杂度，并且通过意图片段更新可以很快速的解决新query的意图识别。意图片段在一定程度上类似模板，但相比模板挖掘方法更加简单，泛化性更好。

比如要计算query“信用卡取现手续费？”的意图，如果知道query的成分片段的意图，如“信 用卡”，“信用卡提现”，“信用卡手续费”的意图分布，那么query的意图可以由这些片段的意图分布推导得到。其中片段是指query分词后，任意n个词的有序组合，词之间不要求在query中紧邻出现。此时问题转成如何离线计算意图片段的意图分布。

1） 离线片段意图挖掘

首先获取片段在搜索上的结果，下图给出了片段“信用卡取现”在百度的结果：

分析搜索结果，我们可以从两方面来判断片段的意图分布。一方面很多url都有明显意图的，比如图中的“51credit.com”，“finance.qq.com”，如果能够知道url对应的意图，那么从url角度可以计算和目标意图类的url意图匹配度url_match；另一方面，可以从返回的doc标题来计算每个doc标题和目标意图类的语义相似性$title_match$。最后通过url_match和$title_match$来计算片段在每个意图类上的得分。其中frag指片段，c为目标意图类，$frag_qv$指片段的qv，用于衡量片段的热度，w1和w2分别是两部分的加权系数。

$$score(c|frag) = (1 - \frac{1}{log(frag_{qv})}) * (w1 * url_{match} + w2 * title_{match})$$

下面具体介绍如何计算url_match和$title_match$。

a) url_match

$$url_{match} = \sum_i indicator(url_i, c) * pos(i)$$

其中，$indicator(url_i, c)$是一个0-1函数，表示如果url_i是属于意图类c，则为1，反之为0，$pos(i)$为位置惩罚函数。这里的url只需要挖掘每个意图类中头部的一些url即可。

b) $title\_match$

$$title_{match} = \sum_i sim(title_i, c) * pos(i)$$

$$sim(title_i, c) = \max_k cosine(title_i\_vector, word_k\_c\_vector)$$

在$sim(titlei，c)$计算中，对于每个意图c类，挖掘意图类c中最相关的一批词。基于word2vec，分别计算doc标题和c类中的每个词向量的余弦相似度，然后取最大的作为doc标题和目标意图c的语义相似性，$pos(i)$为位置惩罚函数。

2） 在线query意图推导

$$score(c|query) = \sum_{frag \in query} w_{frag} * score(c|frag)$$

通过query片段的意图分布加权求和，可以计算query在意图类上的得分，其中w_frag是片段的重要性。接着需要将得分转化成意图类上的概率分布，相比于直接求比例，这里引入每个意图类中的片段最大得分做归一化，解决每个意图类上得分都很低时，概率计算不置信问题。比如query在3个意图类abc上的score都为0.1，不平滑时p(a|query)=0.33，如果采用下述公式进行平滑，则p(a|query)=0.074，相比于不平滑计算的概率更加合理。其中α调节平滑比例。

$$p(c|query) = \frac{score_{q-c}}{\alpha * \sum_i score_i + (1-\alpha) * ln(\frac{max(score_c)}{score_{q-c}})}$$

3） 高频query意图挖掘

上面的方法主要是根据搜索展现进行挖掘。对于高频query，其点击信息比较丰富，点击相比展现更能反映用户的意图。因此对于高频query，在拥有用户点击数据的情况下可以统计其点击的url分布来计算其意图分布。

4） 专有实体补充

枚举生成的query片段覆盖率不全，还可以引入每个意图类特有的一些实体资源增加覆盖率。比如，购物意图类中的商品名，品牌名等，音乐意图类中的歌手，歌曲，专辑等。

相关阅读

1. Query理解 - 搜索引擎“更懂你”
2. 搜索引擎新的战场 - 百度、头条、微信
3. 当我们关注舆情系统时，我们忽略了什么？
4. 搜索引擎的两大问题（1） - 召回
5. 搜索引擎的两大问题（2） - 相关性
6. Query词权重方法（1） - 基于语料统计
7. Query词权重方法（2）- 基于点击日志