topicDict算法介绍

韩云飞

topicDict是一种对作文的切题程度进行量化评判的算法，算法思想是在工程实践过程中逐渐成熟的，下面对其进行简单的介绍。

1．主要过程：

1） 对每篇essay进行分词，过滤，[词根还原]，得到可以代表essay的词组tokens；

2） 收集所有作文的tokens到一个dictionary中,dictionary中每个token有两项：

[token : {frequency, goal}]

(frequency即token在所有作文的tokens中出现的次数，goal = frequency/所有tokens总数，代表该词语对主题的贡献度。)

3） 对一篇作文onTopic分数由如下公式得到：

onTopic_goal(essay) = $\sum_{i=1}^{n} \text{dictionary}[\text{token}_i][\text{goal}]$ （n表示essay中tokens数目）

4）上面的公式对于作文长度相近的情况是适用的，但仍倾向于长作文得分高(作文越长，贡献的tokens就越多)。 所以需要对作文的长度进行惩罚，修正的公式为：

onTopic_goal(essay) = $(\sum_{i=1}^{n} \text{dictionary}[\text{token}_i][\text{goal}]) \div n$

2．算法分析：

通常的跑题判别做法：

对于一个题目，我们假设其对应的主题词汇为(T_1,T_2,T_3 ..., T_n)， 则每篇作文都可以提取为n个特征：<T_1频数，T_2频数，...， T_n频数>，然后以这n个特征作为输入，用某种方法对作文的切题情况进行评判。

这类方法存在两个问题：

1） 不可能找到所有的主题词汇，而且特征空间可能会很大；

2） 主题词之间没有权重区分，不同特征的贡献度无法体现。

topicDict和这一类方法最大的不同是：

1）不关心作文题目到底是什么；

2）不对每篇essay进行特征提取，而是把整个待评作文集作为处理对象，得到一些全集上的信息(特征)，然后利用这些信息对每一篇作为进行评价；

3）得到的结果是量化的，可以刻画出作文的切题程度。

根据所有待评作文的用词构造词典，并根据词语的频数得到词语对主题的贡献度(goal)。这里有个隐含的前提是词语出现的次数越多(被多篇作文使用)，与作文主题越相关。

topicDict在作文集越多时，效果会越好。 如果作文集过少，可能会发生主题偏移的情况：即作文的主题是A，大多数作文体现的主题是B。 但这种情况只有在作文集很小的情况下才会发生。

3．下一步工作：

1）在topicDict的基础之上，把作文题目纳入考虑范围(作文题目具有多种多样的形式，会比较难处理)；

2）设计好的测试用例对topicDict进行评估，和一些现存的方法进行比较；

3）如果验证该算法的效果较好，可以尝试将其提炼为更加通用，普适的算法。