数据工程 (一)

作者：Gaoshengyue
原文链接：https://ld246.com/article/1577762462564
来源网站：链滴

概述

一般拿到这种需求的工程师,都是非常头疼的。因为啥呢，大部分需要提取关键词的数据，都是非常杂、毫无章法的。比如爬虫啊、数据结构化啊，都非常需要提取关键词这一步，我也是。。。哈哈，所简单利用分词、词频分析啊乱七八糟的东西，简单写一个ｄｅｍｏ。当然，关键词重要与否，有效与，是需要具体数据分析的，在这里就不详细说数据分析了，后面会详细写一篇文章讲一下。

关键词提取步骤

按照逻辑上的惯例，除非是人工标注，否则我们是很难提取到想要的信息的。但是人工标注，干过的都知道，实在是太累了，而且效率极低。而且模板的提取方法是**余弦聚类**，我个人认为是效率比较低一种方法。（当然我也这么干了一阵。。）

所以，开始使用特征语言处理ＮＬＰ一类的东西，去尝试做这一事项。

分词

<hr>
分词没太多可说了，现在已经有很多比较成熟的分词库了，jieba之类的应用已经很广泛了。不过具体还在分词的使用上，例如根据各个词性啊怎么组合啊，词频怎么推导啊乱七八糟的，这块不赘述了。

词频分析

<hr>
这里我们可以直接调用jieba的cut去做分词，将每个词组合成的语料库矩阵做排列，拿到每个词的词，在去做排序就可以了。不过这里的词频分析并不代表这个词汇是可用的，因为某些无用词也是会有高的出现频率。
这个时候我们需要去找一些行业词库，比如金融、IT、新闻。
我这里的方法是根据行业词库与语料库矩阵做交集处理，交集包含度高的预料拿到语料库中做最终的

配。
这里我们可以用TF-IDF来训练模型进行词频统计，匹配。

TF-IDF = TF * IDF，其中TF（Term Frequency）表示一个词在文档中出现的次数。DF（Document requency）表示整个语料库中含有某个词的文档个数。IDF（Inverse Document Frequency）为逆档频率，其计算公式为：IDF= log(语料库中文档总数/(包含该词的文档数+1))，+1 的作用是做平滑理。

有效数据拆分

<hr>
这里就要简单的多了，根据我们匹配出的语料库，可以根据keywords匹配出我们的语句是否含有或偏向我们的行业方向，如果偏向，那我们可以根据设定的阈值与keywords做配合，提取数据。具体阈值需要看业务需求了，还有行业设定，并不是每一个阈值都代表了所有。
有效数据提取出来后，并不是说这条数据就有用了，接下来就是关键的结构化。

结构化

<hr>
目前探索到的结构化方法有两种，其实是有很多种。

第一种是匹配模板，这个方法是比较耗时的，也就是我开头所说的人工标注，这样匹配的结果是最精的，但是相较于我们快速提取数据的需求来说，效率上过于低下，而且能用度不高。

比如我有100亿条数据，人工设定了100个模板智能提取出来1000W条数据进行结构化，我们能说剩的99亿9000万条数据是无用数据吗？人工的话100个模板都够累了，就算用上余弦聚类方法，也是杯车薪啊。

所以我比较推荐的是**第二种**，也就是目前常说的NLP处理，现在很多成熟的库都是有关键信息提取的就算是业务比较复杂，也可以根据比较通用的正则去做一部分提取。

最高难度的还是我们要为文本做分类，要提取出关键数据，我们还是需要根据语义去做分析，我们到是要拿到哪些数据？到底是需要怎么做？

数据分析步骤

初步分析方法

<hr>
这里分两种，一种是已经结构化的数据分析，承接上一步结束，另一种是非结构化数据分析，也就是键词分析。
这里推荐个库，missingno，这个库看数据的缺失程度和关联性还是很不错的，不过具体的分析还是要pandas和jupyter-notebook类似的平台去做了。

数据缺失程度

<hr>
这一步主要是看我们的数据价值，覆盖率。比如说我们拿到了某10个同学的数学考试成绩，然而我们有其中7个同学的数据，那我们怎么去求平均分呢？

所以说数据缺失程度的分析必不可少，做了这一步也方便我们做下一步的特征处理。

数据关联性

<hr>
这一步也至关重要，例如我们想知道某个同学理综一共考了多少分(原谅我还停留在分文理的年代)，我们必然要拿到物理、生物、化学的各科成绩才行，我们没有生物成绩，能知道他的理综成绩吗？必不可能，这就是强制关联。

数据特征处理

<hr>
这里要结合我们分析数据缺失程度的一步，而且要判断是否需要做数据特征处理。
例如我们要求全班同学数学成绩的平均数，然而缺少一个同学的数据，那我们就可以根据众数、中位、平均数等方法，为他填充一个数据，前提是不影响整体的结果的情况下。

数据分布

<hr>
数据分布是我们从小学就开始学习的了，比如男生多少人，女生多少人。
男生多少个90分以上的，女生多少个90分以上的。
简单来说，我们要做的分布也就是这种情况，根据分布情况可以大概率的淘汰一批数据，或者说挖掘这批数据的其他特性。

简单demo

github上有一个简单的demo，结合fastapi简单写了一个。

具体使用方法见github的Readme吧，thank you

https://github.com/Gaoshengyue/Semantic_cs

老铁门有做数据处理的可以一起交流下。