基于文本挖掘应用的研究

摘要

近年来，随着互联网的广泛应用，网络问政平台已经成为政府了解民意的重要渠道。因此运用文本挖掘对留言划分和热点整理的效率有显著的提升。

对于问题 1，首先循环读取 excel 里面的单元格文本，利用 Python 中文分词组件 jieba 对留言内容进行分词，载入停用词表，除去无用词，再利用 Tensorflow 构建深度学习框架，实现 word2vec 模型，训练词向量，得到词典并计算留言中一级分类关键词相关联的词的出现概率，提取出关键词，最后进行分类操作后，保存完成归类的 excel 文本。

对于问题 2，利用对附件 3Excel 进行去重后，通过之前建立的模型对其中留言进行归类整理，得出热点问题表以及热点问题留言明细表。

对于问题 3，根据研究结果，分析附件 4 中对留言信息答复的相关性、完整性、 可解释性等做出评价方案。

关键词：分词　word2vec 模型　Tensorflow　去重　热点问题

目录

1、问题重述 ……5

2、分析与过程 ……5

2.1 问题 1 分析与过程 ……5

2.1.1 流程图 ……6

2.1.2 文本预处理 ……7

2.1.3 计算文本相似度 ……10

2.1.4 留言详情分类 ……11

2.1.5 模型的评价 ……11

2.2 问题 2 分析与过程

2.2.1

2.2.2

2.2.3

2.2.4

2.3 问题分析与过程

2.3.1

2.3.2

2.3.3

2.3.4

3、结果分析

3.1 问题 1 结果分析

3.2 问题 2 结果分析

3.3 问题 3 结果分析

4、结论

5、参考文献

1、问题重述

此次建模的目标是利用 tensorflow 框架构建神经网络，并实现 word2vec 训练词向量、python 中文分词工具 jieba 对群众留言信息进行分词、提取关键词和循环分类达到以下三个目的：

1) 根据附件二的留言详细进行一级分类，提高查准率。

2) 减轻了群众留言信息分类工作的巨大压力，极大程度的提高了工作效率。

3) 通过文本挖掘解析留言信息并统计相关问题，从中找到重要的热点问题予以关注解决，对良好的社会风气形成有极大的好处。

2、分析与过程

2.1 问题 1 分析与过程

2.1.1 流程图

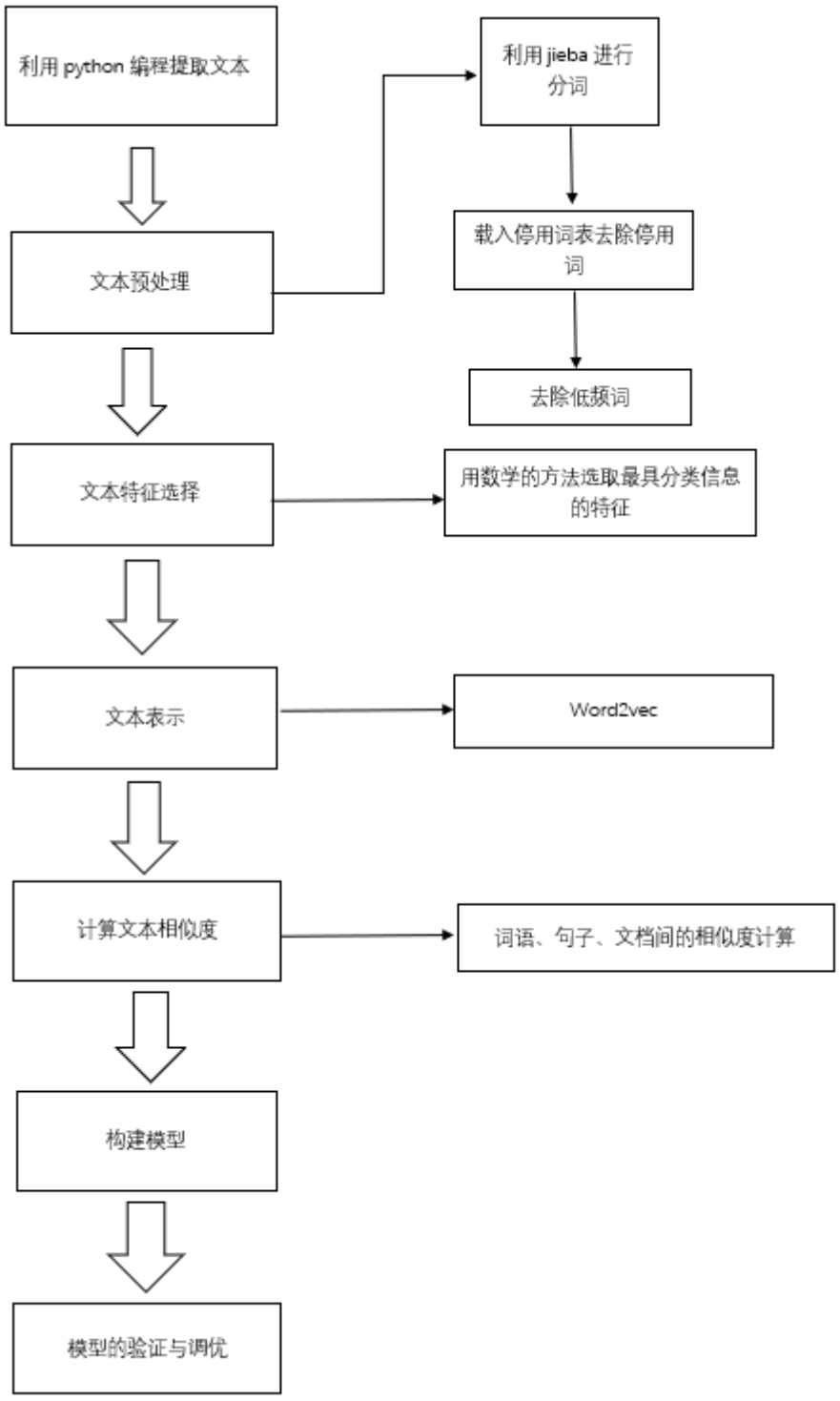
利用 python 编程提取文本
利用 jieba 进行分词
文本预处理
载入停用词表去除停用词
去除低频词
文本特征选择
用数学的方法选取最具分类信息的特征
文本表示
Word2vec
计算文本相似度
词语、句子、文档间的相似度计算
构建模型
模型的验证与调优

图 1：总流程图

2.1.2 文本预处理

2.1.2.1 文本信息的提取

在题目所提供的附件中除需要用到的留言详情外，还有很多无效数据，在进行文本分类时，仅仅用到留言详情，因此需要进行预处理，先将 Excel 文本中留言详情一列读入 Python 程序中，再提取后需要在其后加上一级分类信息，利用循环结构提取留言详情文本，依次进行分类。

2.1.2.2 对文本进行分词、去停用词

对文本进行挖掘分析前，首先要把文本信息转换为计算机能够识别的结构信息。为了操作便捷，利用 Python 中文分词模组 jieba 对这些留言详情进行中文分词。jieba 分词系统是基于前缀词典实现词图进行扫描，生成句子中汉字所有可能成词情况所构成的有向无环图（DAG），采用动态规划查找最大概率路径，找出基于词频的最大切分组合；对于未登录词，采用基于汉字成词能力的 HMM 模型，采用 Viterbi 算法进行计算。

分词之后，需要去掉无效的分词，例如标点符号等，同时载入停用表，对比停用表去掉停用词。

2.1.2.3 构建 word2vec 模型

在对留言详情进行分词后，把词语转换为向量，使用 word2vec 模型，训练词向量，并进行文本特征提取。

Worde2vec 是一个词向量计算模型：输入大量已分词文本；输入用 word2vec 的详细实现，简而言之，就是一个三层的神经网络。要理解 word2vec 的实现，需要的预备知识是神经网络和 Logistic Regression。

上图是 Word2vec 的简要流程图。首先假设，词库里的词数为 10000; 词向量的长度为 300（根据斯坦福 CS224d 的讲解，词向量一般为 25-1000 维，300 维是一个好的选择）。下面以单个训练样本为例，介绍每个部分的含义：

（1）输入层：输入为一个词的 one-hot 向量表示。这个向量长度为 10000。假设这个词为 ants，ants 在词库中的 ID 为 i，则输入向量的第 i 个分量为 1，其余为 0。[0, 0, ..., 0, 0, 1, 0, 0, ..., 0, 0]

（2）隐藏层：隐藏层的神经元个数就是词向量的长度。隐藏层的参数是一个[10000 ，300]的矩阵。 实际上，这个参数矩阵就是词向量。回忆一下矩阵相乘，一个 one-hot 行向量和矩阵相乘，结果就是矩阵

的第 i 行。经过隐藏层，实际上就是把 10000 维的 one-hot 向量映射成了最终想要得到的 300 维的词向量。

（3）输出层：输出层的神经元个数为总词数 10000，参数矩阵尺寸为[300，10000]。词向量经过矩阵计算后再加上 softmax 归一化，重新变为 10000 维的向量，每一维对应词库中的一个词与输入的词（在这里是 ants）共同出现在上下文中的概率。

（4）训练：训练样本（x，y）有输入也有输出，我们知道哪个词实际上跟 ants 共现，因此 y 也是一个 10000 维的向量。损失函数跟 Logistic Regression 相似，是神经网络的最终输出向量和 y 的交叉熵（），最后用随机梯度下降来求解。

上述步骤是一个词作为输入和一个上下文中的词作为输出的情况，引入实际训练时的两个模型 skip-gram 和 CBOW：

（1）skip-gram： 核心思想是根据中心词来预测周围的词。假设中心词是 cat，窗口长度为 2，则根据 cat 预测左边两个词和右边两个词。这时，cat 作为神经网络的 input，预测的词作为 label。

（2）CBOW（continuous-bag-of-words）： CBOW 模型指用周围的所有词来预测中心词。每一次中心词的移动，只能产生一个训练样本。

两个模型相比，skip-gram 模型能产生更多训练样本，抓住更多词与词之间语义上的细节，在语料足够多足够好的理想条件下，skip-gram 模型是优于 CBOW 模型的。在语料较少的情况下，难以抓

住足够多词与词之间的细节，CBOW 模型求平均的特性，反而效果可能更好。

其中采用了负采样的思想，最终神经网络经过 softmax 输出一个向量，只有一个概率最大的对应正确的单词，其余的称为 negative sample。现在只选择 5 个 negative sample，所以输出向量就只是一个 6 维的向量。要考虑的参数不是 300 万个，而减少到了 1800 个，大大提升了运算效率。可以利用下面这个公式选择负采样的词。

其中 f(w)是词频。可以看到，负采样的选择只跟词频有关，词频越大，越有可能选中。

最后通过 tensorflow 来实现 word2vec 模型，构建 word2vec 完成后，我们就可以进行训练词向量和文本特征提取了。训练词向量后，我们从 excel 里面提取的留言详情的分词就可以计算机所读取，然后我们再进行文本特征提取得到最具分类信息的特征。

2.1.3 计算文本相似度

2.1.4 留言详情分类

通过计算词频，根据词频构建循环语句，完成一级分类。

2.2.5 模型的评价

参考文献：

1、Le Q V, Mikolov T. Distributed Representations of Sentences and Documents[J]. 2014, 4:II-1188.

2、Mikolov T, Sutskever I, Chen K, et al. Distributed Representations of Words and Phrases and their Compositionality[J]. Advances in Neural Information Processing Systems, 2013, 26:3111-3119.

3、Word2Vec Tutorial - The Skip-Gram Model

4、Udacity Deep Learning

5、Stanford CS224d Lecture2,3

6、https://www.jianshu.com/p/f38b39de9667h

停用词表(stop_word.txt)：

ttps://github.com/multiangle/tfword2vec/commit/013713c22ce8f5263c0542ab871dacb65a9138aa?diff=unified#diff-34eb45d8142b1e253d440cd996947adc