

**本科毕业设计说明书**

**基于多用户协同的微博短文本主题获取**

TOPIC ACQUSITION OF MICROBLOG SHORT TEXT BASED ON MULTI-USER COLLABORATION

学院(部)： 计算机科学与工程

专业班级： 计算机科学与技术15-3班

学生姓名： 杨红喜

指导教师： 张柱讲师

2019年 5月 28日

基于多用户协同的微博短文本主题获取

# 摘要

随着社交网络的兴起以及微博自媒体时代的到来，互联网上每天能产生数以亿计的博文。海量的微博文数据蕴含了丰富的有关个人、社会、企业、政府多维度、多层次、多样化的信息。如何通过信息融合提高综合信息资源的获取效率已成为研究热点。以微博为例，微博是一个新的多媒体迷你博客，允许用户为所有或有限群体发布短文本信息，近年来，微博以惊人的速度风靡中国，并产生了大量短文本，数据结构不同的短文本，其中包含了大量有用的信息。另外微博是基于多用户的社交平台，它的信息相比其他平台是公开化的，这无疑使得微博成为了数据挖掘的一个热门平台，也逐渐成为了研究的热点。但是，类似于微博，论坛的回复信息，即时通讯和电子邮件的内容较少，从几十个单词到一百个单词左右不等。与长文本中的特征选择技术相比，短文本选择的主要问题是,在短文本中，特征空间稀疏，难以充分利用其特征之间的相关性。

本文在参考了大量对微博主题提取的科研文献基础上，解决了以下问题。对微博网页进行url分析，用scrapy爬虫爬取微博短文本内容，然后以微博短文本为研究基石，进行了进一步的深入探讨数据获取和信息挖掘的必要性，对短文本进行了数据清洗，停用词过滤等，从众多的短文本中，提取出了最受用户关注的词汇，并通过一定数量的词汇来刻画出各个主题，完成主题获取。

对博文进行内容分析，主题词提取，完成对博文中蕴含的主题的分析等，有重要的理论研究价值和应用价值，为了对主题词大限度的应用，可以将主题词进行分类，方便检索和使用。本实验中我们将借鉴相关算法和爬虫的相关技术，构建自己的主题词提取算法。本文第二节，第三节，第四节中首先介绍常用的不同的主题提取方法，第五节则对比他们的优缺点，第六节进行相关实验验证，最终则是在全文最后进行了相关总结。

**关键词：微博，scrapy，信息挖掘，数据清洗，主题获取**

**TOPIC ACQUSITION OF MICROBLOG SHORT TEXT BASED ON MULTI-USER COLLABORATION**

# ABSTRACT

With the rise of social networks and the advent of the era of micro-blogging from the media, hundreds of millions of blog posts can be generated on the Internet every day. The vast amount of micro-blog data contains abundant information about individuals, society, enterprises and the government, which is multi-dimensional, multi-level and diversified. How to improve the efficiency of comprehensive information resources acquisition through information fusion has become a research hotspot. Take micro-blog as an example, micro-blog is a new multimedia mini-blog, which allows users to publish short text information for all or limited groups. In recent years, micro-blog has become popular in China at an alarming rate, and produced a large number of short text with different data structures, which contains a large number of useful information. In addition, micro-blog is a multi-user social platform, and its information is more public than other platforms. This undoubtedly makes micro-blog a hot platform for data mining, and has gradually become a research focus. However, similar to microblogging, forum response information, instant messaging and e-mail content is less, ranging from dozens of words to about 100 words. Compared with feature selection technology in long text, the main problem of short text selection is that in short text, feature space is sparse and it is difficult to make full use of the correlation between features.

On the basis of referring to a large number of scientific research literature on microblog topic extraction, this paper solves the following problems. After URL analysis of microblog web pages, scrapy crawler is used to crawl the short text content of microblog. Taking the short text of microblog as the research cornerstone, the necessity of data acquisition and information mining is further discussed. The short text is cleaned up and the stop words are filtered. From the numerous short texts, the words most concerned by users are extracted, and a certain number of words are used. To describe the various topics, complete the theme acquisition.

It has important theoretical research value and application value to analyze the content of the blog, extract the subject words, and complete the analysis of the subject contained in the blog. In order to maximize the application of the subject words, we can classify the subject words, which is convenient for retrieval and use. In this experiment, we will draw on the relevant algorithms and crawler technology to build our own keyword extraction algorithm. In the second section, the third section and the fourth section of this paper, different methods of topic extraction are introduced firstly. In the fifth section, the advantages and disadvantages of these methods are compared. In the sixth section, relevant experiments are carried out to verify them. Finally, relevant summaries are made at the end of this paper.

**KEYWORDS：weibo, scrapy, information mining, data cleaning, topic acquisition**

目录

[摘要 I](#_Toc11323266)

[ABSTRACT II](#_Toc11323267)

[1 绪论 1](#_Toc11323268)

[1.1 背景 1](#_Toc11323269)

[1.2 国内外研究现状 3](#_Toc11323270)

[1.3 主题提取模型算法 4](#_Toc11323271)

[2 TF-IDF主题提取方法 6](#_Toc11323272)

[2.1 TF-IDF基础 6](#_Toc11323273)

[2.2 TF-IDF改进 7](#_Toc11323274)

[3 TextRank主题提取方法 8](#_Toc11323275)

[4 LDA主题提取方法 10](#_Toc11323276)

[5 主题词提取方法比较 12](#_Toc11323277)

[6 实验与验证 13](#_Toc11323278)

[6.1 相关技术和工具 13](#_Toc11323279)

[6.1.1 相关工具 13](#_Toc11323280)

[6.1.2 核心技术 16](#_Toc11323281)

[6.2 实验分析 20](#_Toc11323282)

[6.2.1 实验文本缺陷分析 20](#_Toc11323283)

[6.2.2 分析概述 21](#_Toc11323284)

[6.3 框架设计 21](#_Toc11323285)

[6.3.1 整体模块设计 21](#_Toc11323286)

[6.3.2 分模块设计 23](#_Toc11323287)

[6.4 具体实现 25](#_Toc11323288)

[6.4.1 文件清单 25](#_Toc11323289)

[6.4.2 具体实现 25](#_Toc11323290)

[6.5 算法结果验证 29](#_Toc11323291)

[6.5.1 测试 29](#_Toc11323292)

[6.5.2 功能与性能测试 30](#_Toc11323293)

[总结 32](#_Toc11323294)

[参考文献 33](#_Toc11323295)

[谢辞 35](#_Toc11323296)

1 绪论

信息高速发展的时代，毫无疑问，数据是最有价值的。如何有效地掌控海量信息，有效地提取并组织海量网络信息中的主题词串，成为网络信息爆炸时代亟待解决的需求。自微博开放API平台，新浪微博每天发博数巨大，之后随着微博3.0版本的推出，微博用户更是达到了5亿，而在2015年，微博官方对微博字数进行140字的限制，促使了微博短文本这一概念的产生和针对微博短文本主题研究的开始。面对如此大量的数据与信息量的传播，微博数据的挖掘变成了一个热点，对于微博短文主题的获取，此问题的解决蕴涵着广泛的应用前景：对于个人，它能更方便地发现和组织当前重要资讯，并且可以在特定或众多关注者和粉丝的大量微博中，省略一条条观看的繁琐过程，直接获取主题，达到获取他人关注点的目的，简单，快捷，而且高速，对于企业，它能及时掌握企业相关领域、战略伙伴及竞争对手的最新动态，还可以客观的统计出多用户关注的中心，为企业提供价值，帮助企业实现用户量的大量增长和持续盈利，打造出企业自身的的核心产品，对于国家，它能及时了解当前社会重要事件，流行趋向，舆论方向。

## 1.1 背景

微博即博客(MicroBlog)，是最近几年才出现的，属于一种社交平台。用户能够在任何时候通过网站、手机等方式实现状态的改变和发布信息。sina是现在使用人数比较多，国内最受欢迎的社交网站。与长文本比较，微博数的数据很不规则、文本数量巨大和时间立即性。因此怎么样从这么多、不同的个人数据中很准的获取出用户在什么时候发生了什么事情，是现在的微博主题提取研究主要的难点。本文把短文本主题提取当作背景，以微博短文本作为原始的数据，以提取主题看作是一种方式，进行了的短文本的简单处理，停用词的过滤，分词，词向量化，算法主题建模等步骤，对微博短文本进行主题的提取。

由于在微博网站上产生了大量的微博文本数据，且日益呈现大数据化趋势。Deloitte公司的一项调查表明，82％的人在购买商品或服务时会参考网上的一些微博文本等，微博文本间接影响了他们的购买决策吲。然而面对如此大数量的文本，顾客／商家，以及其他的组织直接通过这些短文本来进行某项措施是非常困难的，所以，对端文本进行主题的提取是必须的。

想从非常多的没有规律的微博短文本数据中获取想要的信息，通过人体自身进行处理根本就是不可能的，本文将介绍如何通过项目实现自动化，简便化的通过机器对短文本主题进行提取。在这样的一个背景前提下，出现了针对短文本的主题提取(Topic Extraction)技术。主题提取 (Topic Extraction)，也叫主题挖掘(Topic Mining)，主要研究和分析人们对实体对象，如商品、服务、组织、个人、问题、事件的主题，其中涉及的一些关注和看法[3,4]。主题提取是近几年来在自然语言处理和文本挖掘研究领域中兴起的研究热点，在电子商务、商业智能、信息监控等方面都有着重要的应用。对微博短文本进行主题提取，可以知道某个用户的关注点和对这个关注点的态度，包括正向、中性和负向，从而使得潜在用户无需逐条浏览和分析历史微博文本，就能做出决策，大大提高便捷性。

微博短文本通过nature语言所表示的没有规律的一些文本，文本的数量非常的多，需要整体的运用语言分析，文本分析和数据提取技术，在此同时，需要正确的减小数据文本的维度，这时，使用lda就可以减小文本数据的维度，从而达成短文本主题此提取的目的，还可以通过主题聚合来判断主体间的关系，不过获取的主题词大多是粗粒度、全局的词。细粒度词则是包含局部性，词的出限次数低且词语间相关关系不是很明显，其提取的复杂性比全局词更加的艰难。

已经存在的LDA模型偏向于全局特征词和全局主题词，在主题词语的概率分配中未曾关注词和词之间的含义关系，导致一些具有隐含语义关系的特征词和主题词提取的准确率不是很高，主要展示在以下几个方面：

(1)对频率低的词语提取很难进行。微博短文本的局部特征词相比较全局特征词和主题词，其出现的次数明显不高，包含一些次级特征词、非典型特征词等。LDA模型偏于发现出现频率较高的词语，这就使得出现较少次数的主题词获取不准确，而这些局部性词获取的准确率会对细粒度主题的获取产生很大的影响。

(2)不能发现频率低的局部特征词的互相关系。特征词和主题词之间虽然有修饰相关，但是一些低频共同出现的词，同样有特征和主题词的修饰相关。由于LDA主题模型着重于发现较高文档共同出现的词语之间的相关关系，对于出现次数较低的句子级别的共同展现的关系识别度不高

(3)全局特征词对局部特征词有一定概率影响。全局特征词比较于局部特征词有非常高的词语出现频率，容易被LDA以更大的概率分配到不同主题，这会对其他的低频词的主题分配上产生很大的影响，使这些局部词语在主题的概率分配上会比较低。

本文将从新的角度对微博文本中主题模型的作用机理进行研究，探索大数据背景下的主题模型在微博短文本分析研究中的新途径，将以国内知名网站新浪微博的文本作为数据源，通过对中文文本的语法、语义结构进行分析，构建符合中文商品评论特点的细粒度特征词提取模型，具有以下研究意义：

(1)基于LDA语义提取目标，从句法分析、词义理解和语境相关等角度进行语义获取研究，拓宽了文本语义挖掘的研究视角。

(2)从提高主题模型的语义理解能力入手，对主题模型的语义化机制进行研究，拓展了概率型主题模型进行语义提取的研究思路。

(3)对语义约束主题模型进行构建研究，丰富了复杂语境下主题提取的研究方法。

除此之外，研究成果还可以应用于其他领域海量文本的主题知识的自动化提取、分析及处理，具有广泛的应用价值和推广前景。

## 1.2 国内外研究现状

自从20世纪90年代以来，有关主题词提取问题的研究就得到了国内外普遍的关注。早先 的研究者主要是从文档、短语和词汇的角度来分析和研究主题词提取问题，研究多偏向于粗颗粒度。Kim和HovyHo[3]对意见挖掘中的“意见 (opinion)”作了定义，即观点(可以当成主题)是由4个基本元素组成的四元组。由此，意见挖掘被划分成为了4个主要子任务：(1)主题抽取 (topic extraction)：识别主题术语和领域相关的本体概念；(2)意见持有者识别(holder identification)：确定意见表述的作者和说话者；(3)陈述界定(claim delimitation)：确定意见表述的范围；(4)主题分析(topic analysis)：决定主题的语义倾向(sentiment orientation)，也就是极性 (polarity)分析。本文中的研究主要集中在第1和第4个任务，即主题(词)抽取和主题分析。

目前，与主题词抽取和主题分析相关的研究不是太多。对主题模型的研究最早来自1998年Christos H. Papadimitriou、Prabhakar Raghavan、Hisao Tamaki和Santosh Vempala提出的潜在语义索引（Latent Semantic Indexing, LSI）。1999年，Thomas Hofmann提出了概率性潜在语义索引（Probabilistic LST, PLST）。Hu和Liu[4-5]利用词共现技术，采用数据挖掘中的关联规则方法(基于Apfiofi算法)提取产品评论中的主题词。该方法首先挖掘了频繁主题词，然后又通过这些频繁词的邻近形容词发掘非频繁主题词，但该方法的正确率和召回率都不是很理想。Popeseu和Etzioni[6]提出了一种无监督的方法(oPINE)，使用了SO-MI方法，基于WordNet的IS-A 层次关系和词语形态线索来计算“主题词”与“领域”的关系，例如“Part-of”以及“property-of”。该实验结 果比Hu和Ku的方法偏好，但由于WordNet的局限性(而且并不能用于汉语领域)，不能有效识别领域相关的词汇。Yi等[7]根据名词短语的组成和位置特点，采用了相似性测试方法来确定主题。这种方法的覆盖性效果不是很理想。Wei Wei等[8]。采用了信息抽取中一种领域用语(domain terms)的提取方法，并且添加了一些主题词修剪算法作为辅助的提取方法。高磊等(2015)[10]提出基于特征选择、词频和点互信息剪枝的商品属性提取方法；文献[11-12-13]用相邻词语的共现规则来识别特征词，并提出了相应的特征筛选方法对噪声数据进行筛除。

现在随着时代的发展，主题词提取的方法已经有了很多，例如名词短语方法和共现规则方法，机器学习方法等等。由于文本数量和文本内容的不断扩大，这使得人工已经试图使用机器相关的主题模型提取方式，在完成文本降维的同时完成对主题词的提取，构成主题词之间的隐射关系。因为选择了非常多的短文本特征词语，基于算法模型的主题词提取方式逐渐的成为了热点，并且发展了起来。同时，许多研究在标准主题模型中又进行了改进和发展，以许多不同的存在方式进行了主题词语的提取，和主题词之间词意的关联，主题词的提取目前在国内外发展很流行，尤其是通过主题模型对主题词的提取。

## 1.3 主题提取模型算法

主题模型（topic model）是以非监督学习的方式对文集的隐含语义结构（latent semantic structure）进行聚类（clustering）的统计模型。主题模型主要被用于自然语言处理（Natural language processing）中的语义分析（semantic analysis）和文本挖掘（text mining）问题，例如按主题对文本进行收集、分类和降维；也被用于生物信息学（bioinfomatics）研究。隐含狄利克雷分布Latent Dirichlet Allocation, LDA）是常见的主题模型。

对主题模型的研究最早来自1998年Christos H. Papadimitriou、Prabhakar Raghavan、Hisao Tamaki和Santosh Vempala提出的潜在语义索引（Latent Semantic Indexing, LSI）。1999年，Thomas Hofmann提出了概率性潜在语义索引（Probabilistic LST, PLST）。2003年，David M.Blei、Andrew Ng和Jordan I. Michael提出了[隐含狄利克雷分布](https://baike.baidu.com/item/%E9%9A%90%E5%90%AB%E7%8B%84%E5%88%A9%E5%85%8B%E9%9B%B7%E5%88%86%E5%B8%83/15716062" \t "_blank)（Latent Dirichlet Allocation, LDA）。LDA得到了广泛使用，并衍生出了很多改进版本，例如在2006年由Wei Li和Andrew McCallum提出的弹珠机分布模型（pachinko allocation model）。

(1)特征提取主题模型

无监督特征提取模型：

Titov[5-6]LDA模型是多力度topic模型 (multi．grain LDA， MG．LDA)，促发 MG．LDA能够实现细小力度的topic建模；Das在后续中从微博短文本中获取相关的精确的主题词，也就是一些关注很高的关键词，这起着非常重要的作用。

弱监督特征提取模型：

Andrze[8]提出了整合专业相关的DF (Dirichlet forest)模型；Zh提出含有限制的LDA(Constrained．LDA)来完成文本的主题词提取，指定了mustli和cannotlink 2种限制方式；Bagheri[14]提出在迪利克雷基础上的特征展现 (Aspect Detection Model Based on LDA，ADM)，马柏樟和其他一些人通过对一些短文本进行语句分词和词性标识，得到粗的词语，之后使用LDA筛选出特征词集，最后使用相同意思词的词林拓展和一些过滤方式得到想要的特征词集；Chen和其他一些人针对无监督抽取的主题词缺点是可能结果是不一致的。Mukherjee和其他一些人在别人提供的种子的基石上，提出了自动对关键词进行获取的概率方式，从而可以实现用户的一些制定的要求；Chen等(2013)将must-set和cannot-set引入LDA，提出了MC．LDA(LDA with m-set and c-set)模型，用于提取特征词。

(2)主题词提取主题模型

无监督主题词提取：

Titov等对MG．LDA模型进行了扩展，提出了MAS(multi．aspect sentiment)模型；刘远超和王晓龙等人利用粗集理论对关键词短语的构成规则进行了挖掘,将挖掘出的规则用于指导关键词的自动抽取,避免了一些错误的搭配被抽取,从而提高了系统的性能,使抽取结果更加符合人们的习惯[3]。马亮和何婷婷等人采用查询相关性特征和话题相关性特征来对关键词语进行打分,最后将这两个特征进行线性组合来得到关键词语的重要度；孙艳等[15]考虑到有监督、半监督的评论文本数据集的标注工作量较大，且存在标注样本不容易获取的问题，提出一种无监督的主题情感混合模型 (unsupervised topic and sentiment unification，UTSU)模型，通过在标准LDA模型中 融入情感来实现文档级别的情感分类。

弱监督主题词提取：

Mukherjee等的TME(topic and multi．expression)模型对评论中共现的各类情感短语和主题建模，并利用最大熵知识改善TME中的Beta先验分布的粗糙度和微弱性，文献幂U用最大熵来影响LDA对于主题和情感的生成；一些研究利用贝叶斯模型、变分平均场推导算法、分层特征模型和图学习算法对LDA进行扩展，有效俘获文本中的特征和情感。任克强和赵光甫等人提出以带权语言网络来表征HTML标记对网页文本的影响,给出了综合介数指标与紧密度指标的词语中心度度量方法,实现了网页关键词的抽取算法,表现出良好的抽取效果以及可解释性。李丕绩等[16]对每个评价对象的所有评论文档，根据句子结构中的依存关系，构成对应实体的特征标签库，并从显式语义角度对标签库去重；Turney 与Witten 分别开发了系统GenEx 与KEA ,这两个系统在关键词抽取的发展史上具有重要的意义。他们首次利用监督学习的方法训练已标注关键词的语料,然后通过训练出的关键词抽取模型对未标注关键词的文档进行关键词抽取,此方法在准确率与召回率上都超越了前人的工作。欧阳继红[17]基于主题情感混合模型JST和R-JST(reverse joint sentiment topic model)，考虑到整体分布与局部分布的关系会影响分类效果，提出了MG-JST(multi grain JSrr)和MG．R-JST(multi grain reverse JST)模型。

所有主题模型都基于相同的基本假设：

每个文档包含多个主题，每个主题包含多个单词，换句话说，主题模型围绕着以下观点构建：实际上，文档的语义由一些我们所忽视的隐变量或「潜」变量管理。因此，主题建模的目标就是揭示这些潜在变量——也就是主题，正是它们塑造了我们文档和语料库的含义。本文将继续深入不同种类的主题模型，试图建立起读者对不同主题模型如何揭示这些潜在主题的认知，具体的主题提取方法将会在后续章节介绍。

# 2 TF-IDF主题提取方法

## 2.1 TF-IDF基础

TF-IDF（Term Frequency-Inverse Document Frequency），词频-逆文档频率算法，它是一种统计方法，用于评估一字词对一文件集或一语料库的中的某一篇文档的重要性，字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。

总结一下，就是一个词在一篇文档中出现的次数越多，同时在其他所有文档中出现的次数越少，这个词越能代表这篇文当的内容，比如‘的’这个字肯定在所有文档出现的次数都很多，不是一篇文档独有的，那这个词的重要性就很小，对应的TF-IDF值也很小，而在由体育新闻和娱乐新闻组成的文档集中，‘篮球’肯定在介绍篮球的新闻中出现的次数多，而在娱乐新闻中出现的次数少，那篮球这个词就能代表他所在文档的内容，这个词的TF-IDF就会比较大。

TF-IDF加权的各种形式常被搜索引擎应用，作为文件与用户查询之间相关程度的度量或评级。除了TF-IDF以外，因特网上的搜索引擎还会使用基于链接分析的评级方法，以确定文件在搜寻结果中出现的顺序。

TF-IDF的主要思想是：如果某个单词在一篇文章中出现的频率TF高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类

（1）TF是词频(Term Frequency)

词频指的是某一个指定词在一篇文档中出现的次数，为了避免词频偏向长文档（同一个词可能在长文档里比短文档里出现的次数多，而不管重要与否），所以用词出现的次数比文档的总词数作为归一化公式以防止它偏向长的文章，TF的计算如公式①所示。

公式①

（2）DF是文件频率(Document Frequency)

文档频率，这个参数是某个词在全局的文档中出现的篇数，如果DF越大可以认为词越重要，但是这样很多常用词也会变成重要词，所以进行IDF计算

（3）IDF是逆向文件频率(Inverse Document Frequency)

有些通用词在每个文档中都会大量出现，比如‘的’这样的词，用TF公式计算出来的权重肯定很大，但是这样的词无法反应一篇文档的主题，我们需要那些在一篇文档中出现的多而在其他文档中出现的少的词，这一类的词才能反映文档主题，显然TF是做不到这一点的，而逆向文件频率恰好可以做到这一点，如果包含词条t的文档越少, IDF越大，则说明词条具有很好的类别区分能力，IDF的计算如公式②所示。

公式②

其中，**|D|是语料库中的文件总数。分母表示包含词语ti的文件数目**（即ni，j≠0的文件数目）。如果该词语不在语料库中，就会导致分母为零，因此**一般情况下使用1+|{j:ti∈dj}|，IDF的计算如公式****③所示。**

**公式③**

（4）TF-IDF实际上是：TF \* IDF

某一特定文件内的高词语频率，以及该词语在整个文件集合中的低文件频率，可以产生出高权重的TF-IDF。因此，TF-IDF倾向于过滤掉常见的词语，保留重要的词语，计算如公式④所示。

公式④

**注：**TF-IDF算法非常容易理解，并且很容易实现，但是其简单结构并没有考虑词语的语义信息，无法处理一词多义与一义多词的情况。

## 2.2 TF-IDF改进

TF-IDF的改进：

(1)TF部分的改进

这里考虑将文档内的词频率更改为同一类文档内的词频率可以在一定程度上解决上面提到的第2项不足之处。

(2)IDF部分的改进

传统的IDF通常可以写作：IDF=log(总文档数N/所有含特征词文档数n+0.01)

在我查阅的所有论文中都提到了上面的第3项不足，这是TF-IDF应用于分类问题上的一个很明显的不足，针对这个不足，这些论文中也提到了不同的解决方法：

(1)IDF=log(本类含特征词文档数m\*总文档数N/所有含特征词文档数n+0.01)

(2)用真数的分子表示特征词mk在当前类别中的频率，用真数的分母表示特征词Mk在其他类别中的频率，对IDF计算如公式⑤所示。

公式⑤

在真数中，当分子很大，IDF的绝对值反而小，则对他取反，根据log函数的特性，自变量要大于0，IDF要为正值，最后修正IDF计算如公式⑥所示：

公式⑥

进一步计算可得值的计算如公式⑦所示：

公式⑦

对改进算法进行举例说明：

有C1，C2两个类，C1类中有14篇文档，C2类中有6篇文档，共计20篇文档，m1，m2为两个特征值词。在C1，C2中包含m1，m2特征词的文档数分别为9篇、5篇、1篇、5篇。根据传统的IDF计算方法，m1，m2在类C1，C2中IDF值分别为IDF1，IDF2，则IDF1=IDF2=log(20/10+0.01)，两者之间无法判断哪个词更易于区分类别，而根据实际观察m1分布不均匀而m2分布很均匀，表明m1比m2具有更好的类别区分能力。

根据改进方法：

(1)IDF1=log(9\*20/10+0.01)=1.2555，IDF2=log(5\*20/10+0.01)=1.0004

根据改进方法：

(2)IDF2=log(1+(9/14)/(1/6))=0.552，IDF2=log(1+(5/14)/(5/6))=0.155

不难看出用传统IDF无法判断而实际又有很好区分能力的特征词在使用2种改进方法后均可以找出区分能力更强的特征词，但这两种改进方法到底哪一个比较好我暂时还没得出明确结果，还需要用实际的训练测试集进行分类测试再来判定。

总的来说，利用tfidf算法进行主题词提取，实际上是在使用过程中，先使用准本好的数据计算出每个词的IDF值，作为一个原始信息，在对新内容进行处理时，只需要计算出TF值就可以了，然后对这篇内容的所有词计算出TFIDF值，然后进行排序，得出词频最高的指定个数主题。

# 3 TextRank主题提取方法

TextRank是在Google的PageRank算法启发下，针对文本里的句子设计的权重算法，目标是自动摘要。它利用投票的原理，让每一个单词给它的邻居（术语称窗口）投赞成票，票的权重取决于自己的票数。这是一个“先有鸡还是先有蛋”的悖论，PageRank采用矩阵迭代收敛的方式解决了这个悖论，PageRank计算如公式①所示。

公式①

其中S(Vi)是网页i的中重要性（PR值）。d是阻尼系数，一般设置为0.85。In(Vi)是存在指向网页i的链接的网页集合。Out(Vj)是网页j中的链接存在的链接指向的网页的集合。|Out(Vj)|是集合中元素的个数。

PageRank需要使用上面的公式多次迭代才能得到结果。初始时，可以设置每个网页的重要性为1。上面公式等号左边计算的结果是迭代后网页i的PR值，等号右边用到的PR值全是迭代前的，基于PageRank的TextRank关键词提取方法具体如下。

TextRank 的原理和概念：

TextRank 算法是一种用于文本的基于图的排序算法。其基本思想来源于谷歌的 PageRank算法（其原理在本文在下面）, 通过把文本分割成若干组成单元(单词、句子)并建立图模型, 利用投票机制对文本中的重要成分进行排序, 仅利用单篇文档本身的信息即可实现关键词提取、文摘。和 LDA、HMM 等模型不同, TextRank不需要事先对多篇文档进行学习训练, 因其简洁有效而得到广泛应用。

TextRank 一般模型可以表示为一个有向有权图 G =(V, E), 由点集合 V和边集合E组成, E 是V ×V的子集。图中任两点 Vi , Vj 之间边的权重为 wji , 对于一个给定的点 Vi, In(Vi) 为 指 向 该 点 的 点 集 合 , Out(Vi) 为点 Vi 指向的点集合，具体计算如公式②所示。

公式②

关键词抽取的任务就是从一段给定的文本中自动抽取出若干有意义的词语或词组。TextRank算法是利用局部词汇之间关系（共现窗口）对后续关键词进行排序，直接从文本本身抽取。

基于TextRank的关键词提取的步骤如下：

(1)文章分词： 对每一篇文章进行分词，分词系统主要由坤雁分词系统、ansj分词，结巴分词等。

(2)分词结果数据清洗：主要包括去停用词、去除符号字母数字等。

(3)构建候选关键词图：根据设定的词语选择窗口截取文本的分词结果，将每个词语作为候选关键词图的节点，截取的每一段文本中的词语作为相邻的边，以此构建候选关键词图。

(4)关键词提取： 利用pagerank思想循环迭代候选关键词图， 每个节点的权重初始化化为1.0f，通过设定的迭代次数达到稳定后，对节点权重进行倒序排序，从而得到最重要的num个单词，作为候选关键词。

其细节的步骤如下：

(1)把给定的文本T按照完整句子进行分割，即 下载 (1)

(2)对于每个句子，进行分词和词性标注处理，并过滤掉停用词，只保留指定词性的单词，如名词、动词、形容词，即下载 (2)，其中下载 (3)是保留后的候选关键词。

(3)构建候选关键词图G = (V,E)，其中V为节点集，由(2)生成的候选关键词组成，然后采用共现关系（co-occurrence）构造任两点之间的边，两个节点之间存在边仅当它们对应的词汇在长度为K的窗口中共现，K表示窗口大小，即最多共现K个单词。

(4)根据上面公式，迭代传播各节点的权重，直至收敛。

(5)对节点权重进行倒序排序，从而得到最重要的T个单词，作为候选关键词。

(6)由(5)得到最重要的T个单词，在原始文本中进行标记，若形成相邻词组，则组合成多词关键词。例如，文本中有句子“Matlab code for plotting ambiguity function”，如果“Matlab”和“code”均属于候选关键词，则组合成“Matlab code”加入关键词序列。

# 4 LDA主题提取方法

LDA是一种非监督机器学习技术，可以用来识别大规模文档集（document collection）或语料库（corpus）中潜藏的主题信息。它采用了词袋（bag of words）的方法，这种方法将每一篇文档视为一个词频向量，从而将文本信息转化为了易于建模的数字信息。但是词袋方法没有考虑词与词之间的顺序，这简化了问题的复杂性，同时也为模型的改进提供了契机。每一篇文档代表了一些主题所构成的一个概率分布，而每一个主题又代表了很多单词所构成的一个概率分布。由于 Dirichlet分布随机向量各分量间的弱相关性（之所以还有点“相关”，是因为各分量之和必须为1），使得我们假想的潜在主题之间也几乎是不相关的，这与很多实际问题并不相符，从而造成了LDA的又一个遗留问题。

LDA模型较为复杂，涉及了大量的数学知识，例如gamma函数，二项分布、多项分布、beta分布、Dirichlet分布，共轭先验和贝叶斯框架等等，这里制作简单的介绍。

通俗理解如下：

(1)LDA是一种无监督的贝叶斯模型。

(2)LDA是一种主题模型，它可以将文档集中的每篇文档按照概率分布的形式给出。

(3)LDA是一种无监督学习，在训练时不需要手工标注的训练集，需要的是文档集和指定主题的个数。

(4)LDA是一种典型的词袋模型，它认为一篇文档是由一组词组成的集合，词与词之间没有顺序和先后关系。

它主要的优点就是可以对每个主题，都找出一些词来描述它。

LDA的分类原理：

原理为： (词│文档)的概率等于(词│主题)和(主题│文档)的概率乘积。

也就是词→主题→文档这样的关系，同一主题下，某个词出现的概率，以及同一文档下，某个主题出现的概率，两个概率的乘积，可以得到某篇文档出现某个词的概率，我们在训练的时候，调整这两个分布就可以了，两个分布如图4-2所示，其中两个分布的横轴分别为主题和词。

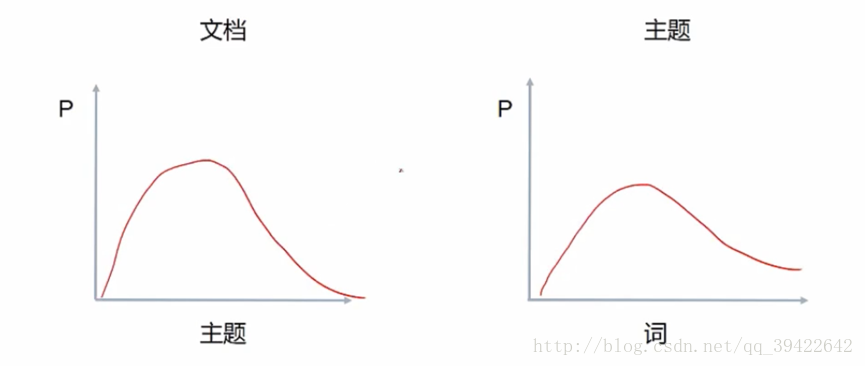


图4-2 两种概率分布

由此可以定义LDA的生成过程：

(1)对每篇文档，在主题分布中抽取一个主题（相当于主题文档分布）

(2)对抽到的主题所对应的单词分布中随机抽取一个单词（在主题词分布中抽取）

(3)重复上述过程直至遍历整篇文档中的每个单词

经过以上三步，就可以看一下两个分布的乘积，是否符合给定文章的分布，以此来调整。

稍微具体点讲的话如下：(w代表单词;d代表文档;t代表主题; 大写代表总集合，小写代表个体。)

D中每篇文档d看作个单词序列为<w1,w2,…,wn>，其中wi代表第i个单词。

D中涉及的所有不同单词组成一个词汇表大集合V (vocabulary)，LDA以文档集合D作为输入，希望训练出的两个结果向量 (假设形成k个topic，V中共有m个词):

结果向量1:对每个D中的文档d，对应到不同主题的概率θd为<pt1,…,ptk>，其中pti表示d对应k个主题中第i个主题的概率，计算如公式①所示。

d 中多少个词是第i个主题也有的 中所有词的总数

公式①

结果向量2:对每个T中的主题t，生成不同单词的概率向量ϕt为<pw1,…,pwm>,其中pwi表示主题t生成V中第i个单词的概率，计算如公式②所示。

主题t对应到V中第i个单词出现的次数

主题t下的所有单词总数

公式②

说了这么多，总的来说，利用LDA来提进行主题提取的核心计算如公式③所示：

公式③

用表达式为：

学习过程可以表示为：

LDA算法开始时，先随机地给θd，ϕt赋值(对所有的d和t)

针对特定的文档ds中的第i单词wi，如果令该单词对应的主题为tj，可以把上述的计算改为如公式④所示:

公式④

枚举T中的主题，得到所有的pj(wi|ds).然后可以根据这些概率值的结果为ds中的第i个单词wi选择一个主题，最简单的就是取令Pj(wi|ds)概率最大的主题tj。

如果ds中的第i个单词wi在这里选择了一个与原先不同的主题，就会对θd，ϕt有影响，他们的影响反过来影响对上面提到的p(w|d)p(w|d)的计算。

对文档集D中的所有文档d中的所有w进行一次p(w|d)计算，并重新选择主题看成是一次迭代。迭代n次之后就可收敛到LDA所需要的分类结果了。

利用LDA进行主题提取的实现步骤和其他方法大的方面都是一样的，分词，停用词和其他无用的词过滤，建模，主题提取，不同的是，需要计算和比较的方面不同，本文使用的就是LDA进行微博短文本的主题词提取，后续章节将主要介绍微博主题词提取的详情。

# 5 主题词提取方法比较

前文详细介绍了三种主题提取的方式，他们各自不同，对不同的文本，采取不同的方式，会有不同的结果，精确度也会有所不同和，当然，方法的不同，用于同一文本的主题词提取上也会产生不同的结果，这是不可避免的，也是正常的。

通过前文详细的介绍主题提取的方法，不难看出，词频法，TextRank，LDA三种方法的优缺点：

对于TF-IDF而言，可以计算出在一文档集合中特定文档里所包含的区别于其他文档的重要词语，换言之就是关键词。而在文本分类问题中，仅仅是每篇文档区分度强的关键词还不足以作为分类的评判标准，即传统的TF-IDF还存在许多不足，在查阅了很多相关论文后我列举了以下一些传统TF-IDF的不足之处如下：

(1)没有考虑特征词的位置因素对文本的区分度，词条出现在文档的不同位置时，对区分度的贡献大小是不一样的。

(2)按照传统TF-IDF函数标准，往往一些生僻词的IDF(反文档频率)会比较高、因此这些生僻词常会被误认为是文档关键词。(换句话说，如果一个特征项只在某一个类别中的个别文本中大量出现，在类内的其他大部分文本中出现的很少，那么不排除这些个别文本是这个类中的特例情况，因此这样的特征项不具有代表性。)

(3)传统TF-IDF函数中的IDF部分只考虑了特征词与它出现的文本数之间的关系，而忽略了特征项在一个类别中不同的类别间的分布情况。

(4)对于文档中出现次数较少的重要人名、地名信息提取效果不佳。

除了这些不足之处，TF-IDF的主要优点就是该方法提取主题词非常简洁，只需要根据关键词在文档中出现的频率就可以进行关键词的提取，易于理解。

对于TextRank而言，它本质上属于无向带权图，具有以下优点。

(1)将有向图算法改进为无向图算法，不再区分结点的指向性，而只关注结点的邻居结点及本身的连通度。

(2)具有较高权重的结点也被认为与更多重要结点具有较高的连通度，可以抽象为网络的“中心”，而在多文档文摘中，中心结点代表包含重要信息的文摘句。

不足之处在于，基于图形，并且理解起来比较复杂，而且在众多的主题提取方法中，它并不出众。

最后对于LDA而言，LDA比较复杂，其内部涉及了大量的数学知识，但是除去其复杂的内部原理，目前使用起来还是很简单的，因为高度封装的模块可以帮助使用，具体优缺点如下：

LDA算法的主要优点有：

(1)在降维过程中可以使用类别的先验知识经验，而像PCA这样的无监督学习则无法使用类别先验知识。

(2)LDA在样本分类信息依赖均值而不是方差的时候，比PCA之类的算法较优。

LDA算法的主要缺点有：

(1)LDA不适合对非高斯分布样本进行降维，PCA也有这个问题。

(2)LDA降维最多降到类别数k-1的维数，如果我们降维的维度大于k-1，则不能使用LDA。当然目前有一些LDA的进化版算法可以绕过这个问题。

(3)LDA在样本分类信息依赖方差而不是均值的时候，降维效果不好。

(4)LDA可能过度拟合数据。

总的来说，三中方法中词频法最简单，三种方法的精确度都不是很高，所以在使用中，想要达到高的精确度，最好的方法就是结合三种方法的优缺点，选择其中的两种或多种组合使用，从而达到更好的效果，本文主要使用lda进行主题提取，后续将会主要对lda提取主题方法进行实验验证。

# 6 实验与验证

## 6.1 相关技术和工具

### 6.1.1 相关工具

(1)mongodb数据库

作为定义，MongoDB是一个开源数据库，它使用面向文档的数据模型和非结构化查询语言。它是当今最强大的NoSQL系统和数据库之一,作为NoSQL工具意味着它不使用我们与关系数据库管理相关联的常用行和列。它是一个基于集合和文档的架构。此数据库中的基本数据单元由一组键值对组成。它允许文档具有不同的字段和结构。此数据库使用名为BSON的文档存储格式，它是JSON样式文档的二进制样式。MongoDB遵循的数据模型是一种高度灵活的数据模型，它允许您组合和存储多变量类型的数据，而不必在强大的索引选项，数据访问和验证规则上妥协。想要动态修改模式时没有停机时间。那么这意味着您可以更专注于使数据更加努力，而不是花更多的时间来为数据库准备数据。

MongoDb具有出色的可扩展性。它可以轻松获取数据并提供连续和自动的集成活性使MongoDB如此强大。

让我们来介绍下Mogodb的详细架构图，具体如图6-1所示。

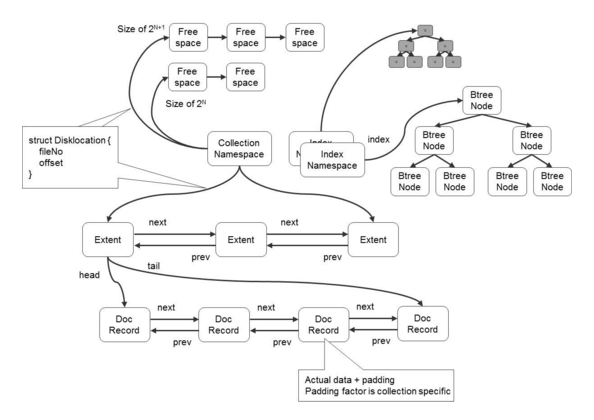


图6-1 mongodb架构图

MongoDB主要有以下功能。

(1)每个数据库都包含集合，而集合又包含文档。每个文档可以具有不同数量的字段。每个文档的大小和内容可以彼此不同。

(2)文档结构更符合开发人员在各自的编程语言中构建类和对象的方式。开发人员经常会说他们的类不是行和列，而是具有键值对的清晰结构。

(3)如NoSQL数据库的介绍中所示，行（或MongoDB中调用的文档）不需要预先定义模式。相反，可以动态创建字段。

(4)MongoDB中可用的数据模型允许您更轻松地表示层次关系，存储数组和其他更复杂的结构。

以下是为什么应该开始使用MongoDB的几个原因。

面向文档 - 由于MongoDB是NoSQL类型的数据库，它不是以关系类型格式存储数据，而是将数据存储在文档中。这使得MongoDB非常灵活，能够适应真实的商业环境和要求。即席查询 - MongoDB支持按字段，范围查询和正则表达式搜索进行搜索。可以进行查询以返回文档中的特定字段。索引 - 可以创建索引以提高MongoDB中的搜索性能。可以索引MongoDB文档中的任何字段。复制 - MongoDB可以为副本集提供高可用性。副本集由两个或多个mongo数据库实例组成。每个副本集成员可以随时以主副本副本的角色运行。主副本是主服务器，它与客户端交互并执行所有读/写操作。辅助副本使用内置复制维护主数据的副本。当主副本失败时，副本集会自动切换到辅助副本，然后它将成为主服务器。负载平衡 - MongoDB使用分片的概念通过在多个MongoDB实例之间拆分数据来水平扩展。MongoDB可以运行在多个服务器上，平衡负载和/或复制数据，都可以便于在硬件发生故障时保持系统正常的运行。

Mogodb通过类比得出的的重要概念如表6-1所示。

表6-1 mongdb数据库重要概念表

|  |  |  |
| --- | --- | --- |
| Sql术语 | Mongodb术语 | 解释说明 |
| Database | Database | 数据库 |
| Table | Collection | 数据库表/集合 |
| Row | Document | 数据记录行/文档 |
| Column | Field | 数据字段/域 |
| Index | Index | 索引 |
| Table jions |  | 表连接，mongodb不支持 |
| Primary key | Primarykey | 主键，默认是id字段 |

资料来源：网络.sql文档和mongodb文档对比分析，相关官方文档.2019：5-19.

(2)jupyter notebook

本系统主要采用jupyter notebook用生成画图的方式演示结果，下面对改工具进行简单介绍。

Jupyter Notebook是基于网页的用于交互计算的应用程序。简而言之，Jupyter Notebook是以网页的形式打开，可以在网页页面中直接编写代码和运行代码，代码的运行结果也会直接在代码块下显示。Jupyter Notebook是一个基于Web的应用程序，适用于解决计算问题并将结果与代码，解释性文本和背景一起呈现的过程。

为什么要使用jupyter notobook工具，因为使用了该工具，可以代替简单的web前端界面进行鉴定那图像的展示，因为他具用一大特性，那就是对象的富媒体表示，例如使用HTML和CSS进行图像自定义。

使用方式。最快的方法是使用终端/ CMD。从命令行执行命令jupyter notebook将启动笔记本服务器并打开Web浏览器到Web应用程序仪表板。该命令还将在控制台中打印有关笔记本服务器的一些信息。第一次连接时，您可能需要将控制台中显示的URL复制并粘贴到浏览器中以打开应用程序仪表板，运行后出现如图6-2所示则为成功(内容不会相同但界面相同)。

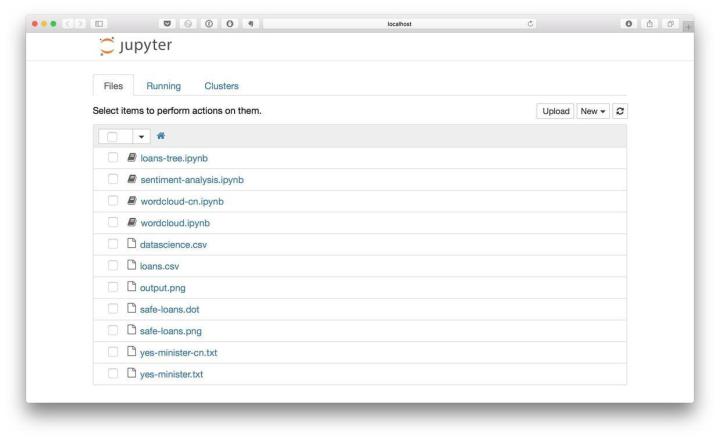


图6-2 jupyter初始界面图

### 6.1.2 核心技术

(1)scrapy框架

Scrapy一个开源和协作的框架，其最初是为了页面抓取 (更确切来说, 网络抓取 )所设计的，使用它可以以快速、简单、可扩展的方式从网站中提取所需的数据。但目前Scrapy的用途十分广泛，可用于如数据挖掘、监测和自动化测试等领域，也可以应用在获取API所返回的数据(例如 Amazon Associates Web Services ) 或者通用的网络爬虫，其主要部分及流程如下：

引擎(EGINE)：引擎负责控制系统所有组件之间的数据流，并在某些动作发生时触发事件。有关详细信息，请参见上面的数据流部分。

调度器(SCHEDULER)：用来接受引擎发过来的请求, 压入队列中, 并在引擎再次请求的时候返回. 可以想像成一个URL的优先级队列, 由它来决定下一个要抓取的网址是什么, 同时去除重复的网址

下载器(DOWLOADER)：用于下载网页内容, 并将网页内容返回给EGINE，下载器是建立在twisted这个高效的异步模型上的

爬虫(SPIDERS)：SPIDERS是开发人员自定义的类，用来解析responses，并且提取items，或者发送新的请求

项目管道(ITEM PIPLINES)：在items被提取后负责处理它们，主要包括清理、验证、持久化（比如存到数据库）等操作

下载器中间件(Downloader Middlewares)：位于Scrapy引擎和下载器之间，主要用来处理从EGINE传到DOWLOADER的请求request，已经从DOWNLOADER传到EGINE的响应response，你可用该中间件做以下几件事爬虫中间件(Spider Middlewares)：位于EGINE和SPIDERS之间，主要工作是处理SPIDERS的输入（即responses）和输出（即requests）

(2)主题模型算法

线性判别分析是一种监督算法，因为它考虑了类标签。这是一种减少“维度”的方法，同时尽可能多地保留类别歧视信息。

LDA可帮助您找到类集群周围的边界。它将您的数据点投影到一条线上，以便您的簇尽可能分开，每个簇与质心具有相对（接近）的距离。所以问题就出现了 - 如何定义这些集群，如何在LDA的情况下获得简化的特征集？基本上，LDA会找到每个类数据点的质心。例如，利用十三个不同的特征，LDA将使用十三个不同的特征数据集找到其每个类的质心。现在在此基础上，它确定了一个新的维度，它只是一个应满足两个标准的轴：

1. 最大化每个类的质心之间的距离。
2. 在每个类别中最小化变化（LDA调用散射并由s2表示）。

因此要点是：

注意：这里的“卑鄙”只不过是班级的质心。变化只不过是飞机上数据的传播。因此，如果数据的变化最小，那么类之间的重叠将更少，并且将在不同类之间保持最大分离。

因此，无论新轴的哪个坐标满足这两个标准，它们都构成了数据集的新维度。

恭喜，您现在是LDA内部工作的主人。现在让我们在wine数据集上使用sklearn跳转到LDA的实现，看看数据如何从更高的维度转换到更低的维度。

让我们在wine数据集上执行LDA并分析图表：  
 (1)将wine数据集加载到内存并执行特征提取。  
 (2)缩放数据集-可以使用min-max标准化来缩放数据集，其平均值为零和单位标准差。

(3)在sklearn中应用具有内置LDA功能的LDA一个。来自sklearn.decomposition导入LDA。

总之，如果每个类的样本数量较少，PCA的表现会更好。而LDA对于具有多个类的大型数据集更好;类可分性是降低维数的重要因素。

(3)ip代理池

Ip代理池是一种接口，主要用于当对微博在短时间内请求数据过多时，避免微博认为这不是用户而是爬虫，从而不给返回数据，防反爬的一种手段，本项目用到的ip代理池是proxy\_pool,而且主要对如图6-3所示。

由于api是免费使用的，所以，这些网站所提供的ip中有很多都是无用的，大量的ip经测试只有少量的能用，而且用这些ip进行数据爬取速度非常慢，远远比不上不用ip代理池的情况，不过为了避免微博反扒数据，我们依旧采用了ip代理池，虽然效率有所下降，不过安全性有所提升。



图6-3爬取这些网站提供的ip

本项目使用的Ip代理池proxy\_pool，是一款开源免费，高效，扩展性高的ip代理池，它含有丰富的数据库支持，使用时只需要修改相关配置，类似于支付宝接口一样，很方便，但免费软件修很多弊端，上面已经有所提及，这里就不说了，它的本质也是一款封装的爬虫，使用时需要先运行它，他会从相关代理网站爬取ip数据，并且对这些ip进行判断是否可用，如果可用，它就会将相应ip存储到数据库中，在进行使用时，项目会根据特定url对i爬取过的ip进行获取，每时每刻ip都是不同的，所以，用ip代理池进行微博数据爬取，不失是一种反扒措施。

注意，使用IP代理池时，所爬取的ip数据，会默认的存放在redis数据库中，并且值得注意的是，使用该ip代理池时，应注意通过以下方法进行操作，如图6-4所示。



图6-4 ip代理池的操作简介

(4)语料库和分词工具

语料库(corpora)是一大堆文本。它是一种书面或口头材料，语言分析基于此。复数形式的语料库是语料库。一些流行的语料库是英国国家语料库（BNC），COBUILD /伯明翰语料库，IBM / Lancaster英语口语语料库。单语语料库仅代表一种语言，而双语语料库代表两种语言。欧洲语料库倡议（ECI）语料库是多种语言，有土耳其语，日语，俄语，中文和其他语言的9800万字。语料库可以由书面语言，口语或两者组成。口语语料库通常采用录音形式。语料库可以是开放的或封闭的。一个开放的语料库是一种不要求从而特定的区域包含的所有数据封闭语料库声称包含来自特定字段的所有或几乎所有数据。例如，历史语料库是关闭的，因为没有对区域的进一步输入。

那语料库又有什么用呢？语料库为语法学家，词典编纂者和其他感兴趣的人提供更好的语言描述。计算机可处理的语料库允许语言学家采用完全问责制原则，检索特定单词或结构的所有出现以供检查或随机选择的样本。语料库分析提供词汇信息，形态句法信息，语义信息和语用信息。

Corpora用于开发[NLP](http://language.worldofcomputing.net/nlp-overview/natural-language-processing-overview.html)工具。应用包括拼写检查，语法检查，语音识别，文本到语音和语音到文本合成，自动抽象和索引，信息检索和[机器翻译](http://language.worldofcomputing.net/machine-translation/machine-translation-overview.html)。Corpora还用于为学习者创建新的词典和语法。比较著名的英文语料库是英国皇家语料库，[英国国家语料库](http://www.natcorp.ox.ac.uk/corpus/index.xml)（BNC）包含一个代表当代英国英语世界的样本集。BNC是一种平衡的语料库，因为它试图捕获各种语言使用。它也是一个包含书面和口头语言的混合语料库。口头文本是叙事演说的转录。据估计，BNC语料库有1亿字。百分之九十的BNC由书面文本组成，而中文语料库则是由多种组成，例如，大学的语料库和百度语料库等等。

分词系统主要是对语料库进行分词，将语料库按照一定要求，分割开来，形成一个个单独的词汇，以便于其他方面的数据采集和分析等应用。

本项目用到的分词系统是jieba分词工具，在分词的过程中，自定义添加了一些附加条件，让分词变得更加精确一下，然后用了众多的停用词对分词进行了过滤，最终分词完成。

## 6.2 实验分析

### 6.2.1 实验文本缺陷分析

此处以微博数据加以说明，进行微博短文本的特点分析，如表6-2所示。

表6-2 微博数据抽取表

|  |  |
| --- | --- |
| Id | 短文本数据 |
| 1 | "体育课笑喷，情景再现<span class=""url-icon""><img alt=[笑cry] src=""//h5.sinaimg.cn/m/emoticon/icon/default/d\_xiaoku-d320324f00.png"" style=""width:1em; height:1em;"" /></span>//<a href='/n/城南邮局'>@城南邮局</a>:上体育课的女生哈哈哈哈哈真实//<a href='/n/然后下面就没了'>@然后下面就没了</a>: 哈哈哈哈哈//<a href='/n/段子楼'>@段子楼</a>:上体育课的女生" |
| 2 | *"*《愿无岁月可回头》正式预售，希望这本书，你会喜欢，希望你读到时，不会失望。转发里抽4位朋友，每人5000，共20000元， |
| 3 | *"*"url-icon""><img alt=[拜拜] src=""//h5.sinaimg.cn/m/emoticon/icon/default/d\_baibai-71b47dffdc.png"" style=""width:1em; height:1em;"" /></span>//<a href='/n/休闲璐'>@休闲璐</a>:这个超好看，马一下晚上再看一次//<a href='/n/张奶霞'>@张奶霞</a>:我闻讯赶来//<a href='/n/夏目家的小诗哥'>@夏目家的小诗哥</a>:我的躺床读物来了 以前天涯真的精彩贴子太多//<a href='/n/过顺发欢欢喜喜'>@过顺发欢欢喜喜</a>:马克//<a href='/n/老鸡灯儿'>@老鸡灯儿</a>:不上班了我细细品味*"* |
| 4 | 嗯…… 是我 |

资料来源：微博数据.微博博主随即数据获取.中国新浪微博，2019：05-18.

从表1-1的的四条数据可以看出微博短文本主题的相关特征：

短文本的长度为140字左右，其中的消息很杂乱，可以包含表情，链接地址，图片，各种各样的符号和数字等等，具体可以分为以下几个方面：

(1)数据样本的数据稀疏性。短文本的字数比较少，这就不可避免地造成了数据的稀疏，对仅有几个或几十个字符的文本进行主题提取，这必定会造成严重的数据稀疏问题。

(2)数据样本的不规范性。可以看出数据中含有大量的链接，符号，而且更甚者存在一些非官方符号表情，除此之外，观看多个用户微博可以发现用户大都很喜欢发送表情，符号等等，大量如此的操作，势必存在及其不规范性的问题。

(3)数据样本的实时性。微博数据用户发微博更新很快，这就造成了数据转变很快，数量过多，会导致样本分布不均匀，产生大量的噪音，这就对程序有很大的性能要求。

(4)数据样本的语义多样性。数据太多，导致一词多义现象严重，这对主题提取有严重的影响。造成主题提取有巨大偏差。

### 6.2.2 分析概述

基于技术可行性，操作可行性衡量拟议项目解决问题的程度，并利用范围定义中确定的机会以及它如何满足系统开发的需求分析阶段确定的要求。运营可行性审查了组织支持拟议系统的意愿。这可能是最难以衡量的可行性。为了确定这种可行性，了解管理层对拟议项目的承诺非常重要。如果请求是由管理层发起的，则可能存在管理支持，系统将被接受和使用。但是，员工群体接受变更也很重要。

总而言之，可行性研究用于确定一个想法的可行性，例如确保项目在法律和技术上可行以及经济上合理。它告诉我们一个项目是否值得投资 - 在某些情况下，一个项目可能不可行。这可能有很多原因，包括需要太多资源，这不仅会阻止这些资源执行其他任务，而且可能会花费超过组织通过承担无利可图的项目而获得的成本，可行性的考虑是非常重要的一个方面。

另外最重要的就是经济可行性了，也就是项目成本，项目成本是采购所有所需产品，服务和资源以成功交付项目所需的成本。示例：在建筑项目的示例中，成本估算从土地购置成本，建筑成本，材料成本，管理成本，人工成本和其他直接和间接成本开始。

现在您已了解项目成本。接下来，让我们看看为什么它很重要。

为什么这很重要，没有预算就没有项目开始。项目成功取决于项目成本在项目中的处理情况。很多时候，项目可能无法在项目成本内完成。这意味着当比较项目成本与项目利润时，项目成本可能已经超出，当然它被视为项目失败。因此，提出项目所需的正确成本估算非常重要。为了得出准确的成本估算，需要了解项目中涉及的项目成本类型。

总而言之，基于各项分析，以及实验样本的缺陷克服，实验是值得验证的。

## 6.3 框架设计

根据功能需求，对实验项目个部分进行分块，整合，按照结构化程序设计的要求，该项目分为整体模块，前台展示模块，后端数据采集模块，后端数据提取模块等。

### 6.3.1 整体模块设计

传统的网页分析应该这样进行，首先对网站的数据综合分析，大体的了解一下网站数据信息。打开一个网站数据源代码查看，分析一下里keywords关键词，分析网站的源代码 h1标签、h2标签、strong标签网站的TDK、ALT属性等。分析网站内链结构，总而言之就是利用页面标签对网站的数据进行爬取，但是这种方法本身不仅要求自身具备很扎实的基础，还要对前端页面的代码进行解析，无疑这样是非常麻烦的，与之相比，本项目不需要进行繁琐的源代码分析，我们主要是根据url返回的数据进行的，而不是从标签中获取，具体如下面所叙述。

网页分析是非常重要的，从网页分析的过程中，我们可以学习到一些技巧，同时也可以加深我们对整个项目框架的深入了解，有些网站的网页都是ajax动态刷新，尽管页面内容发生了改变，但url不变，这样的网页url很难获得，而且如果不对网页分析进行设计，那么整个程序都将无法进行，页面分析设计主要是对页面进行分析，对于新浪微博的url是会变化的，比较好分析，然后根据页面url的规则，进行自定义url的设计，进行整个项目的开始和整体设计。

进入微博网站m.weibo.cn并且打开浏览器的开发者工具点击xhr，分别进入用户微博界面，用户关注列表，用户粉丝列表，注意在进入相关页面后需要点解左边的清除按钮将开发者工具中的信息进行清除，然后对preserve进行打勾，之后通过不断下拉相关页面，(微博采用下滑方式代替翻页),观察xhr会发现随着不断翻页，其中会郑家内容，增加的内容就是所需要的数据url，双击该url，浏览器中就会出现新浪微博后台所返回的json数据。

点击xhr后，浏览器会打开新窗口，这些就是新浪微博通过api的url返回的后台数据，它类似于字典，我们不能直接使用，哪怕被转化后也是一样的，我们必须对其进行分析，从json数据中抽取所需要的部分，这些提取的数据就是微博文本了，当然分析数据的过程有些麻烦，因为返回的数据很长。

下面就是原始的json数据了，可以发现服务器后台返回的数据很乱，原因是没有格式化，我们使用是需要对其进行代码转化，具体获得的json原始数据如图6-5所示。



图6-5 json事例数据

原始数据转化后的真实数据如图6-6所示，注意必须要用代码进行转化。



图6-6 反序列化后的json数据

基于页面分析的整体模块设计如图6-7所示

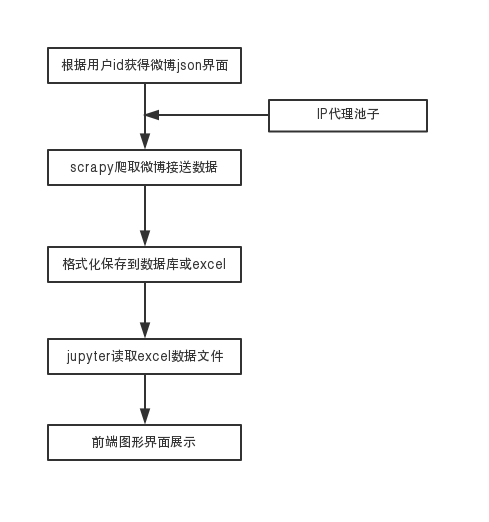


图6-7整体设计模块图

### 6.3.2 分模块设计

(1)scrapy爬取多用户微博数据

模块设计，如图6-8所示。

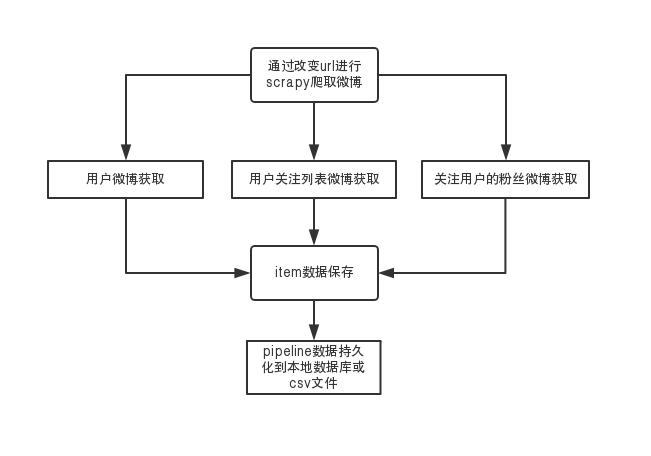


图6-8 scrapy爬取微博数据整体模块流程

Scrapy爬取数据保存的csv文件内容格式为一行一条数据，如图6-9所示。

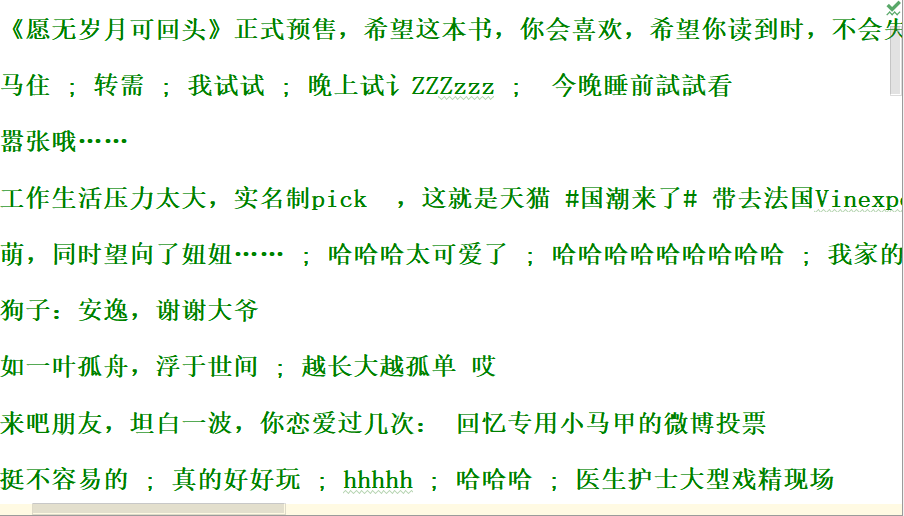
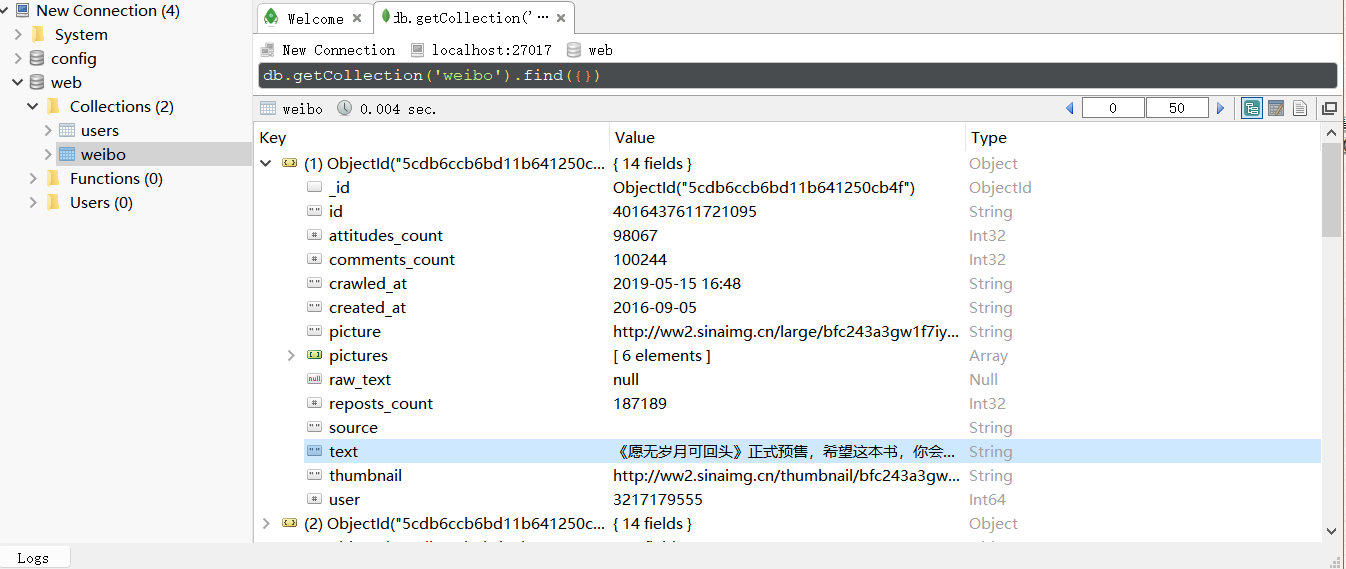


图6-9csv保存数据格式

(2)数据库表设计

数据库表设计可以自动生成无需用户创建，主表是微博短文本内容表，因为是对微博短文本进行提取，所以，唯一表只用来存放ip的redis数据库表，如图6-10所示。

图6-10 数据库表设计图

## 6.4 具体实现

### 6.4.1 文件清单

本项目实现起来较为繁琐，功能比较单一，主要包含获取数据的模块和数据分析的模块两部分，项目主要清单如下：

Weibocn.py：对微博进行爬去过滤

Items.py：对数据进行格式化

Pipelines.py：将数据进行持久化到本地

Middlewares.py：自定义的中间件

Settings.py：数据获取配置

Weibo.csv：用来存储处理过的微博文本数据

Config/settings.py：ip配置

Run/main.py：运行ip代理池

topic-get.ipynb：主题提取并生成图表

### 6.4.2 具体实现

(1)url生成与分发

第一步对概要设计中的url进行设计，并分发到爬取函数进行数据爬取，具体流程如图6-11所示。

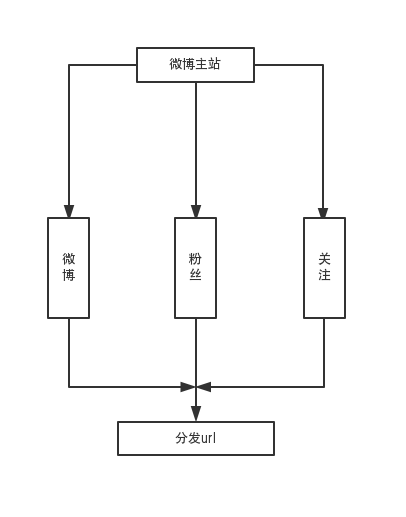


图6-11 url分发流程

其中微博主站是指微博的主页站点，微博，粉丝，关注，分别指的是用户的个人微博站点，用户的粉丝站点，用户的关注列表站点，url分发是指将用户的微博站点，粉丝站点和关注列表站点分别分发到相应的解析函数尽享解析

Url生成的核心代码如下所示。  
 用户的url地址拼接，用来获取用户的关注列表，粉丝列表

user\_url='https://m.weibo.cn/api/container/getIndex?uid={uid}&luico

de=10000011&lfid=230413{uid}\_-\_WEIBO\_SECOND\_PROFILE\_WEIBO&type=uid&valu= {uid}&containerid=100505{uid}'#用户的个人详情连接

关注用户的url拼接以及粉丝列表的url拼接与用户类似，这里就不展示了。

(2)获取原始数据

爬取数据首先爬取用户微博，用户关注列表和用户粉丝列表，然后通过用户关注列表和粉丝列表对用户关注的用户的微博进行了爬取，同时通过用户的粉丝列表对用户粉丝的微博进行了爬取，具体流程如图6-12所示。

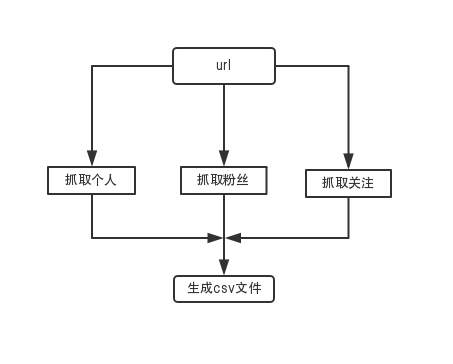


图6-12 爬虫流程

其中url是(1)中生成的url，抓取个人，抓取粉丝和抓取关注，是三个抓取数据的模块，最后把抓取的数据存到csv文件中，用于(3)的主题提取，主要实现如下。

关注列表获取  
 uid = user\_info.get('id')  
 yield Request(self.follow\_url.format(uid=uid,page=1), callback=self.parse\_follows,meta={'page': 1, 'uid': uid})  
 粉丝列表获取和关注列表类似。

爬取用户的粉丝列表分发给微博解析函数

与爬取关注用户同样的原理，对分发过来的url页面进行解析，获得页json数据分析json数据找到与粉丝列表有关的字段，获取关注列表中所有的用户id。

爬取微博则是接收上面的分发请求，对所有id的微博进行数据爬取，并且用re对其中的无用内容进行过滤，将处理好的传给数据处理模块存储到数据库和csv文件中，具体实现和爬取粉丝类似，这里就不展示了。

(3)数据主题提取

本文虽然对其他主题提取方法进行了详细介绍，但是最终主要应用的是lda主题提取方法，所以这里详细着重的说明下lda的主题提取。

数据被数据库和csv存储之后，通过jupyter对数据进行分词，停用词过滤等操作，对端额文本主题进行提取，然后通过自带模块进行画图表展示主题，具体流程如图6-13所示。

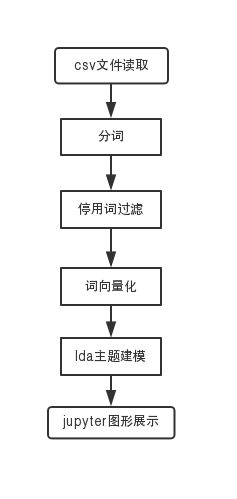


图6-13 主题提取流程

具体实现如下。

停用词：结合多个停用词表  
 custom\_stop\_words=set()  
 stop\_words=stop\_words | custom\_stop\_words  
 stopwords=loadstop\_words(lis)  
 分词，并添加限制条件分词  
 for word in s:  
 if len(word) <= 2 and word not in stopwords and not word.isdigit():  
 s1.append(word)  
 df['text\_cutted']=df.text.apply(chinese\_word\_cut)

词语向量化

tf\_vectorizer = CountVectorizer(strip\_accents = 'unicode',  
 max\_features=n\_features,  
 stop\_words=stopwords,  
 max\_df = 5,  
 min\_df = 1)

Lda主题模型建模  
 lda = LatentDirichletAllocation(n\_topics=n\_topics, max\_iter=50,  
 learning\_method='online',  
 learning\_offset=50.,  
 random\_state=0)

## 6.5 算法结果验证

项目运行测试与维护是整个项目最容易让人忽视的部分，也是检验成果的部分，对整个系统的完整性起着很大的重要性，测试是为了发现程序中的错误，当然了发现错误不是我们进行运行测试的主要原因，最主要的是我们可以通过不断测试，发现项目所存在哪些缺陷，发现bug等等，发现问题当然是为了解决问题，所以我们可以对成果进行改进，与此同时有额外的收获，并且产生一个质量比较高的系统。

维护则是通过将项目交给用户后，在用户使用的过程中，对用户提出的建议，对用户发现的问题，进行改进，这是一个长期的过程，最终在长时间的改进中，让程序越来越优化，结构化越来越高，使用的辅助文档也越来越详细等等，让维护越来越容易，简单地说，我认为维护就是防止程序出问题，无论哪个项目，都不可缺少运行测试，当然至于维护，那是在有用户使用的情况下而言的，对于本项目，同样的运行测试也是必不可少的。

### 6.5.1 测试

在开发项目的过程中，虽然要求的功能单一，但是每个阶段都不可避免地产生很多问题和错误，所以，在项目生存周期的每个时间段，我们都力求，每个阶段要进行严格的检查，希望可以找出某些错误，并进行改进，避免在使用中影响使用，当然了因为所使用的工具的原因仍存在一些小的缺陷，但那时编译器的原因，在后续的过程中，肯定要进行改进的。

项目最大的遗憾就是使用了模块，降低了难度，导致在数据提取阶段，没有使用面向对象也就是类，而是采用函数的方式进行了提取，当然这也没什么，因为项目功能代码比较少，但是如果项目代码进行了扩展，那么使用面向对象的技术会有很大的好处，因为面向对象的特性不仅能够去除重复代码，还能够利于扩展。

本项目的测试就是简单的测试，并没有进行专业的黑盒测试和白盒测试，一反面考虑到项目规模不大，所以简单的测试已经足够了，如果有错误，很明显就能发现，另一方面，是因为本项目不属于前后端项目，仅仅是提取主题，并没有要求取运用主题，就像从网站获取视屏一样，不会有界面，而你如果去运用这些视屏，比如显示在视屏管里系统自己的网页上，这就属于前后端项目了，所以没有良好的界面供用户选择，所以无法进行黑盒测试，当然后期会想办法添加一个界面。

项目的测试主要包括对详细设计部分的内容进行分部分测试，主要有爬取数据的测试，存储数据的测试，爬取主题的测试和主题展示等方面。

### 6.5.2 功能与性能测试

基于详细设计中的爬取数据详细设计进行测试，主要测试数据是否能够爬取成功，和爬取的数据是否符合要求，通过运行entrypoint入口文件，在数据库中，发现爬取的数据发现是已经处理好的，所以，数据获取成功。

提取数据测试，打开数据分析专用软件，运行主题提取文件，发现如果数据量过多，运行的时间会很长，于是做了多线程并发运行，虽然面对数据链非常大的时候依旧如此，但项目并不是一次对所有用户的微博主题进行提取，而是对某一用户和其相关的用户的主题进行提取，所以虽然不是很快出结果，但是也已经满足要求了，功能结果为两种，一种是对美条微博都提取出一个主题，提取的主体测试如图6-14所示，其中每个主题topic由几个词刻画而成。

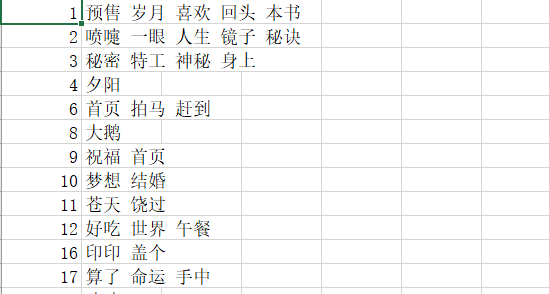


图6-14单条微博数据

该用户的原微博第一条如图6-15所示。

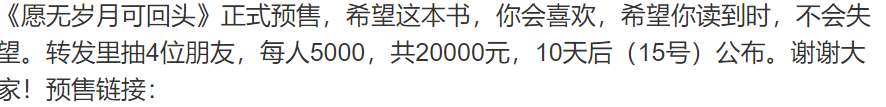


图6-15测试用户的第一条微博

对比第一条可以发现，可以知道主题预售本书，由于微博数据脏乱，停用词无法完全起作用，所以测试成功。

对所测试用户及其好友的微博数据进行关注度最高的主题提取，由于数据量极大，所以不是很精确，其中大的圆圈代表主题，图表是动态的，当鼠标停放在该主题上的时候，右边相应的会出现该主题有哪些词刻画，并表示出其占有的比重，则可以得知这个主题主要是表达的什么，当然lda的弊端在这时就会显现了，它是一种线性回归算法，并不能过于精确，另外通过jupyter可以用图形的形式对结果进行展示。

从本质上讲，除此之外，项目维护是在整个项目过程中实践一些非常简单的价值观的问题。这意味着非常有意识地追踪您的里程碑和目标的进展，而不是假设一切都按计划发生。这意味着通过清晰和持续的沟通保持正常。

一个优秀的项目是永远也不会完工的，他是会不断变化的，所以更新和添加功能也是后期的一个重要部分。

首先要深入下学习关于数据分析的知识，查看有没有可以优化的部分，其次创建一套自己的的美化界面，运用在数据上，其次是合理的运用一些工作软件让你的软件锦上添花，多使用模块和公用文件，并且多做注释，尽量使用常见函数，让代码简洁易懂。

经过这次项目让我接触到了数据挖掘和机器学习，对于这一次毕业设计所用到的内容，我以前完全没有接触过，到现在，我对所用到的机器学习算法lda的内部原理，仍然有迷惑的部分，所以在后续中，我会去深入了解这些公式，然程序的结果变得更加精确，这是我目前最在意的地方，后续肯定会做优化。

后期要做的项目维护，大致这些，到此本项目将告一段落。

# 总结

我们的世界中存在大量原始数据，人类或手动应用程序无法直接处理的数据。由于开发了强大的存储和连接工具，技术作为万维网，工程和科学应用程序和网络，业务服务以及更多数据以指数级增长生成数据。由于这种巨大的数据增长而无法轻易获得有组织的知识和信息，也无法轻易理解或自动提取。这些前提导致了数据科学或数据挖掘的发展，这是一个在信息时代当前世界中越来越多的着名学科。

前文通过对不同方式的主题获取的介绍，对比得出了他们之间的优缺点，通过这些优缺点，可知，在数据挖掘中，使用正确的方式不仅可以使得结果更加准确，而且，也更加快速简洁，使得事情事半工倍。

总而言之，数据挖掘被定义为用于从更大的任何原始数据集中提取可用数据的过程。它意味着使用一个或多个软件分析大批量数据中的数据模式。数据挖掘在多个领域都有应用，如科学和研究。作为数据挖掘的应用，企业可以了解有关其客户的更多信息，并制定与各种业务功能相关的更有效的策略，从而以更优化和更有洞察力的方式利用资源。这有助于企业更接近目标并做出更好的决策。数据挖掘涉及有效的数据收集和仓储以及计算机处理。为了分割数据和评估未来事件的概率，数据挖掘使用复杂的数学算法。

数据挖掘和数据分析，随着技术的发展，大量数据的产生，未来发然前景，是一种必然趋势，值得每个人去研究分析。

参考文献

[ 1]Magnus Lie Hetland.Python基础教程.北京：人民邮电出版社，2010.7

[ 2]François Chollet.Python深度学习.北京：化学工业出版社，2018.1

[ 3]Kim S，Hovy E．Determining the Sentiment of Opinions[c]／/Proc．of the Intl．Conf．on Computational Linguistics(COLING’04)．[S．1]：[S．n]，2004.5

[ 4] Hu M，Ⅱu B．Mining and Summary Customer Reviews [C]／／KDD’04．[s．1]：[S．n]，2004.1

[ 5] Hu M，Liu．B．Mining Opinion Features in Customer Reviews[C]／／AAAI’04，[s．1]：[S．n]，2004.1

[ 6] Popeseu A 0．Etzioni．Extracting Product Features and Opinions from Reviews[C]／／Proc．of Conference on Empirical Methods in Natural Language Processing (EMNLP’05)．[s．L]：[s．n．]，2005.1

[ 7] Yi J，Nasukawa T，Bunescu R．Sentiment Analyzer： Extracting Sentiments about a Given Topic Using Natu— ral Language Processing Techniques[C]／／Proceedings of the 3rd IEEE International Conference on Data Mining．Melbourne，USA：[S．n]．2003.4

[ 8] Wei W，“u H Y，He J，et a1．Extracting Feature and Opinion Words Effectively from Chinese Product Reviews[C]／／Fifth International Conference on Fuzzy ·Systems and Knowledge Discovery(FSKD，2008)．[s． 1]：[S．n]，2008.1

[ 9]Alex Martelli，朱萍.Python Cookbook.北京：人民邮电出版社，2005.1

[10]高磊，戴新宇，黄书剑，陈家骏．基于特征选择和点互信息剪枝的产品属性提取方法[J]．北京：化学工业出版社，2015.1

[11]栗春亮，朱艳辉，徐叶强．中文产品评论中属性词抽取方法研究[J]．北京：清华大学出版社，2009.1

[12]伍星，何中市，黄永文．基于弱监督学习的产品特征抽取叨.北京：人民邮电出版社，2005.1

[13]Mark Lutz.Python学习手册.北京：机械工业出版社，2009.8

[14]罗刚.自己动手写网络爬虫.北京：清华大学出版社，2010.10

[15]孙艳，周学广，付伟．基于主题情感混合模型的无监督文本情感分析[J].北京大学出版社，2013.1

[16]李丕绩，马军，张冬梅，韩晓晖．用户评论中的标签抽取以及排序.人民邮电出版社，2012.1.

[17]欧阳继红，刘燕辉，李熙铭，周晓堂．基于LDA的多粒度主题情感混合模型 [J].人民邮电出版社，2015.1

[18]Luciano Ramalho.流畅的Python.人民邮电出版社，2017.

[19]David M.Beazley.Python Cookbook.人民邮电出版社，2015.

[20]Jeff Forcier.Django Web开发指南.机械工业出版社，2009.

谢辞

感谢我的导师张柱老师，他严谨细致、一丝不苟的作风一直是我工作、学习中的榜样；越有内涵的人越虚怀若谷，像您这样有内涵的人我十分敬佩；从开始的搜集材料，初稿的审核到最后论文的定稿的整个过程中，老师都及时给予我建议，指出不足之处，帮助我更好地组织论文结构，不断充实论文的内容。

在这里要感谢我的大学老师，感谢他们四年的陪伴和教导，让我可以顺利完成我的学业。

另外感谢我的朋友，感谢他们的帮助，个人如果但靠自己，如果置身于集体的关系之外，置身于任何团结民众的伟大思想的范围之外，就会变成怠惰的、保守的、与生活发展相敌对的人。一支优秀的团队，塑造出优秀团队文化，一个优秀的团队文化，酿造出优秀的人。感谢这样的集体。

毕业论文暂告收尾，这也意味着我在安徽理工大学的四年及将结束，回首既往，论文的写作是非常枯燥而又富有挑战的，这也是学术界一直讨论的话题，尽管枯燥，但老师和同学的帮助让我有了进步的勇气和信心。时间的仓促和专业水平的缺乏，整篇论文肯定存在未曾发现的错误和不足，对阅读此篇论的读者仅以致谢。