本科毕业（设计）论文

题　　目　基于大数据平台的股评信息文本挖掘研究

学　　院　信息科学与工程

专　　业　信息与计算科学

学生姓名　蔡　维

学　　号　20141041220　年级　2014 级

指导教师　王伟钧　职称　副教授

二〇一八年五月三日

基于大数据平台的股评信息文本挖掘研究

专　　业：信息与计算科学　　学　　号：201410412201

学　　生：蔡维　　指导教师：王伟钧

摘要：随着互联网的成长和普及，网络平台和论坛已成为股票投资者探讨与会商的主要平台，一些网络的评价信息对股票市场的影响日趋明显。现在处于数据量快速增长的时代，大数据技术成为了数据处理的必要条件。本文针对股票评论信息进行文本挖掘。首先爬取某股票网站的评论数据并利用大数据存储平台 HDFS 进行数据存储；然后通过 Ansj、Kafka 和 SparkStreaming 的整合对文本进行实时分词处理；利用“互信息”算法对所分词汇进行筛选得出与股票相关的特征词；根据筛选的特征词词对股票进行分层评分。最后从不同角度分析股票的得分，为股票投资者提供一定的决策支持。

关键字：股评信息；大数据；文本挖掘；Kafka；SparkStreaming

Research on Text Mining of Stock Comments Based on Big Data Platform

Specialty: Information and Computational Science Student Number: 201410412201

Student: Cai Wei Supervisor: Wang Weijun

Abstract：With the growth and popularization of the Internet，the network platform and BBS became the main platform for stock investors to discuss.The influence of some network evaluation information on stock market is becoming more and more obvious. Now in the era of rapid data growth, big data technology has become a necessary condition for data processing. This paper carries out text mining for stock comment information.Firstly, it use the comment data of the stock website and uses the big data storage platform HDFS for data storage; Through the integration of Anaj, kafka and sparkstreaming, the text is processed in real time; use the "mutual information" method to filter the data to get the words related to the stock; The stock is graded according to the selected word;Finally, analyze the score of stock from different angles and provide some guidance for stock investors.

Key words：Stock Comment Information; Big Data; Text Mining; Kafka; SparkStreaming

目录

绪论

1. 背景

近几年，随着手机应用的崛起，用户以及其他数据的数据量越来越大。2012年开始全球移动数据流量以 66%的平均年增长率递增；在 2017 年的时候全球移动数据总量达到了 134EB。庞大的数据量为公司带来了更多的收益，也为人们提供了方便，但不可避免的也造成了很多的问题。例如过多的数据导致让我们无法很好的定位到关键的数据上，并去理解它。传统的数据库也仅仅是对所获得的数据进行简单的进行存取等操作，然而人们对这些数据的理解也是非常有限的。在股票市场，这种情况也是非常严重的，每支股票所产生的数据也是繁多的，但股票市场的效益丰厚使得越来越多的投资者进入了股票市场。如果没有一种快速合理有效的处理方法对这些数据进行挖掘的话，投资者将会淹埋在巨大的数据中，无法对股市行情进行一个明确的判断和投资，给投资者带来巨大的损失。如果数据仅仅是表现为简单的储存而不经过处理那么这种数据可以认为这是毫无价值的，只有经过处理的数据才能体现出他的价值[1]。随着人们对有用数据的强烈需求数据挖掘技术应运而生，并迅速得到了发展。经过多年的研究和实践，数据挖掘技术吸收各种科学研究成果，形成了独具特色的研究方法，数据挖掘概念也逐步被大多数专家和学者所接受，研究的成果也得到大家的认可，同时也吸引了越来越的研究者，但目前的数据挖掘还是存在许多待研究和探索的问题需要专家和学者继续开发新的挖掘方法[2]。

2. 目的和意义

目的：从海量股市文本信息中挖掘出有用的文本信息，通过大数据平台对数据进行分析，为投资者提供一些决策支持。

意义：①数据挖掘与分析是当前最活跃的话题之一，目的主要是通过量大的原始数据进行分析，从而满足用户的需求。股票投资是机遇和风险并存的。如何做出好的选择？如何投资上一个好的目标，是投资则最大的问题。目前市场上大多数的股票分析软件都是利用传统的统计分析技术（如柱状图分析法、移动平均法、K 线图分析法、趋势分析、基本面分析等），简易层度并不高，对于刚入门的投资者在极端的时间内很难掌握这么多复杂的分析技术。有能力和投资经验丰

富的投资专家也不一定以它为投资依据，大多数的时候都是靠实践的经验来判断。所以采用大数据相关技术和一些分词技术对文本信息进行挖掘。②大数据平台，保证了在数据量大的前提下能快速稳定的解决问题，也保证了数据的可靠性。

3. 研究现状分析

当前大数据技术的研究发展状况主要体现在基础理论、关键技术、应用实践、数据安全等四个方面。大数据充斥着人类经济社会的角角落落，正是因为大数据巨大的商业价值，国内外学者从理论、技术及实践进行了深入的研究。阿尔文·托夫勒在 1980 年就认为大数据是“第三次浪潮的华彩乐章”，IBM 提出大数据有 5V 特点，即 Volume（大量）、Velocity（高速）、Variety（多样）、Value（低价值密度）、Veracity（真实性）[3]。

当前大数据技术在发展过程中所面临的问题主要有两点。首先，现有的 IT 技术架构无法适应大数据技术的发展要求。科学技术的迅速发展推动了企业在数据生成、储存等多方面的长足进步，一方面，企业爆炸式的数据增加加剧了原有数据存的储存压力；另一方面，大量的数据给传统的数据分析处理技术带来巨大挑战。这就要求 IT 行业必须及时革新数据储存和分析处理能力，重构 IT 技术架构以满足大数据的技术需求。其次是传统信息安全措施的失效[3]。

大数据的时代下，通过结合证券行业的业务需求，基于人工智能、数据挖掘、文本挖掘等前沿技术自动分析海量文本数据并从中提取相关有价值信息，证券行业的发展，给各级证券企业带来了挑战，给证券行业带来了互联网业务。智能方向的新机遇推动了证券行业基于文本信息服务的多个创新产业的出现[4]。

对于目前国内股市行业的研究现状，文本挖掘方面的研究比较多。但在大数据时代下，利用大数据的技术实现文本挖掘的研究很少。综上所述，本文利用一些大数据的相关技术来文本数据进行文本挖掘，使用 Jsoup 爬取数据、Ansj 进行文本分词、Kafka 和 SparkStreaming 流数据处理、“互信息”算法筛选词汇。最后通过自己定制的规则对词汇打分。

1. 股票市场概述

1. 1. 中国股市简介

中国股市是中华人民共和国股市。1989 年开始作为试点，本着试得好就上、试不好就停的理念建立。因此，1995 年以前股市操作最大的坏消息是中国股市应该停止试点，股市应该关闭。 受“3. 27 国债期货事件”影响，1995 年中国期货市场全面清理，中国股市成为支撑对象。 因此，股市迎来了真正的利益，并进入了一个大发展的时期。 中国股市最大的特点是国有股和法人股在上市时没有承诺上市。

1. 2. 影响股市因素

1. 2. 1 宏观因素

在影响股市的因素中，宏观经济因素应该是影响最大的因素，原因在于宏观经济因素可以从各个方面影响一个企业的经营以及其股票的价值。公司的运作可以影响人们的收入和心理预期，这将对股市的供求产生巨大的影响。就宏观经济因素而言，不得不说经济周期的“大头”，货币的变化以及国际贸易收支。

1. 2. 2 微观因素

在影响股价波动的微观经济因素中，上市公司是决定其股票价格的主要因素。评估公司的业绩主要集中在公司的净资产，盈利能力，股票拆分和资本扩张，资本削减和增资，营业额，公司股息等相关信息。从公司的行业，竞争地位和经济效率分析公司的增长。

1. 2. 3 市场因素

市场反映了股票的供需环境，股市的供求关系，最后形成股价的条件，所以市场供求关系，市场价格波动，市场投资者构成，交易系统和工具，市场心理等因素都会影响股价。市场因素包括：供求关系，整体价格，投资者构成，交易系统，市场操纵，市场心理等方面。

2. 大数据技术概述

本章主要对使用到的一些大数据技术进行简单介绍，包括 HDFS 的概述、角色、特性以及读写流程；Zookeeper 的概述和 Leader 的选举机制；Kafka 的基本介绍、消息存储、副本策略、消费原理；SparkStreaming 的概述、特性和执行流程。

2. 1 HDFS 文件系统

2. 1. 1 HDFS 概述

与传统文件系统相比，HDFS 是搭建在集群上的分布式文件系统，平常都用于大数据集上的程序，为程序提供一个高容错、高吞吐量和的数据访问平台，对客户端而言，HDFS 就像传统的文件系统。可以使用新建、删除、移动或重命名等方式对文件进行修改。一般的 HDFS 集群里面拥有三种角色。这些角色分别是 SecondNameNode，DataNode 和 NameNode。DataNode 在 HDFS 中负责存放数据，所有数据存放在 DataNode 里面；NameNode 在 HDFS 内部管理元数据, 记载着每个文件所处的位置信息。集群中只拥有一个 NameNode，很容易造成单点故障，所以想出了另一种搭建集群的方式。搭建一个 HA 的 HDFS 集群，这样的话会在两台虚拟机上各自拥有一个 NameNode。

2. 1. 2 HDFS 角色简介

NameNode：

NameNode 起一个元数据管理的作用，记载了数据存储的节点信息和位置信息，用户向 NameNode 发出访问指令来对数据的访问，NameNode 则返回给用户一个数据存储的 DataNode 的信息。NameNode 中带有：文件名和数据块之间的联系（由 edits 和 fsimage 两个文件来实现），数据块和 DataNode 节点之间的联系。edits 文件与 fsimage 文件是 NameNode 结点上的主要文件。

DataNode：

HDFS 中真正储存数据的节点。DataNode 中的文件是以数据块的方式进行存储，DataNode 也会时刻和 NamoNode 保持着心跳，只要这个心跳停止，那么 NamoNode 就会认为该服务器上的存储节点死掉了。会让其他服务器节点备份一份这个服务器的数据。

2.1.3 HDFS 特性

存储在 HDFS 文件系统中的文件被按服务器配置中配置的大小大小切分成块，随后将这些块备份到多个服务器中的中 DataNode 节点中，默认是三份，一份是本机上，一份是同一机架的不同机器上，一份是不同机架的不同机器上，这样就既能保证数据的快速访问，又能保证数据的容灾。块的大小默认为 128M，低版本为 64M，当然块的大小和备份的块数量可以在配置中进行修改。

2.1.4 HDFS 读写流程

写流程通过 Client、NameNode 和 DataNode 节点来实现（参见图 2-1）。

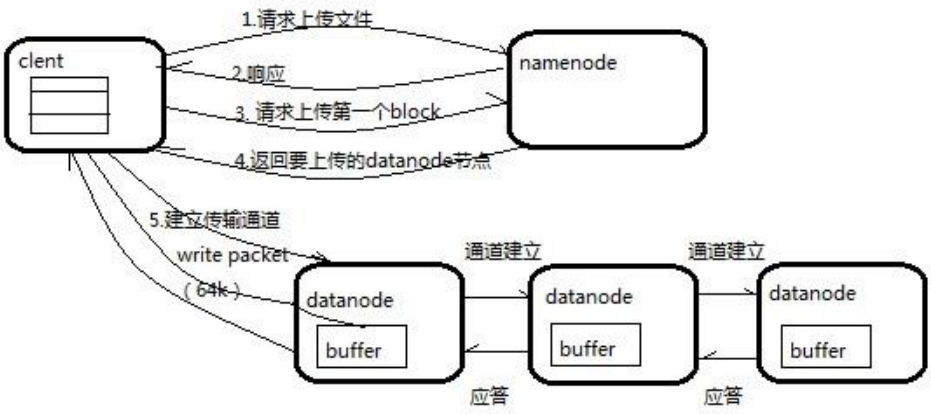

图 2-1 HDFS 写文件流程

1、客户端和 NameNode 发送信息请求上传文件到 HDFS，NameNode 检查目标文件信息。

2、NameNode 给客户端答复能不能上传。

3、当返回可以上传后，客户端将首先对文件进行切割，如果配置的一个数据块 128M 的话，那么一个 300m 的文件就会被切割成 3 个块，两个 128m 的数据库、一个 44m 的数据块，然后请求一个 DataNode 服务器并上传第一个数据块。

4、NameNode 返回 DataNode 的服务器的信息。

5、客户端请求上传数据（基本上是一个 RPC 调用，建立流水线），第一个 DataNode 接收数据后将继续调用第二个 DataNode，依次下去。

6、客户端开始往 A 上传第一个数据块，以 packet 的单元（packet 是 64KB）。当写入数据时会检查数据。它不是以一个 packet 为单位的检查，而不是一个 chunk 为单元（512 字节）。

7、当一个数据块传输完成以后，client 再次向 NameNode 发出请求来获得上

传第二个数据块的服务器具体信息。

读流程通过 Client、NameNode 和 DataNode 节点来实现（参见图 2-2）。

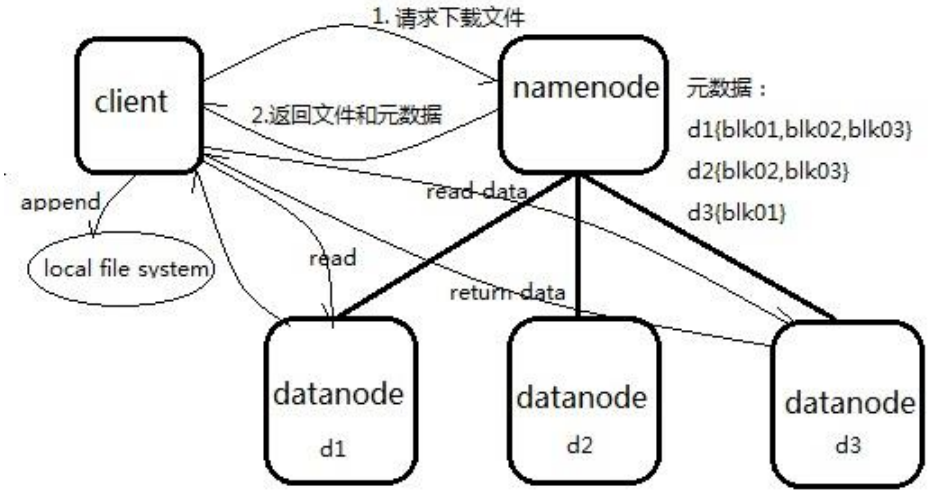

图 2-2 HDFS 读文件流程

1、客户端和 NameNode 发送读文件消息。NameNode 会查询目标元数据，找到目标文件块所在的 DataNode 服务器。

2、然后 NamoNode 会挑选一台相对较近的 DataNode（就近原则，然后随机）服务器。

3、DataNode 开始发送数据到客户端（从磁盘里面读取数据放入流，以 packet 为单位来做校验）。

4、客户端按照 packet 为单位一个一个接收，先进行本地缓存，等一个文件块被发送完后，在将缓存的数据写入目标文件中，然后进行下一个数据块的缓存。之后的数据块就依次的追加到前面的数据块，最后合并生成最终需要的文件。

2.2 Zookeeper 协调服务

2.2.1 Zookeeper 概述

Zookeeper 是一种服务于分布式程序 HBase，Kafka 等所设计的高新能、高可用且一致的协调应用。Zookeeper 的优点有：服务器进程之间的彼此排斥和协作、简单的分布式协调、序列化（按照特定的规则对数据进行编码）、有序的消息、原子性、可靠性。Zookeeper 特征使它能够在分布式的程序当中得到使用。从 Zookeeper 可靠性来讲，它是没有单点故障的，Zookeeper 必须存在 2n+1 个节点。从分布式协调来讲，它能够保证分布式程序中数据的一致性。所谓分布式数据一致性，即让集群中传递的数据保持一致。

2. 2. 2 Leader 选举机制

1、当 master 的 Zookeeper 启动时，通过 zoo. cfg 中配置的其他 Zookeeper 机器进行通讯，发现只有一台 Zookeeper 启动，不能提供服务。

2、当 slave1 的 Zookeeper 启动时，通过 zoo, cfg 中配置的 Zookeeper 机器进行通讯。发现此时已经启动了两台 Zookeeper。超过 Zookeeper 集群的半数。Zookeeper 间进行选举。

选举投票：

① 每个 Zookeeper 都投递出自己的 myid。Master 投出 0，slave1 投出 1.

② 投票结果不统一。进行第二轮投票

③ Master 发现自己的 id 小于 slave1 的。则投递出 1，slave1 发现自己的 id 大于 master，则继续投递 1

④投票结果统一，则 leader 为 slave1. master 身份为 follower.

3、当 slave3 启动时，发现已经由 Leader 存在了。则以 Follower 身份与 Leader 进行通讯（同步数据）。

2. 3 Kafka 消息系统

2. 3. 1 Kafka 基本介绍

Kafka 由 Linkedin 公司开发并发布，它是一个支持多个生产者、多个消费者、多个分区、单个分区多副本消息系统，它也是一个基于 Zookeeper 管理的可横向拓展的分布式消息系统。常常作为流式数据处理程序的高级数据来源。Kafka 最终被捐献给 Apache 基金会，并成为 Apache 的 TOP 项目。

Kafka 主要的设计目标：

1. Kafka 利用时间换空间的方式，在数据量特大的时候也能保证正常的访问性能。

2. 高吞吐量，即使在配置底下的商用机器上也能做到每秒 100KB 的消息传输能力

3. 支持 topic 的分区，将一个 topic 分为多个区；支持分区消息的多副本。每个分区内的消息始终是保持着顺序传输，这是因为同一个分区的消息是以 offset 的顺序进行存储。

4. 支持离线和实时两种数据处理。

一个原始 Kafka 集群应当包括多个消费组、多个消费者、多个生产者、多个 Broker 和一个 Zookeeper 集群。Zookeeper 为 Kafka 管理消费者消费数据的偏移量和一些配置。包括 Kafka 的 Leader 选举等。生产者利用推送消息的方式向 Kafka 的 Broker 推送消息，消费者使用拉取消息的方式从 Kafka 的 Broker 获取消息。

Kafka 包括的角色有：Broker、Topic 、Partition、Replication、Segment、offset、Producer、Producer、Consumer 和 Consumer Group。

2.3.2 Kafka 消息存储

在 Kafka 中一个 Topic 通常存储的时同一类的消息，类似于在新闻中的某一类新闻，每个 Topic 是由多个分区组成，其中每个分区在磁盘中是以 append log 文件存放。

每个分区文件被切分为多个大小相等但消息数量不一定相等 segment 数据文件。这种存储方式让老的 segment 文件能快速被删除。每个分区只负责顺序读写文件，segment 文件生命周期由服务端配置参数决定。

2.3.3 Kafka 副本策略

Kafka 在 0.8 版本才开始提供分区的副本机制，为什么提出副本机制呢，因为之前的分区都是单节点，只要分区节点宕机，那么这个分区的数据全部都将丢失，并且该分区的节点不能提供所有的服务了。而分区中又没有备份的数据，致使数据可靠性极低。所以 0.8 后提供了副本机制来保证 Kafka Broker 节点的故障转移。

引入副本策略以后，一个分区会有多个副本，而这时需要在这些副本之间选举出一个 Leader，生产者和消费者只与这个 Leader 交互，其它副本作为 Follower 从 Leader 中拉取数据（参见图 2-3）。

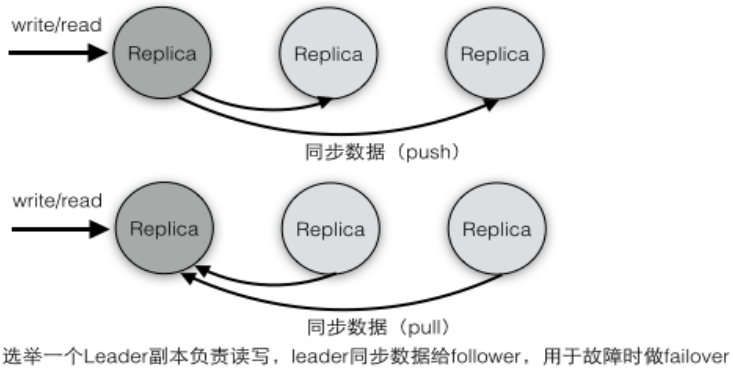

图 2-3 Kafka 副本同步

2.3.4 Kafka 消息消费原理

一个 Topic 的一条消息只能被同一个消费组的一个消费者消费，不同消费组的不同消费者可以消费同一条消息。

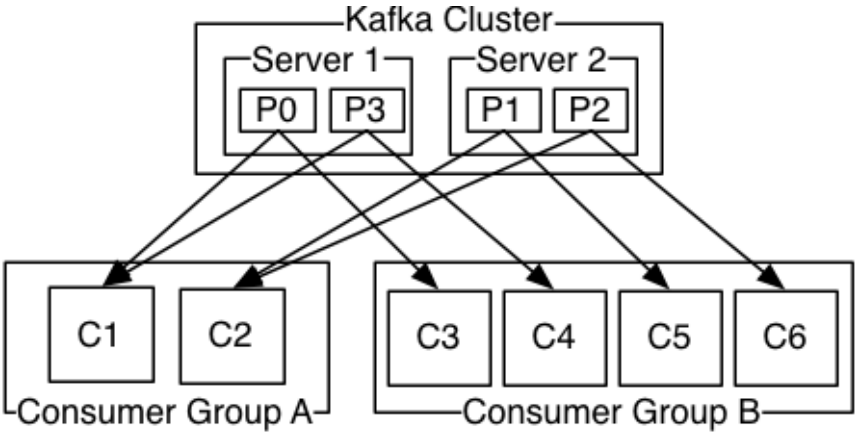

图 2-4 Kafka 消费原理

因为 Kafka 是一个消息系统，所以 Kafka 也像传统消息系统（MQ）一样，通过发送消息和消费消息来传输数据，选择由生产者向 Broker 推送消息并由消费组从 Broker 拉取消息。

推送模式很难适应消费速率不同的消费者，因为消息发送速率控制权在 Broker 上。推送模式的目的是尽量以最快传输速度传递消息，这样就很容易造成消费者来不及处理消息，然后造成数据丢失和网络拥塞。而拉取模式的消费速率可以由消费者本身的消费能力来自主控制。

对于 Kafka 消费者而言，拉取的模式更合适。拉取模式可降低 Broker 的工作量。

2.4 SparkStreaming 流处理

2.4.1 概述

SparkStreaming 作为 Spark API 核心的拓展，支持实时数据流并提供容错的、高吞吐量数据的处理。支持从多种数据源获取数据，包括 Kafka、Flume、Twitter 等。从数据源获取数据后，可以使用 join、map、reduce、second 和 window 等高级函数进行复杂算法的处理。最后还可以将处理结果存储到分布式文件系统、数据库等。

2.4.2 特性

SparkStreaming 是一个对实时数据流进行高吞吐量、容错处理的流式处理系

统，可以对多种数据源（如 Kdfka、Flume、Twitter、Zero 和 TCP 套接字）进行类似 Map、Reduce 和 Join 等复杂操作，并将结果保存到外部文件系统、数据库或应用到实时仪表盘。

2.4.3 执行流程

（1）Application 运行于 StreamigContext 和 SparkContext 之上。

（2）SparkStreaming 为输入源启动接收器 Receiver，Receiver 以任务 Task 的形式运行在应用的执行器 Executor 中。

（3）Receiver 接受输入源并拆分为块（离散化），这些块其实就是 RDD。

（4） Receiver 收集到输入数据后会把数据复制到另一个 Executor 中来保障容错性。

（5）SparkContext 会周期性的（由开发者设置的时间间隔决定）在内存中运行 job 来处理这些 RDD 数据。

3. 股评文本数据采集概述

3. 1 网络爬虫概述

网络爬虫（又被称为网页蜘蛛，网络机器人，在 FOAF 社区中间，更经常被称为网页追逐者），是一种按照一定的规则，自动的抓取万维网信息的程序或者脚本，已被广泛应用于互联网领域[4]。搜索引擎使用网络爬虫抓取 Web 网页、文档甚至图片、音频、视频等资源，通过相应的索引技术组织这些信息，提供给搜索用户进行查询。网络爬虫也为中小站点的推广提供了有效的途径[4]。

3. 2 Java 爬虫 Jsoup 技术

Jsoup 是一个用于处理 HTML 的 Java 库。它提供了一个非常方便的 API 来提取和操作数据。Jsoup 有自己专门的 DOM 代码体系。如果想用 XML 的 API 来操作 Jsoup 是不行的，但也是由于这个原因，使得 Jsoup 里没有 XML 相关的诸多 API，让代码变得很简洁。

1. 从 URL，文件或字符串解析 HTML
2. 使用 DOM 遍历或 CSS 选择器查找和提取数据
3. 操纵 HTML 元素，属性和文本
4. 根据安全的白名单清理用户提交的内容，以防止 XSS
5. 输出整齐的 HTML

4. 文本分析相关技术与算法概述

4.1 文本分词

4.1.1 中文分词概述

汉语分词（中文分词）是指将一个汉字序列或一个句子分割成一个单词。分词是根据一定的规则将连续的单词序列重组成一个单词序列的过程。在英语中，词语是由空间隔开的，而汉语中只有句子和段落可以通过不同的划界来划界，即没有一种形式的划分。即使在英语中，也存在短语划分的问题，但是在词类划分方面，汉语词语划分是比英语要复杂得多，难度也大。中文分词是文本挖掘的基础部分。汉语分词应用于汉语的一段，可以自动识别句子的意义。

4.1.2 Ansj 中文分词概述

Ansj 中文分词是基于 n-Gram+CRF+HMM 的 Java 实现。Ansj 的分词速度很快，能到达每秒 200W 字左右，在高速分词过程中同时也能达到差不多有 96%的准确率。目前实现了分词，姓名识别和关键字提取等一些功能。Ansj 包括了四种分词模式，分别是：基本分词、精准分词、NLP 分词和面向索引分词。其中 NLP 分词所具有的分词功能最多，但大多数时候采用的是精准分词的方式。

4.2 分词数据清洗

4.2.1 数据清洗概述

数据清洗指的是讲原有的数据通过自己定义的规则来进行筛选，清洗。

数据清洗从名字上来看就是对数据中的脏数据进行清洗，然后把无用的数据丢弃，只留下觉得好的数据，或则有用的数据。那什么样的数据脏数据，什么样的数据优势好的数据呢，这个是由项目中需要的数据类型，或者数据于该项目的相关性来决定的。比如本文是针对股票评论信息的研究，那么在这个项目中进行数据清理后的数据都必须能与股票相关联。否者就会被视为脏数据进行丢弃。当然数据清洗也包括清除掉重复的数据。

4.2.2 互信息过滤算法概述

互信息是计算语言学模型分析的常用方法，它度量两个对象之间的相互性。在过滤问题中用于度量特征对于主题的区分度。互信息的定义与交叉熵近似。互信息本来是信息论中的一个概念,用于表示信息之间的关系， 是两个随机变量统

计相关性的测度，使用互信息理论进行特征抽取是基于如下假设：在某个特定类别出现频率高,但在其他类别出现频率比较低的词条与该类的互信息比较大。通常用互信息作为特征词和类别之间的测度，如果特征词属于该类的话，它们的互信息量最大[5]。由于该方法不需要对特征词和类别之间关系的性质作任何假设，因此非常适合于文本分类的特征和类别的配准工作[5]。

一般地，两个离散随机变量 X 和 Y 的互信息可以定义为：

$$\mathrm{I(X;Y)} = \sum_{y\in Y}\sum_{X\in x} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \qquad (1)$$

公式（1）中 $p(x,y)$ 是 X 和 Y 的联合概率分布函数，而 $p(x)p(y)$分别是 X 和 Y 的边缘概率分布函数。

在连续随机变量的情形下，求和被替换成了二重定积分：

$$\mathrm{I(X;Y)} = \int_Y\int_X p(x,y)\log\left(\frac{p(x,y)}{p(x),p(y)}\right) \qquad (2)$$

公式（2）中 $p(x,y)$当前是 X 和 Y 的联合概率密度函数，而$p(x)$和$p(y)$分别是 X 和 Y 的边缘概率密度函数。

5. 系统设计概述

本章主要针对本文章设计的系统进行简单的描述，描述分为系统架构、系统模块概述和数据库设计三个方面。其中系统架构描述系统的设计思路和检点介绍了系统的流程；系统模块描述主要介绍了各个模块所做的工作；数据库设计介绍了数据库表的设计和数据库 E-R 图。

5. 1 系统架构

5. 1. 1 系统架构概述

系统分为四个模块：爬取数据模块，实时分词模块，词汇筛选模块, 股票打分模块, 通过 Java 和 Scala 编写。由 Maven 对项目就能行管理。这四个运行在三台 Linux Centos 7 虚拟机上。三台虚拟机分别为 master、slave1、slave2。按照传统搭建系统的方式，爬取数据，将数据传给后面的代码进行处理是容易理解的一种方式，但是在大数据量的情况下，程序间高度的耦合会造成不可挽回的局面，当前一步执行失败之后，程序后面的执行过程全部会跟着一起失败，这是一直想避免的，所以在程序设计上，追求的是高聚合、低耦合。在本系统中，采用了消息系统中间件，很好的达到了程序间的解耦，如果在消息的上一层出现了程序中的错误，并不会带向消息系统的下一层。当消息系统的下一层出现了错误，并不会将错误返回给消息系统的上一层，消息系统只负责了系统中数据的传输和存储。系统中同时提供了多种存储方式接口，爬取好的数据可以通过调用相关的接口，存入相关的文件系统中。当系统传输的数据量和需要处理的数据量很多的时候。系统中 HDFS、Kafka 和 Spark 组件都可以进行横向扩展来增加系统的处理能力和处理速度。系统的架构图如图 5−1 所示：

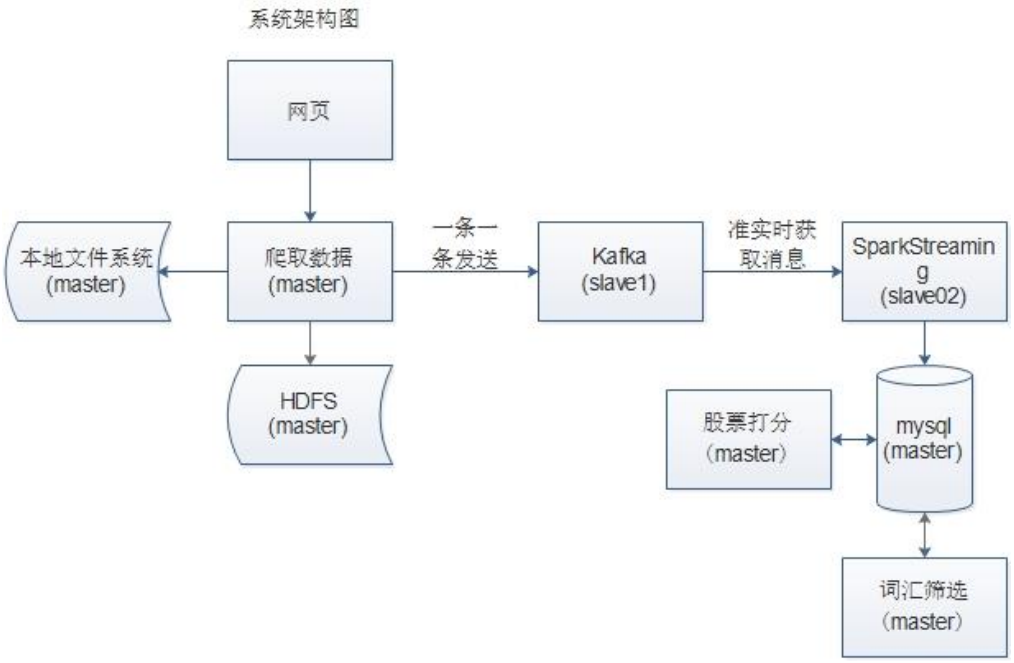

图 5-1 系统架构图

5.1.2 系统流程简要概述

该系统通过从网页爬取股票的文本信息，在实时爬取的时候，将数据实时的发送给 Kafka 消息系统，Kafka 消息系统的数据被 SparkStreaming 组建实时消费。爬取的数据也可以调用存本地系统和存分布式文件系统 HDFS 的接口。分词后的数据存入 MySQL 中，存入 MySQL 的数据被词汇筛选组件获取，通过一些过滤算法对词进行筛选，筛选过后调用股票打分组件，通过对词汇的打分，最终得出股票在各个层次的分数。

5.2 系统模块概述

5.2.1 爬取数据模块

该模块通过 Jsoup 解析网页爬取相关股票评论信息，在对每一条信息进行处理、筛选之后。将该条信息发送给 Kafka。在处理完并发送完所有数据后，可调用不同的接口将数据放入不同的文件系统中。

5.2.2 实时分词模块

该模块是一直运行在 Linux Centos 7 上的程序，程序一直保持活跃，当 Kafka 消息系统中有数据之后，就会从其中拉取消息下来进行分词处理。排除网络 IO 的影响，基本上和爬取数据保持同时进行。分词后的数据存入 MySQL 数据库。

5.2.3 词汇筛选模块

该模块从 MySQL 获取数据，进行两轮筛选，首先通过词汇模板进行筛选，将每支股票词汇模板频数排名前 8 的相关词汇进行保留，在通过使用互信息的方式将分词数据进行筛选。最终将筛选的数据存入 MySQL 数据库。

5.2.4 股票打分模块

该模块从 MySQL 获取数据，对词汇进行打分，打分的依据为该词汇的语句中是否存在褒义或者贬义词，最终获得每个词的最终分数，再将这些词汇通过手动划分层次，划分为三层。最后通过加权平均获得每支股票每个层面的得分，将数据存入 MySQL 数据库。

5.3 数据库设计

5.3.1 MySQL 数据库概述

MySQL 是现在比较流行，在组织中普及比较广泛的 RDBMS。也就是我们常说的关系型数据库管理系统。最初是由一个瑞典公司研发的。现在被甲骨文收购了。MySQL 在百万级别以下的数据量可以提供毫秒级别的响应。MySQL 表中的数据是以行为单位进行存储，每一列的数据类型是固定的。由于开发成本低，MySQL 是很多中小型网站数据库的最佳选择。MySQL 的 SQL 是 SQL92 标准。除了支持一些聚合，求和等函数，MySQL 还支持很多其他函数，比如 GROUP_CONCAT 函数、AES_ENCRYPT 函数等。MySQL 也支持很多拓展，比如讲 MySQL 搭建为分布式的，让 MySQL 的读写进行分离等等。这样也就可以让 MySQL 的数据多了一份副本，保证了数据的可靠性。

5.3.2 数据字典

数据库采用 UTF-8 编码，其中的表有：股票信息表，评论表，股票词汇表，筛选词汇表，股票评分表。

股票信息表（stock）：

主要字段有：主键 ID，股票代码，股票名称，行业名称，爬取标志，筛选标志，打分标志。如表 5-1 所示。

表 5-1 股票信息表

字段名	数据类型	宽度	约束	描述	备注
Id	Varchar	64	非空	编号、主键	UUID 生成
Stock_code	Varchar	10		股票代码	股票的代码
Stock_name	Varvhar	20		股票名	股票名称

Industry	Varchar	20		行业	股票所属行业
Jsoup_flag	Tinyint	1		爬取标记	1 代表未被爬取，0 代表被爬取
Screen_flag	Tinyint	1		筛选标记	1 代表未被筛选，0 代表被筛选
Score_flag	Tinyint	1		打分标记	1 代表被打分处理，0 代表未被打分

评论表（comment）：

主要字段有：主键 ID，股票代码，评论内容，创建时间。如表 5-2 所示。

表 5-2 评论表

字段名	数据类型	宽度	约束	描述	备注
Id	Varchar	64	非空	编号、主键	UUID 生成
Stock_code	Varchar	20		股票代码	股票代码
Comment	Varcahr	255		评论	股票的评论
Create_time	date	0		创建时间	爬取时间

股票词汇表（stock_term）：

主要字段有：主键 ID，股票代码，评论 ID，句子 ID，词号，词，词性，创建时间。如表 5-3 所示。

表 5-3 股票词汇表

字段名	数据类型	宽度	约束	描述	备注
Id	Varchar	64	非空	编号、主键	UUID 生成
Stock_code	Varchar	20		股票代码	股票代码
Comment_id	Varchar	64		评论 ID，外键	评论表中的 ID
Sentence_id	int	11		句子 ID	句子在本段中的排序
Stock_term_id	Int	11		词 ID	词在句子中的排序
Stock_term	Varchar	255		股票词	股票评论切分后的词
Nature	Varvhar	255		词性	词的词性
Create_time	Date	0		创建时间	分词时间

筛选词汇表（screen_term）：

主要字段有：主键 ID，股票代码，评论 ID，句子 ID，词号，词，词性，创建时间。如表 5-4 所示。

表 5-4 筛选词汇表

字段名	数据类型	宽度	约束	描述	备注
Id	Varchar	64	非空	编号、主键	UUID 生成

Stock_code	Varchar	20		股票代码	股票代码
Comment_id	Varchar	64		评论 ID，外键	评论表中的 ID
Sentence_id	int	11		句子 ID	句子在本段中的排序
Stock_term_id	Int	11		词 ID	词在句子中的排序
Stock_term	Varchar	255		股票词	股票评论切分后的词
Nature	Varvhar	255		词性	词的词性
Create_time	Date	0		创建时间	筛选时间

股票得分表（stock_score）：

主要字段有：主键 ID，股票代码，宏观分数，微观分数，市场分数，创建时间。如表 5-5 所示。

表 5-5 股票得分表

字段名	数据类型	宽度	约束	描述	备注
Id	Varchar	64	非空	编号、主键	UUID 生成
Stock_code	Varchar	20		股票代码	股票代码
Mac_score	Double	11		宏观得分	宏观层面得分，小数点后面保留 5 位
Mic_score	Double	11		微观得分	微观层面得分，小数点后面保留 5 位
Mar_score	Double	11		市场得分	市场层面的份，小数点后面保留 5 位
Create_time	datetime	0		创建时间	打分时间

5.3.3 数据库 E-R 图设计

数据库中有五张表，分别是股票信息表(stock)，评论表(comment)，股票词汇表(stock_term)，筛选词汇表(screen_term)，股票得分表(stock_score)。其中评论表的 id 对应筛选词汇表和股票词汇表的 id；评论表的 stock_code 对映股票信息表中的 stock_code。筛选词汇表和股票词汇表的 stock_code 对应股票信息表中的 stock_code。股票评分表的 stock_code 对应股票信息表的 stock_code。对应的数据库 E-R 图如图 5-2 所示。

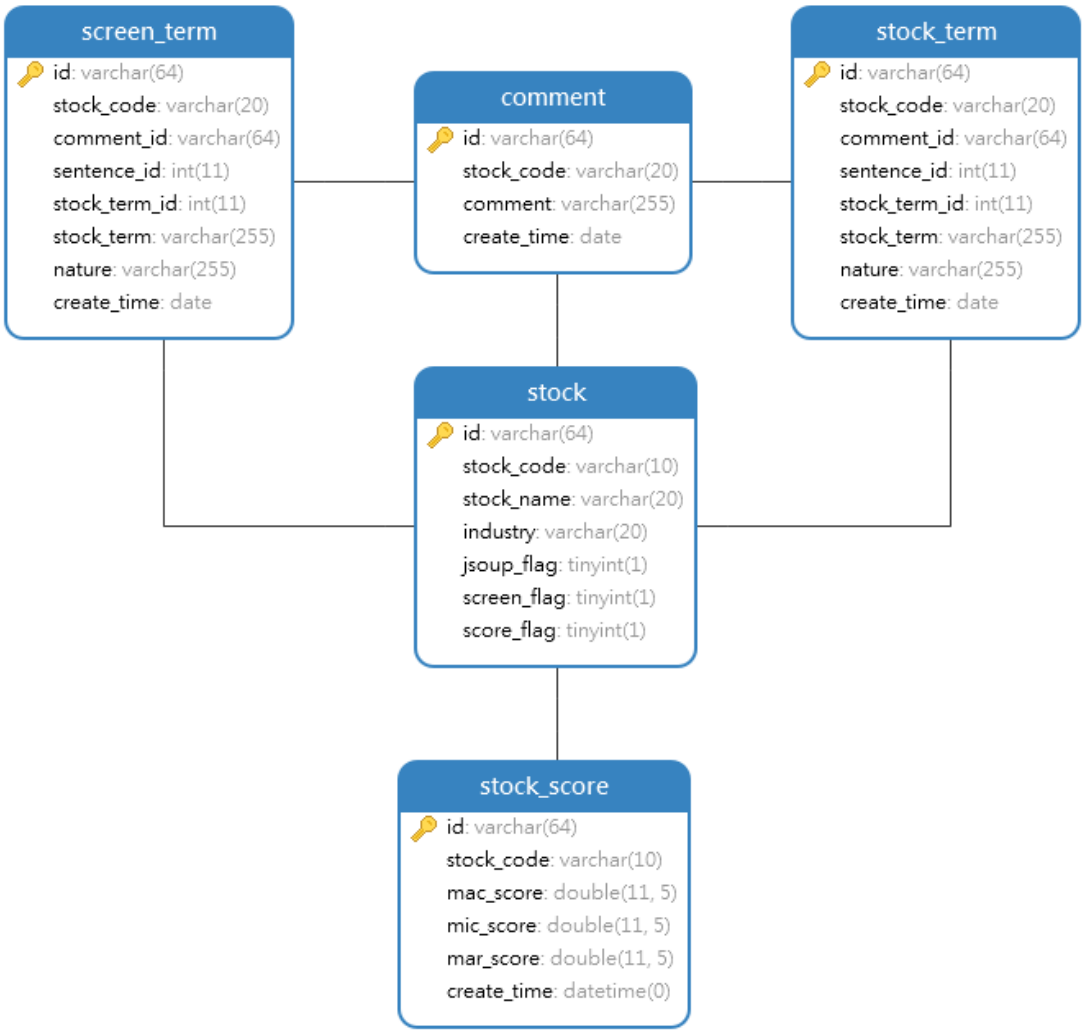

图 5-2 数据库 E-R 图

6. 基于大数据平台的股评信息文本挖掘实现

本章主要讲述了基于大数据平台的股评信息文本挖掘的实现过程，其中包括分布式文件系统搭建，消息系统搭建，数据爬取实现，大数据实时分词，词汇数据清洗，股票打分和结构分析。

6.1 分布式文件系统搭建

本节详细介绍了 JDK 的安装、分布式文件系统 HDFS 的搭建以及 HDFS 文件系统的配置优化。

6.1.1 JDK 的安装

在官网下载 JDK1.8 版本，下载完成后上传 JDK 文件至虚拟机，通过解压命令 tar -zxvf 解压文件。在环境变量中配置 Java 的环境变量，配置好之后执行 source /etc/profile 应用配置，通过命令行输入 java 如果出现 Java 相关的内容说明 JDK 安装成功。

6.1.2 HDFS 安装

在官网下载 Hadoop 2.6.5 版本，下载完成后上传文件至虚拟机，通过解压命令 tar -zxvf 解压文件。本文中使用伪分布式安装，也就是只需要一台主机，但是可以提交相应的出口给其他程序访问。启动后三个角色都在同一台虚拟机上。

在对外提供访问地址时，一般不建议用 IP 地址，而是用主机名。一般情况在网络中，IP 会出现冲突的问题，如果有很多软件在该主机上，当 IP 发生改变之后，所有用到 IP 的地方都需要修改。如果用主机名，只需要修改主机名和 IP 的映射关系就可以了。打开/etc/hosts 文件在其中添加映射即可。

配置 Hadoop：进入 Hadoop 配置文件目录，在 hadoop-env.sh 文件中，修改 JAVA_HOME 为的 JDK 安装目录；创建入口：在 core-site.xml 文件中加入 fs.defaultFS 属性；修改副本数：在 hdfs-site.xml 文件中添加 dfs.replication 属性；配置环境变量；配置完成，首次启动之前要先对 hadoop 执行格式化，格式命令为:hdfs namenode -format;格式化之后通过启动命令：start-dfs.sh 启动 HDFS。查看进程，如果 Namenode、DataNode、SecondaryNameNode 三个角色都成存在的话，说明 HDFS 安装成功。对应文件的配置如表 6-1 所示：

表 6-1 HDFS 相关配置

文件名	属性名	属性值

core-site. xml	fs. defaultFS	https://localhost:9000
hdfs-site. xml	dfs. replication	1

6. 1. 3 HDFS 优化配置

HDFS 默认是将数据文件存放在/tmp 目录下，所以有可能会数据丢失，所以在 core-site. xml 加上 hadoop. tmp. dir 属性的配置，由于目录格式发生了改变，所以需要重新对 HDFS 格式化。对应的优化配置如表 6-2 所示：

表 6-2 HDFS 优化配置

文件名	属性名	属性值
core-site. xml	hadoop. tmp. dir	/data

当启动 hdfs 时，会输入三次登录密码，当主机数很多时，会出现输多次密码的情况，所以一般情况下，为了操作方便，会采用免密。在任意目录下执行：

```
ssh-keygen -t dsa -P '' -f ~/. ssh/id_dsa
```

然后将公钥放在指定文件下：

```
cat ~/. ssh/id_dsa. pub >> ~/. ssh/authorized_keys
```

hdfs 默认会提供一个 WEB 界面，可以通过 windows 浏览器进行访问，但一般情况下，一般是访问不了的，因为没有关闭 Linux 防火墙，执行命令 systemctl stop firewalld 关闭防火墙。

6. 2 消息系统中间件的搭建

本节对 Kafka 消息系统搭建进行了详细说明，以及在消息系统中创建 Topic 和一些参数说明。

6. 2. 1 Kafka 搭建

Kafka 是依赖于 Zookeeper 的，所以在安装的时候要先安装 Zookeeper，在单机版 Kafka 中，Zookeeper 是自带的。这里只介绍单机版的安装。从官网下载 Kafka 的压缩包，上传到 Linux 虚拟机中，使用 tar -zxvf 命令解压文件，进入/etc/profile 文件中配置环境变量。

进入 kafka 的工作目录，通过以下命令启动 Zookeeper：

```
zookeeper-server-start. sh config/zookeeper. properties
```

这种方式启动，会挂载在当前的窗口，最好采用后台运行，命令为：

```
nohup zookeeper-server-start. sh config/zookeeper. properties &
```

Zookeeper 启动之后，启动 Kafka, 启动命令为

nohup kafka-server-start. sh config/server. properties &

6. 2. 2 Kafka 中 Topic 的创建

生产者在生产数据和消费者在消费数据的时候，都是在一个主题中进行的，但是刚刚搭建的 kafka 是没有主题的。可以通过 kafka-topic. sh 脚本在 kafka 中创建主题。创建命令如下：

kafka-topics. sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic test001

其中的参数解释如表 6-3 所示：

表 6-3 Kafka 创建 Topic 参数

属性名	属性值	备注
--create	无	创建命令
--replication-factor	1	副本数
--partitions	1	分区数
--topic	test001	主题名
--zookeeper	localhost	Zookeeper 连接地址

6. 3 数据爬取模块

本节主要是对数据爬取模的详解。包括了数据源介绍，Jsoup 提取网页数据，Kafka 生产者接口，存储数据到本地文件接口，存入分布式文件系统接口和存入 MySQL 接口。

6. 3. 1 网站数据源以及地址解析

通过在网上筛选，最终选择在证券之星网站爬取股票的评论数据。网页的内容如图 6-1 所示：

序号	代码	简称	收盘价(元)	涨跌幅(%)	日期	评论
1	900905	老凤祥B	3.611	1.064	2018-04-27	继续震荡格局:观望。
2	900905	老凤祥B	3.573	-0.557	2018-04-26	短期走势已走弱:减磅为主。
3	900905	老凤祥B	3.593	-0.554	2018-04-25	回调没有到位:逢高减磅。
4	900905	老凤祥B	3.613	0.083	2018-04-24	仍无转强迹象:减磅。
5	900905	老凤祥B	3.610	-0.824	2018-04-23	仍有下调压力:建议减磅。
6	900905	老凤祥B	3.640	-0.628	2018-04-20	主力小幅洗盘:逢低介入。
7	900905	老凤祥B	3.663	1.048	2018-04-19	走势属多头势:介入为主。
8	900905	老凤祥B	3.625	-0.685	2018-04-18	回调没有到位:建议减磅。

图 6-1 爬取网页

网页整体展现出来的是一个表格形式，其中又分为了很多页，爬取会将每一页的数据都会进行爬取。在爬取的时候，需要爬取其中的代码，日期和评论，数据的主键 ID 通过 UUID 生成。

爬取的地址为：http://stock. quote. stockstar. com/stockinfo_info/comment. aspx?code=900905&pageid=2。通过观察规律发现网址的组成是：http://stock. quote. stockstar. com/stockinfo_info/comment. aspx?code=″+股票代码+″&pageid=″+页数。所以在进行数据爬取的时候，只需要将股票的代码和相应的页数传递进去就可以对网址进行访问。

6. 3. 2 Jsoup 提取网页关键数据

调用 Jsoup Api 中的 connect 接口可以解析传递进去的 URL。之后调用 get 方法就会将 URL 中的网页以源代码的形式返回回来。通过 Jsoup Api 中的 select() 方法可以定位到想要的 class。最后通过 text() 方法拿出标签中的值。拿出的一条消息存放在一个 Comment 对象中，每一条 Comment 提取出来后，就会调用 Kafka 生产者接口，并将数据存放进一个集合中。

6. 3. 3 Kafka 生产者接口

在程序刚刚开始运行的时候，会创建一个 Kafka Producer 对象，该对象的生命周期在这个程序结束后就会结束。该对象所依赖的配置已经相应的解释如表 6-4 所示：

表 6-4 Kafka Produce 配置

属性名	属性值	备注
bootstrap. servers	localhost:9092	配置 cluster 中 borker 的 host/port 对
batch. size	1024	RecordBatch 的最大容量。默认值是 16384（16KB）。
linger. ms	5000	Socket ：solinger。延迟。默认值：0ms，即不延迟。
key. serializer	org. apache. kafka. common. serialization. StringSerializer	配置序列化类名
value. serializer	org. apache. kafka. common. serialization. StringSerializer	配置序列化类名

当一个 Comment 对发送到生产者接口的时候，通过调用 Producer 对象中的 send 方法，并传入需要发送的消息，发送之后调用 flush() 方法将消息保存到 kafka

消息系统中去。

6.3.4 评论数据存本地文件接口

当一直股票的所有评论爬取完成之后，这时候有所的评论在一个集合中，这时可以调用存放本地文件的接口，该接口接收一个集合作为参数。并将集合中的所有数据写入到文件中，文件名以股票代码为名字。文件的格式为一行数据为一个评论。

6.3.5 评论数据存 HDFS 接口

存入 HDFS 和存入本地系统接口类似，只不过在些数据之前得先传入参数并创建 HDFS 连接，参数内容如表 6-5 所示：

表 6-5 HDFS 连接配置

属性名	属性值	备注
fs. hdfs. impl	org. apache. hadoop. hdfs. DistributedFileSystem	
dfs. client. block. write. replace-datanode-on-failure. policy	NEVER	
dfs. client. block. write. replace-datanode-on-failure. enable	true	

再写入文件之前会先进行一些判断，比如判断该文件是否存在，如果不存在则会先创建文件，然后通过调用 HDFS Api 的 apennd()方法将数据追加进文件中。文件中数据的格式为一条评论一行。文件的命令为股票的代码。

6.3.6 评论数据存 MySQL 接口

该接口需要接收一个集合作为参数，通过 JDBC 的方式连接到数据库，使用 SQL 语句将数据写入到 MySQL 的 Comment 表中。

6.4 大数据平台实时分词

本节对实时分词模块进行了详细说明，其中包括 Kafka 消费者的配置和实时分词的实现。

6.4.1 Kafka 消费者配置

在创建 Kafka 消费者的时候，往往需要一些关于消费者的一些配置，配置如表 6-6 所示：

表 6-6 Kafka Consumer 配置

属性名	属性值	备注
bootstrap. servers	localhost:9092	配置 cluster 中 borker 的 host/port 对

auto. offset. reset	latest	消费策略
enable. auto. commit	true	是否 commit 消息
key. deserializer	classOf[StringDeserializer]	配置反序列化类名
value. deserializer	classOf[StringDeserializer]	配置反序列化类名
group. id	test01	消费组

因为 SparkStreaming 可以同时消费多个主题的数据，所以主题的设置是在创建 Kafka 消费组的时候，传入装有一些主题名的数组。

6.4.2 实时分词

拿到 Kafka 的数据后，在 SparkStreaming 中，这个数据是以 RDD 的形式存在的，所以调用一个 map 函数，将每一个 RDD，也就是每一条 Kafka 消息中的值取出来，并通过事先定好的规则对值进行切分。取出来之后调用 foreachRDD() 函数，这样就能对单条消息进行各种处理了，首先拿出消息，因为一条消息是一个 Comment 对象的字符串表达形式，所以要取出 Comment 对象中的评论数据。对评论数据进行分句，同时记录下句子的编号。这时评论就是以句子的形式存在，通过调用 Ansj 分词接口，将句子传入其中。返回的时一系列的词对象。拿到词对象后，对词进行顺序封装，最后装入到一个集合中，传给 MYSQL 接口对数据进行存储。

6.5 词汇数据清洗

本节对数据清洗模块的实现做了详细说明，包括词汇模块过滤和互信息过滤。

6.5.1 词汇模板过滤

通过 SQL 查询对已经分词的数据进行模板统计，SQL 见附录。统计的思路为：先通过分词表拿出想要的股票数据，通过评论 ID, 句子 ID 和词 ID 进行排序，然后对评论 ID，句子 ID 进行合并，合并之后将词性进行合并，这样就得出了每一个句子的词性模板，最后通过去 count() 得出每个模板的个数，使用 count 的值从大到小进行排序，取出前 8 条。这里之所以只取前 8 条呢，是因为后面的数据出现的次数特别小，并不能体现出对股票评论的真实性。最后统计结果如表 6-7 所示：

表 6-7 词性模板统计表

词性模板	示例	出现次数
n, v	建议关注	165
t, v	近期介入	119
v, p, vg	出局为宜	117
v	持有	116

v, v	买入为主	88
d, v, v, n	仍有下降空间	83
vn, p, vg	观望为宜	63
n, n, v	均线多头发散	55

6.5.2 互信息过滤

通过词性模板筛选过后的数据中，有一些词虽然出现的次数较多，但并不是与股票相关，所以这里运用了互信息的方法对词汇进行一轮筛选，以提高过滤的准确率。首先从股票词汇中手动筛选出六个股票的典型属性，筛选出来的词分别是压力线、支撑线、突破、回档、均线、回调。将这六个词作为种子词，用词分别和每个种子词组合在百度查出的页数除以各自在百度查出的页数的乘积。得出的值求对数，在将几个六个值求和。就得出这个词的 PMI-IR 值。公式如下：

$$\mathrm{PMI}-\mathrm{IR}(\omega 1)=\sum_{\omega \in seeds} \ln \frac{hits(\omega 1\&\omega)}{hits(\omega 1)hits(\omega)} \qquad (1)$$

公式（1）中：seeds 为种子词汇，$hits(\omega 1)$和$hits(\omega)$分别为种子词和股票词通过百度搜索得出的页面数，$hits(\omega 1\&\omega)$为种子词和股票词组合在百度中搜索得出的页面数。最后得出的 PMI-IR 值如果越高，则表示这个词和股票的相关性越高。这里需要设定一个阈值，就是当大于某个值的时候是相关的，否者就不相关。经过标准词测试得出当阈值等于-18 时。可以达到最好的筛选效果。

筛选的过程中可能会出现多个相同的词进行筛选，如果将词全部拿去进行筛选的话，会造成系统处理时间长，网络 IO 资源占用多等问题，所以在拿到词集合后，通过 Set 集合的特性，对词进行了一次去重，待词筛选过后，通过词在数据库中进行查找，然后通过 MySQL 接口将词存入 MySQL 中。

6.6 股票打分

当词筛选过后，就根据筛选后的词对股票进行打分，打分的词按层次分为三个层次，分别为微观层面，市场层面和宏观层面。

6.6.1 分层词汇打分

根据多个股票的筛选词汇进行手动分析，发现评论中的一些特征词都是差不多的，所以定制了一个层面词汇模板。宏观层面的词有：意义、曙光、形态、平台、走势、大盘、能量、空间。微观层面的词有：特征、横向、趋势、建议、多头、均线。市场层面的词有：成交量、股价、强势、跌势、新股。首先对每个层面的所有词打分。打分的规则为：如果这个词的所属句子中，出现了消极词，则

该词分数得 1 分；如果这个词所属句子中，出现了积极词，则该词分数得 5 分，其他则得 3 分。

$$S(\alpha)=\begin{cases}5，若出现积极词\\1，若出现消极词\\3，\ \ 其他情况\end{cases} \quad （2）$$

6.6.2 各个层面加权平均分

每个层面词汇打分完成之后，会对层面所有词得分数取加权平均，在之前词汇打分中，记录了一个值，就是该词出现得次数，也就代表了这个词被打分分析了多少次。所以可以通过将所有词得分数相加去除以所有词出现得次数得和得到这个加权平均分。

$$\mathrm{Avg(A)}=\frac{\sum_{\alpha\in A}S(\alpha)}{\sum_{\alpha\in A}C(\alpha)} \quad （3）$$

公式（3）中，A 代表某一个层面，α代表层面中得某一个词，S(α)代表某一个词的分数，C(α)代表某一个词的出现次数。Avg(A)则代表层面的加权平均分。

6.7 结果分析

6.7.1 股票评价

从各个层面对股票进行分析，拿出股票的得分信息，通过对股票各个层面的份进行排序然后取出前十名。得出的结果如表 6-8、6-9、6-10 所示：

表 6-8 市场层面股票 TOP10

股票代码	市场得分	股票名
603929	5	亚翔集成
002789	5	建艺集团
002819	5	东方中科
603859	5	能科股份
603777	5	来伊份
603126	3.76923	中材节能
603776	3.57143	永安行
603883	3.51852	老百姓
002717	3.44444	岭南股份
002771	3.41667	真视通

表 6-9 宏观层面股票 TOP10

股票代码	宏观得分	股票名
002649	2.87726	博彦科技
300277	2.83333	海联讯
600716	2.81227	凤凰股份

000150	2.80933	宜华健康
600208	2.79964	新湖中宝
000918	2.79882	嘉凯城
002146	2.79688	荣盛发展
600215	2.78967	长春经开
600645	2.78636	中源协和
002077	2.77897	大港股份

表 6-10 微观层面股票 TOP10

股票代码	微观得分	股票名
603776	4	永安行
603883	3.7	老百姓
603126	3.52632	中材节能
603777	3.52632	来伊份
002717	3.44586	岭南股份
300404	3.42857	博济医药
002819	3.39216	东方中科
002771	3.23881	真视通
603929	3.21212	亚翔集成
002789	3.17857	建艺集团

从上面三个表的数据可以看出，在市场层面，亚翔集成、建艺集团、东方中科、能科股份、来伊份五支股票的评分最高，按打分的规则来说是满分，所以市场层面，这几支股票的表现是很好的。宏观层面得分都差不多，说明这个层面不是很好做出评价。在微观层面，永安行和老百姓两支股票的分数上很高，表现较好。

上面分析了各个层面的情况，也可以将所有层面的分数放在一起进行分析，下面两个表分别是：在各个层面都进入前 20 的股票（表 6-11）；算出所有层面分数的平均数，然后取出分数排名前十的股票（表 6-12）。

表 6-11 各层面进入前 20

股票代码	宏观得分	微观得分	市场得分	股票名
2398	2.74129	2.9767	3.04255	建研集团
300288	2.71533	3.17476	3.3	朗玛信息

表 6-12 得分加权平均 TOP10

股票代码	平均分	股票名
603777	3.678583333	来伊份
603929	3.56251	亚翔集成

2819	3.559943333	东方中科
2789	3.416666667	建艺集团
603859	3.392416667	能科股份
603776	3.32773	永安行
603883	3.287893333	老百姓
603126	3.286256667	中材节能
2717	3.197076667	岭南股份
300288	3.063363333	朗玛信息

通过表 6-11 的数据可以看出，建研集团和朗玛信息两支股票的评分在三个层面都进了前 20，可以看出这两只股票在用户的反馈中各个方面是相对都比较好的。从 6-12 可以看出来伊份、亚翔集成、东方中科和建艺集团等股票的分数较高，在用户的反馈中总体表现较好。

6.7.2 行业评价

将所有股票的所有层面的分数求和算出平局分后，可以分析股票行业在前十中所占的比例。通过比例也可以对相应的行业做出评价。下面表 6-13 是股票得分排行 TOP10 以及其行业。

表 6-13 股票得分排行 TOP10

股票代码	平均分	股票名	行业名称
603777	3.678583	来伊份	批发和零售业
603929	3.56251	亚翔集成	建筑业
2819	3.559943	东方中科	批发和零售业
2789	3.416667	建艺集团	建筑业
603859	3.392417	能科股份	科学技术服务业
603776	3.32773	永安行	科学技术服务业
603883	3.287893	老百姓	批发和零售业
603126	3.286257	中材节能	科学技术服务业
2717	3.197077	岭南股份	建筑业
300288	3.063363	朗玛信息	信息技术业

从上表中可以分析出，在前十的股票中，科学服务行业，建筑业和批发零售业相同，都有 3 家。可以看出这三个行业在用户的评论中是比较好的。

结 论

本文主要研究在大数据平台中，对股票评论信息的文本挖掘。以股票论坛等网站上的评论信息为原数据，通过爬取工具对网站信息进行爬取，通过消息系统和流处理框架的整合，实现了在大数据平台中的文本实时分词。利用词性模板和互信息的方式对分词数据进行筛选，最后通过定制的规则对股票数据进行打分。

在打分数据中，从各个层面来看，在市场层面亚翔集成、建艺集团、东方中科、能科股份、来伊份五支股票在用户的反馈中表现较好。在宏观层面，各个股票的评分都差不多。在微观层面，永安行和老百姓两支股票表现较好。从总体来看，来伊份、亚翔集成、东方中科和建艺集团等股票的表现较好。从行业开看，科学服务行业，建筑业和批发零售业表现较好。

本文中有些处理也不是很完善，在实时分词后面进行词汇筛选，有可能会遗漏一些数据。造成这个问题的原因是我们无法监控流处理的进度以及执行是否完成。后续可以考虑对该功能进行修改和完善。

附 录

附录 1：词性模板过滤 SQL

```
    select
      b. comment_id, b. sentence_id, count（1） as
`num`, b. sentence_nature, GROUP_CONCAT（″-″, b. sentence）
      from
      （select
      GROUP_CONCAT（a. nature） as
`sentence_nature` , GROUP_CONCAT（a. stock_term） as
`sentence`, a. comment_id, a. sentence_id
      from
      （SELECT
      comment_id, sentence_id, nature, stock_term, stock_code
      from stock_term s
      where s. stock_code = ″+股票代码+″
      ORDER BY s. comment_id, s. sentence_id, s. stock_term_id ） as `a`
      GROUP BY a. comment_id, a. sentence_id） as `b`
      GROUP BY b. sentence_nature HAVING `num`>70 ORDER BY num desc；
```

参考文献

[1] 韩春，田大钢. 对股票市场信息的文本挖掘[J]. 中国高新技术企业，2008，5(3):66-75

[2] 冯玉才. 关联规则的增量式更新算法[J]. 软件学报，1998，4(4)：15-17

[3] 邵玮，欧宜鹏，丁逸峰，杨婷婷. 大数据技术的发展现状和应用前景[J]. 科学与财富，2014，3(9)：14-14

[4] 白雪，熊昊. 证券行业文本挖掘技术应用现状与探讨[R]. 上海证券交易所发展研发中心.

[5] Lawrence, Steve;C. Lee Giles （1999）.″Accessibility of information on the web″[M]. Nature 400（6740）:107. doi:10. 1038/21987.

[6] 邓彩凤. 中文文本分类中互信息特征选择方法研究[D]. 西南大学，2011：10-15

[7] 范雪莉，冯海泓，原猛. 基于互信息的主成分分析特征选择算法[J]. 控制与决策，2013，(9):8-8

[8] Tom While. Hadoop:The Definitive Guide[M]. 北京：清华大学出版社, 2017

[9] J Kreps, N Narkhede, J Rao. Kafka: A Distributed Messaging System for Log Processing[J]. In Proceedings of the NetDB, 2011, 2(2), 23-24

[10] Hunt, Patrick. ZooKeeper: Wait-free Coordination for Internet-scale Systems[J]. USENIX Annual Technical Conference, 2010, 8(8), 12-13

[11] Holden Karau, Andy Konwinski, Patrick Wendell, Matei Zaharia. Spark快速大数据分析[M]. 2015

[12] Holden Karau, Andy Konwinski, Patrick Wendell, Matei Zaharia. Learning Spark[J]. O'Reilly Media, 2015, 3(3), 8-9

[13] 汪文佳. 基于数据挖掘技术的股市定价模型[D]. 湖南大学，2013：4-5

[14] P Zikopoulos, C Eaton, D DeRoos, T Deutch and G Lapis, Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data[J]. McGraw-Hill Osborne Media, 2011, 2(2), 5-6

[15] 许郡明. Apache Kafka源码解析[M]. 电子工业出版社，2017

[16] 马兆才. 文本分类中的两阶段特征降维[D]. 甘肃科技，2014：11-16

致 谢

在本文即将结束之际，我要首先感谢我的导师王伟钧副教授。在我的学业和毕业设计中对我的精心指导，不辞辛劳地帮助我。在此，我要向我的导师致以最崇高的敬意和深深的感谢。

同时也感谢大学四年里曾经教授过我知识，指导帮助过我的老师，你们的关怀和教诲深深地影响着我，不断激励着我。

感谢所有大学里所有的同学，与他们交流开阔了我思维，丰富了我的知识，在此，对他们表示诚挚的谢意。最后，也衷心感谢那些曾经帮助过我的人，是你们的帮助让我得到了成长。