学校代码：10272

学　　号：2018213606

上海财经大学

SHANGHAI UNIVERSITY OF FINANCE AND ECONOMICS

工商管理硕士学位论文

MBA DISSERTATION

论 文 题 目　基于自然语言处理与深度学习的网络舆情对股票影响的研究

培养院(系、所)　商学院

学位论文类型　工商管理硕士论文

论文作者姓名　褚剑峰

指　导　教　师　陈艳

上海财经大学

2020 年/8 月/16 日

学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品成果。对本人的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

论文作者签名：杨剑峰　　日期：　2021 年 1 月 13 日

学位论文版权使用授权书

（硕士学位论文用）

本人完全了解上海财经大学关于收集、保存、使用学位论文的规定，即：按照有关要求提交学位论文的印刷本和电子版本。上海财经大学有权保留并向国家有关部门或机构送交本论文的复印件和扫描件，允许论文被查阅和借阅。本人授权上海财经大学可以将学位论文的全部或部分内容编入有关数据库进行检索和传播，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

论文作者签名：杨剑峰　　导师签名：

日期：　2021 年 1 月 13 日　　日期：2021 年 1 月 13 日

基于自然语言处理与深度学习的网络舆情对股票影响的研究

摘要

随着网络技术的发展，博客、微博、贴吧、论坛等自媒体的媒介手段越来越流行，这些信息渗透到了生活的方方面面，用户可以自由在网络上发声，对于感兴趣的内容进行评论、点赞、转发等操作。因而产生了如雪球、集思录、淘股吧等专业的对于股票、债券、投资等进行专业讨论的博客、贴吧等网站及手机 APP。聚集了一批股民，这些人会在股吧、论坛、社区获取上市公司相关的信息，同时对于这些信息进行主观性地评论。大量股民，会浏览这些帖子及评论，对于一条公司的消息报道，当大量用户呈悲观性的评论时，用户便会受到情绪影响，抛售股票，而当大量用户对股票一片叫好时，用户便又会加仓买入，这些网络舆情信息，极大地影响着股民的正常投资决策。这导致了在互联网技术日益发达的今天，用户能够从各种渠道获取公司信息和评论，从而进行股票买卖的决策，而不是根据公司经营基本概况，如主营业务情况、市场环境、公司未来增长等因素。

本文主要研究和探讨上市公司的网络舆情对于股价以及波动率的影响。从帖子和评论量、单位时间新增帖子数、单位时间新增回帖数、用户对于公司情感指标、话题热度、KOL 影响力、股市总体热度等多个角度，通过支持向量机、贝叶斯分类等计量方法以及 RNN、LSTM 等深度学习方法，分析舆情信息对公司股价以及波动率的影响。根据以往这方面的研究分析，正面舆情的传播量对于股票的股价和交易量影响较少，而负面舆情的情况则刚好相反，传播越广，股价下降越大，成交量也越大。但在以往的研究中，只是简单的根据帖子的评论量、转发量这些简单的指标进行分析，并没有对舆情中具体指标以及舆情中情感度指标进行量化分析。因而本文根据股民在雪球、股吧等渠道中每日对于公司的评论信息中，抽取出情感词，设立公司每日的舆情评分，判断股民对于公司每日的评价情感得分。以及当出现爆炸式话题，带来传播度的迅速扩张或持续关注某个帖子等因素时，网络舆情对于公司股价以及波动率的影响。将通过自然语言处理以及深度学习等方法，进行分析，判断这些指标对于公司股价的影响。

本文通过舆情信息的爬取，并通过局部敏感哈希的算法来过滤重复舆情数据，在获取的有效舆情数据中，提取出了 3 个高阶舆情特征：舆情热度特征、舆情情感特征、影响力因子特征，验证得出这 3 个因子对于股价的真实波动率存在显著相关关系。再使用 LSTM 深度神经网络模型去训练这 3 个特征与股票真实波动率之间的关系，通过拟合可以看出，这几个高阶舆情指标可以很好地去模拟股票真实波动率的走势，误

差率为 3.8%，即股票的涨跌在很大程度上受到舆情因素的影响。因而，论证了股票舆情高阶量化信息和股票波动率之间的关系。

关键词：网络舆情，深度学习，市场效应，自然语言处理

Research on the Impact of Internet Public Opinion on Stocks Based on Natural Language Processing and Deep Learning

ABSTRACT

As the development of network technology, some self media such as blog, microblog, forum have become more and more popular. These information has penetrated into the whole process of people's life. People can speak their opinions freely in the network, meanwhile they comment, like, forward the content which they like. So it appears some professional web forums or mobile applications like xueqiu, jisilu, taoguba. In these places, the shareholders can get together at the network. They will get the company's related information and express their opinions subjectively about these company information at the same time. Large amount of shareholders will browse these posts. To a specific company report, shareholders will get upset and sell their stock when a lot of people express their pessimistic opinion, and they will raise these company stock when large people express their optimistic opinions. These behaviors and these forums' posts impact seriously the shareholders' decision. It leads that as the network technology is becoming more and more developed, shareholders can get the company's information and comments from many sources and make a decision to buy or sell stock according to these information. But they don't do the decision by the company's base condition such as main business, market environment or future growth potential.

The article will focus on how company public opinion will impact the company's stock price and volatility. So I research some aspect such as the amounts of posts and comments, the number of new posts at unit time, the number of reply posts at unit time, the emotional index, the topic enthusiasm, the influence of key opinion leaders, the total hot index of the whole market and so on, use some statistical methods such as support vector machine, naïve Bayes classification and some deep learning methods such as Recurrent Neural Network, Long Short-Term Memory to analyze how public opinion impact the company's stock price and volatility. According to the previous analysis, the broadcast of positive public opinions impact the stock price and volatility slightly, but the negative public opinions are in the contrary. When the information spreads broader and broader, the stock price drops bigger and bigger, and the stock volatility become larger and larger. But the previous research only focus on the simple target such as the number of comments and reply posts, they don't make a specific quantitative analysis about the public opinion's

emotional index. So I will use web crawler to get the company's everyday public opinion information to abstract emotional phrase to calculate company's everyday emotional score, and when there is a explosive topic which brings rapid expansion of the spread and the continuous attention about the certain post or some other factors, the impact of network public opinion. I'll use natural language processing and deep learning methods to analyze which indicators are conducted to determine the company's price and volatility.

In this paper, the public opinion information is crawled, and the repeated public opinion data is filtered through the local sensitive hash algorithm. From the obtained effective public opinion data, three high-level public opinion features are extracted: public opinion heat characteristics, public sentiment features, and influence Factor characteristics. I verify that these three factors have a significant correlation with the true volatility of stock prices. Then I use the LSTM model to train the relationship between these three features and the true stock volatility. It can be seen through fitting that these high-level public sentiment indicators can well simulate the trend of the true stock volatility. The error rate is 3.8%, that is, the rise and fall of stocks are largely affected by public opinion factors. Therefore, the relationship between high-level quantitative information of stock public opinion and stock volatility is demonstrated.

KEYWORDS: "network public opinions", "deep learning", "market effect", "natural language process"

目录

第一章 绪论

第一节 研究背景

本文主要研究的内容包括网络舆情、相似文本处理、自然语言处理、舆情分析、深度学习等，从舆情这个行为金融学的角度，去探究和论证证券市场中网络舆情所代表的投资者情绪背后所蕴含的量化指标对于股票价格、股票波动率等指标的影响状况。

国内很早有学者对舆情进行研究。曾润喜（2009）解释了舆论和舆情两者之间的区别，他认为，舆论是狭义上来讲就是公众的言论，但必须是由多数人的认知、情感、态度等聚集产生的，带有一致性的看法。因而，当持有某观点的人达到一定的量，才可以被称为舆论。而舆情则不同，简单来讲就是人们对外表达自己对于某件事情的看法，人们可以各抒己见，并不需要多数人的认同。因此，舆情可以表现为零散的、非中心化的。网络舆情则是舆情的一种特定类别。顾名思义，它是指人们利用互联网的手段，去表述自己看法的一种途径。人们对于某件事情的看法，可以借助于互联网的手段进行传播，从而形成人们对于该事件的所有认知、情绪、立场的集合。

在证券市场上，Das（2007）研究并证明了投资者的情感指数与股票的大盘指数之间是有关联性的。Halil Kiymaz（2002）认为，股票的部分消息传闻能够在传统媒体公开前 4 天对股价产生了正向影响，而在媒体公布后则会产生反向的影响。Johan（2011）对 Twitter 内的社交数据进行了数据挖掘，采用行为经济学的方法进行分析，检验了公众情绪的变化与道琼斯工业指数的收盘值的日变化之间的关系，通过格兰杰检验的方式，证明了这两者之间在统计意义上的相关性是显著的。

近些年来，在自然语言处理领域，文本倾向性分析已经成为了研究的热点问题。其主要的研究任务是分析文本中包括的观点和情感信息，有如下 3 个子分类的任务：1）意见分类；2）基于特征的观点挖掘和摘要；3）比较性句子和比较关系挖掘。

随着文本倾向性分析研究的逐步深入，国内外的研究学者们发现：

1）DSL（domain specific language）领域特定语言和 DSK（domain specific knowledge）领域特定知识，这些内容知识对于研究者们的分析研究工作有着很大的作用。如构建领域专用观点情感词词表，以及跨领域融合的文本倾向性分类等。

2）上下文关系、语境（Context）对倾向性判别起到非常重要的作用。所以，研究者们开始考虑使用计算机编程语言等自动化或者半自动化的方式对互联网上的评论信息进行分析。关于相关领域的研究非常活跃，既有词的层次相关的研究，也有句子、篇章层次关系的研究，既有基于词典的研究方法，也有基于语料库的研究。这些都对后续的研究者们所进行的实际研究，起到了很好的借鉴作用。

文本相似度是指两个文本内容或者句子之间的相似程度，这种技术在搜索引擎、命名实体识别、内容推荐、机器翻译、自动应答等领域，都有着广泛的应用。文本距离衡量的是两个文本之间的距离，即文本和文本之间的相似程度。从文本距离和文本相似度的定义可知，文本距离和文本相似度是存在负相关关系的。通俗来讲，两篇文章的文本距离小，表明两篇文章离得近，那么则文本相似度高；而两篇文章的文本距离大，表明两篇文本离得远，则文本相似度低。自然语言处理过程中，经常会涉及到度量两个文本之间相似性的问题，因为文本可以认为是一种有语义词汇组成的高维空间，那么如何对其进行抽象分解、降维等操作，使得能够量化文本的相似性，就是研究者们所要研究的问题。目前有几种文本之间相似性度量的方式，如利用划分法的K-means、基于密度的 DBSCAN、基于模型的概率方法进行文本之间的聚类分析等。同时，也可以利用文本之间的相似性对大规模语料进行去重预处理，或者找寻某一实体名称的相关名称（模糊匹配）。另外，衡量两个字符串的相似性也有很多种方法，如最直接的利用 hashcode，以及经典的主题模型或者利用词向量将文本抽象为向量表示，并在此基础上通过特征向量之间的距离度量，如欧式距离、皮尔森距离等度量指标进行度量。

深度学习模型是一种神经网络模型，通过多层的神经网络的组合，来解决单层神经网络或者经典机器学习模式中所难以解决的问题。Geoffrey Everest Hinton 和 Ruslan Salakhutdinov（2007）提出了一种在前馈神经网络中进行有效训练分析的算法。此算法通过将网络中的每一层看作是一个无监督的受限玻尔兹曼机，然后，再使用有监督的反向传播算法对模型进行了调优。近年来，深度学习模型的提出，为股票相关指标如股票价格和股票波动率的预测提供了新的基础理论和研究方法。孔翔宇等（2016）使用了自动文本分析和机器学习的模型，利用概率主题模型对新闻进行聚类，得到其文本的主题分布，并且结合了股票市场中的交易数据，使用基于支持向量机的机器学习方法，对股票的未来的价格走势情况进行预测。杨楠（2016）通过对神经网络模型进行深入、细致的分析，通过分析了模型本身、泛化能力、应用能力等方面，以及分别分析了组合神经网络、BP 神经网络、模糊神经网络这三种常用的神经网络模型，探索了神经网络这一人工智能技术在我国股票证券市场的预测这一方向的使用前景。

第二节 研究意义

之前研究者们对文本倾向性相关的研究大多集中在电商商品、电影评分等领域，如研究电商网站商品的评论信息或者微博等社交平台中的评论信息对于商品购买的指导作用或者抑制作用，采用统计学以及人工智能的相关理论来尝试解决问题。对舆情指标信息在证券市场上的分析研究工作比较少。因而，本文采用深度学习的相关算法，通过分析对于上市公司的网络舆情信息、舆情传播度、舆情影响因子等相关指标，

在经典的公司估值的研究理论基础上，通过研究网络舆情对于公司股价的影响，从而对行为金融学理论进行有效的补充。

同时，我国正在大力开放金融市场，保持金融市场稳定、持续、高效的发展。在此时候，我国正在大力支持大数据、AI 等技术在各行各业的应用，大数据和 AI 相关技术的研究是一个非常有理论研究价值和和实际应用价值的场景。所以，本文探索自然语言处理、文本相似度模型筛选、深度神经网络、网络爬虫等技术在金融、证券领域的应用。

第三节 研究思路

本文主要采用一套基于自然语言处理和深度神经网络的理论模型，和基于 python 编程语言相关类库的工程实践相结合，研究舆情信息对于股票的股价和波动率的影响情况。在一般情况下，人们普遍认为舆情信息对于股价肯定是有影响的，但是具体的影响程度是怎样的呢，对哪些行业的影响程度比较高，哪些行业的影响程度比较低，哪些舆情信息对于股票的影响比较大，哪些舆情信息对于股票的影响比较小，这些问题并没有一个完整的解释，本文对于舆情相关的影响因子进行定量的研究，探究具体的影响因子以及舆情的相关参数对于股票价格、股票波动率的影响状况。

研究思路为以实际舆情数据为依托，基于数据所展现出来的高阶特性如情感得分、扩散度、KOL 影响因子等统计数据，探究这些信息对于股价和波动率的影响状态，因为股票信息的随机波动性和不确定性，所以采用神经网络相关算法如 LSTM 进行数据建模，因为 LSTM 模型对于时序数据的一些分析特性，非常适合进行时序数据的分析，可以用于进行后续股票数据的预测分析。所以，使用此深度学习的模型来进行舆情信息对于股票价格、股票波动率的相关性的研究。

第四节 研究方法与创新

一、研究方法

采用定量研究的方法，采用的方法如下：

首先，通过网络爬虫爬取对应网站的数据，以此获取公开数据，并结合数据库中收集的数据，包括如雪球、东方财富股吧等论坛中对应各个公司股票的发帖、回帖数据。具体可以爬取到的数据字段有：发帖人论坛 ID、发帖人用户昵称、发帖人头像链接地址、发表帖子时间、发表帖子内容、回帖人 ID、回帖人用户昵称、回帖人头像链接地址、回复帖子时间、回复帖子内容、帖子浏览次数、帖子回复次数、帖子点赞次数等上市企业的舆情信息。

然后，对于收集的数据，进行数据预处理的操作，如数据清洗（字段格式处理、帖子、回帖内容进行分词以及词性判断，情感指标计算）、数据转换（将网页中半结构化内容转换为能够让 mysql 结构化数据库中进行存储的数据格式字段）、数据存储（对于转换好的数据字段格式，存储入 mysql 对应表结构中，对于明细数据，进行统计整理到统计表中）等操作流程。

针对数据清洗阶段，利用文本相似度模型，去除每日重复或者相似的灌水帖子、重复贴子信息，维持数据的精确度。本文所采用的文本相似度处理方案，是基于谷歌提出的simHash（局部敏感哈希）的方法进行处理，来自于谷歌的Moses Charikar（2008）发表的论文“detecting near-duplicates for web crawling”中提出此算法，专门用于解决互联网上海量的网页内容信息的去重任务。局部敏感哈希算法所采用的原理的主要思想是降维，通过将高维的包含语义信息的特征向量，通过一定的哈希算法进行降维，最终映射成低维的特征向量，之后，通过比较这两个特征向量之间的汉明距离来确定文章是否存在重复或者高度近似的现象。

对数据预处理完后，对于清洗完的数据，进行具体的数理统计分析处理，进行回归分析，获取到和股票真实波动率相关性大的高阶情感指标，用于后续基于深度神经网络 LSTM（长短时记忆网络）的模型训练和参数调整。

最后，基于 LSTM 模型，设定训练集、定义输入层和输出层、设置激活函数、设置训练次数，最后进行图像拟合和误差分析。

二、主要创新点

本文的创新点，主要体现在如下 3 个方面：

1）在舆情信息预处理过程中，采用基于谷歌的 simHash 的算法，能够对于海量文本进行文本相似度分析，快速筛选出重复信息进行数据过滤。相比于基于余弦相似度或者其他的基于分类思想的文本相似度算法，simHash 算法用降维思想对文本进行处理，可以快速进行相似文本的对比工作。

2）采用自然语言处理的方法，对于上市公司的论坛帖子、回帖信息进行极性判断和量化情感得分，可以将公司每日舆情所代表的所有用户的情感进行量化处理，看出论坛用户、股民对于公司未来的发展在舆情方面的得分，到底是在增加还是在下降，也即用户所讨论的内容，表明他们未来是偏向于更加看好公司，还是他们对于公司的未来发展偏向于悲观。

3）采用基于深度神经网络的 LSTM 模型，来预测舆情相关的指标对于公司股票真实波动率的影响关系。LSTM 模型，主要是为了解决 RNN 模型中，在训练基于时间序列数据的过程中，可能出现的梯度消失、梯度爆炸等问题，所应运而生的。LSTM 模型相比于传统的 RNN（循环神经网络）模型，其在长时间序列数据的训练过程中，往往能够起到更好的作用。LSTM 模型是通过门控状态来控制传输状态的，它就如人

一样，可以记住需要长时间记忆的信息，并且忽略不重要的信息。而且，它不像传统的 RNN 模型那样仅有一种记忆的叠加方式，所以 LSTM 模型对很多需要长期记忆的任务来说尤其好用。因而，采用 LSTM 模型，将舆情指标作为输入变量，训练模型，预测股票真实波动率变化情况。

第五节 论文的组织结构

本文共分五章，论文的具体组织方式如下：

第一章阐述本文的研究背景、研究意义、研究思路、研究方法和主要创新点等，最后一小节是本文的主要内容和整体组织结构。

第二章主要对网络舆情、文本情感分析、文本相似度模型、海量数据查重、深度神经网络的相关文献进行阐述，从理论角度去介绍相关理论研究成果，并阐述方法的理论可行性。

第三章主要阐述本文所采用的模型的选择、模型指标的选取以及模型数据来源的筛选、相似文本去重、抽取、转换、加载等具体操作。

第四章主要进行实证分析，对几种高阶舆情指标和股票真实波动率的关系的相关性进行分析，将相关性高的舆情指标用于基于深度神经网络的公司股票真实波动率的预测模型中进行计算。最后，对于本文所采取的方法的效果以及对于不足点进行分析，并且提出后续的优化点。

第五章总结了本文的研究工作，并对未来进行了展望。

第二章 文献综述

第一节 网络舆情文献综述

一、网络舆情研究成果

曾昭皓和李卫东（2013）对网络舆情下了定义，指出网络舆情是在互联网上进行传播的，人们对于社会问题的不同的看法、见解、认识、思考等，可以作为社会舆论的一种特定的表现形式。它的特点是，通过互联网传播。在互联网技术日益发达的当下，公众往往会通过互联网相关技术，如：微博、微信、社区、论坛等，对现实生活中所发生的某些热门、焦点事件，表达出基于自己看法的倾向性的言论和观点。网络舆情的核心是事件，载体是互联网，整体上表现为网民们情感、观点、态度、意见、表达、传播、互动等相关表现所集成的集合。网络舆情的表达方式和传统舆情产生方式相比，呈现出多样、快捷、信息多元、互动等特点。正是因为互联网的去中心化的特点，决定了用户的特点。张蕾（2018）指出，网络舆情具有以下特点：

1）直接性：用户可以随手发表微博、在论坛发帖等，或者通过 BBS（电子公告板）等其他手段。可以随时随地在接入互联网的地方立即发表意见，情绪能够直接传达，表达方式也更加畅通。网络舆情具备了即时、快速传播的特点。在互联网上，用户只要将网络舆情信息复制粘贴或者一键转发至其他平台，这些信息就会在其他媒体、平台上面得到重新的传播。网络的这种特性使得网络舆情的传播成本极低，并且很难溯源。

2）随意性和多元化：如用户可以通过火星文、颜文字、网络流行语等表达情绪，这是因为用户所身处的网络社会，具有无边界性、虚拟性、匿名性、即时交互等特性。从而，使得身处网络社会中的人们所表达观点的方式自然也伴随着网络社会的特点。另外，网络社会上各种文化、思想层出不穷，因而出现了各种不同舆论、思想、道德规范、价值观和思潮。每一种不同的思想，因为互联网的放大效应，导致都会聚集一批彼此相互认同的群体。而这些思想中，有健康的、积极向上的舆论，同时也有灰色的、庸俗低级的舆论。可以说，网络舆论丰富多彩、内容庞杂、五花八门、分门别类。同时，因为在网络上发表言论的代价很小，所以网民可以刻意隐匿自己的身份，交流思想，沟通信息。知乎上人均年收入百万、人均硕士学历的帖子层出不穷，就是这一现象的真实写照。网络既为网民提供了畅所欲言、宣泄情绪的空间，也为监管机构提供了搜集真实舆情信息的素材。

3）突发性：互联网的出现，拉近了世界的距离，在世界上任何角落发生的重大新闻，只要有互联网的存在，都会在事件发生之后极短的时间内，迅速在互联网上以

呈现指数增长的传播速度立刻传播开去，在网络上成为关注的焦点。在获得极大的关注度时，自然也成为人们街头巷尾热议的话题，迅速成为舆论热点。在当今社会，传统媒体的作用逐渐变小，他们在互联网时代，对于突发事件往往采用快速获取消息，立刻跟进，通过官方渠道在网络上发布信息，形成网络热门舆论的方式。同时，因为网络舆情可以以最快的速度传播的特性，导致了自媒体的兴起。使得传统媒体并不一定能够得到第一手现场素材，某些事件用户可以亲眼见证，获取第一手材料，发表到自媒体平台、微博、社区等相关平台，使得由传统媒体主导舆论的方式有所削弱。

4）实时交互性：如发帖、回帖、弹幕文化等。使得人们不管远在地球任意角落，只要接入了互联网，就可以对于舆情信息进行实时互动。B 站弹幕可以成为聊天的地方，论坛帖子盖楼、灌水等成为文化。舆情信息就在这种实时交互的过程中快速产生和传播着。

李守明（2014）指出，近几年我国正在通过使用基于海量数据处理和人工智能的技术，对海量网络舆情信息进行分析和深度挖掘的工作。同时也采用实时计算的技术，达到可以快速汇总舆情信息，代替人工处理、分析的繁复工作的目的。

二、网络舆情研究指标

曾润喜等（2014）指出，对舆情主要研究的指标包括时间维度、数量维度、显著维度、集中维度和意见维度等维度。然后，通过统计计量学的相关方法，对这些指标进行观测，就可以衡量网络舆情的稳定性、舆情的强度以及受众意向的倾向性的大致分布。

基于这些基础指标，应对于不同的场景，又产生了各种应对不同场景的舆情指标研究体系。如：1）基于社会预警研究启示的指标体系；2）基于主题分类的指标体系；3）基于舆情不同发生主体的指标体系；4）基于网络舆情内在机理的指标监测系统。

基于云计算的发展，在云服务上大量的算力能够被同时使用，使得人工智能技术不断得到推进和发展，同时也推进了网络舆情分析的发展，出现了多种不同的针对舆情的量化分析方法，如：1）德尔菲法和层次分析法结合的指标权重确定方法；2）基于多级模糊模型概括评判的网络舆论安全评估模型；3）基于 BP 人工神经网络的方法；4）网络舆情威胁评估模型；5）基于 Elman 神经网络的网络舆论威胁估量方式；6）基于灰色系统理论的网络舆论预警评价方式。

随着网络技术的进步以及理论研究的深入，这些最新研究的指标和方法一起推进着网络舆情研究的新方法、新方向。

三、小结

回顾了国内外网络舆情的研究成果，我们可以得出如下结论：

1）网络舆情因为互联网技术的迅速发展，也在迅速发展开来，人们每天会在网络上产生大量的数据，这些数据需要得到相关监管部门的重视。

2）网络舆情因为互联网本身快速、实时、多元等特点，使得用传统方法去评估舆情变得非常复杂，需要借助于计算机、云计算、机器学习等方法来获取、量化、评估，网络舆情问题变成了一个需要跨领域、跨专业从多个角度一起去解决的学术问题。

3）在已有的研究中，研究者们对于网络舆情的量化指标进行了总结，归纳了具体指标的使用场景和方法。这为后续基于计算机技术和机器学习技术进行网络舆情量化数据分析的工作提供了完善的理论支持。

第二节 文本情感分析文献综述

马力和宫玉龙（2014）介绍文本情感分析研究的主要工作，包括了特征次抽取、观点情感特征词抽取、观点词极性判断等方面的工作。

一、特征词抽取

情感特征词的抽取是指在文本信息中抽取包含情感信息的特征。在抽取过程中，往往要判断这次单词或短语在语句的情感表达中所扮演的角色。主要的任务包括识别情感表达者、识别评价对象、情感观点词识别等。这其中，评价对象和情感词抽取在情感挖掘领域中占有重要的作用。抽取完评价对象、情感词之后，所需要做的工作，就是构建领域相关的主题词、情感词的词表和词库信息。

在抽取评价对象时，通过将其限定在名词或基于名词的短语，使用基于规则或模版的方式进行抽取。这其中，包括了一系列的语义分析、预处理等操作，如命名实体识别、词性标注、句法分析等。

通过对大量的包含情感信息的文本进行分析，抽取出有价值的情感信息，对于上层任务如情感检索、情感分类等，都有很大的帮助。面向中文证券股票论坛的评论中的评价对象-情感词对的抽取，和比较成熟的研究领域如商品评论信息、电影评论信息等的抽取工作相比，会显得更加艰难，主要有如下几个原因：1）评价对象数量繁多，组成形式更加复杂；2）情感词在句中所代表语法成分更加灵活；3）虚指评价对象更加常见，较难识别真正指代对象；4）隐式评价对象在句中的使用更加频繁，很难找出真实评价对象。

江腾蛟等（2017）研究了常见的基于语义分析的评价对象-情感词对抽取的方法，指出有如下三种抽取方法：

1）基于最近距离的方法：研究发现，评价对象和情感词在句子中所在位置通常比较接近，因此在对包含情感的语句进行词性分析后，去除掉对应副词、修饰词等，就可以获取到名词和形容词对，也就是认为其是语句中的评价对象-情感词对。

2）基于机器学习的方法：通过使用如神经网络等机器学习方法，对数据集进行模型训练，提取相关特征。

3）基于句式关联或规则的方法：基于语言学中专家系统的句式关联规则，对语句中进行评价对象和情感词对的抽取。

二、观点词抽取

对于观点词的抽取，是在特征词的研究之后进行开展的。通常认为如：观点词/评价词或者评论对象/属性，可以称为评论标签（属性+评价词），这是规则的评论标签的范畴，对应的还有不规则的标签，目前主要的研究是在论规则的评论标签的提取工作上。所采用的常见的研究方法有无监督学习和有监督学习。

无监督方法包括如根据规则、正则来提取。例如属性一般为名词或名词短语，而评价词一般为形容词或成语。邓崇威（2014）认为，可以把句子进行分词以及词性标注，然后提取出对应的名词和形容词。这种方法的优点是简单并且快速，但是缺点也是是显而易见的，就是精度不高。有可能出现提取的名词和形容词并没有关联关系，也有可能词性标注错了，因为词性标注程序有一定的误差，不可能百分百正确，从而导致被错误召回。比如“苹果开机很慢”用这种方法会召回“苹果”（名词）、“很慢”（形容词），这显然是错误的，因为“苹果”和“很慢”并没有直接关系。原句完整应该是“苹果开机（时间）很慢”，应该召回“开机时间”作为名词、“很慢”作为形容词，因为省略导致最终结果的错误。所以 “时间”和“很慢”是正确的标注。这种省略的语法方式，在中文中是非常常见的，人有先验知识能理解语义，但是计算机却很难懂，这是自然语言处理中面临的一个难题。针对上述方法存在的问题，很自然的引入句法分析来验证句子成分之间的关系。这种方法的流程是构建从分词、词性标注、用规则提取标签、依存分析、评论标签生成、评论标签粒度分析、评论标签归一化的流程。如图所示：

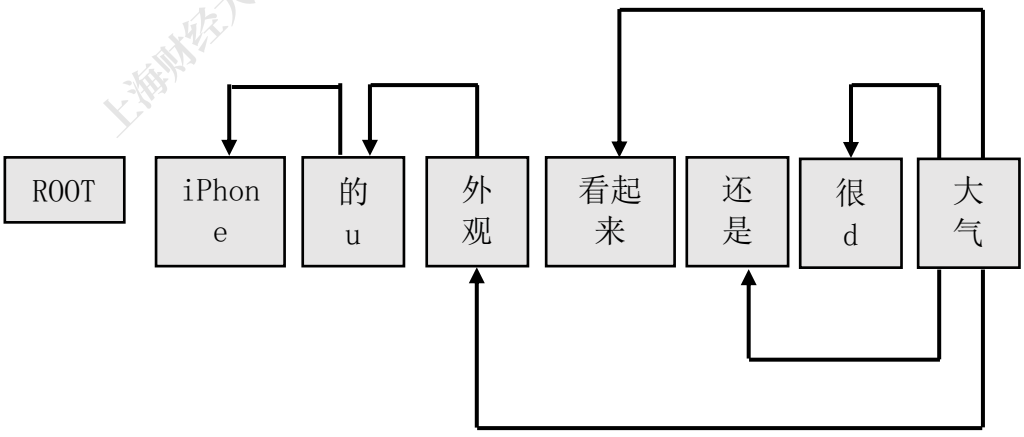

图 2-1 观点词抽取过程

根据规则提取出名词包括：iPhone、外观。形容词包括：大气。“看起来”、“还是”、“很”都是修饰“大气”的，组成 ADV（状中关系），“外观”和“大气”组成 SBV(主谓关系) ，显然句法成分验证没问题，最终评论标签：外观+大气。但是“外观”是一个比较抽象、比较泛泛的属性，例如外观它可以当很多东西的属性，人的外观、手机的外观、汽车的外观等。因此无法很好的用评论标签代表句子。要解决这个方法需要把“iPhone”和“外观”合并，最终的标签为：iPhone 外观+大气，这就是属性的粒度分析。此外可能还需要对评价词-观点词进行归一化，比如：续航+给力、续航+很棒都归一化为续航+很好。以上方法大部分情况下会取得不错的效果，但可能在个别领域效果不太好，原因可能就是属性词很抽象、很模糊。比如“热情高涨”、“情绪低落”等的属性是情感，比较抽象。这时候可以考虑加入一些频繁项集挖掘算法，比如 Apriori 算法，找出频繁词集来当属性。此外还可以针对属性制定一些规则，比如属性词长度、与评价词的距离等。

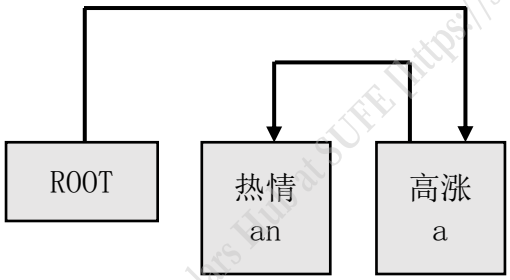

图 2-2 针对属性制定特殊规则

有监督学习的方法在进行任务处理时，会优先考虑适合处理序列标注的方法。比如 HMM、MEMM、CRF 等。

表 2-1 句子词性标注

词	手机	通话	声音	清晰	，	相机	聚焦	脸部	非常	准确
词性	n	n	n	a	w	n	v	n	d	A
标注	N	S	E	N	N	S	N	N	N	N
词/词性	手机/n	通话/n	声音/n	清晰/a	,/w	相机/n	聚焦/n			
标注	N	S	E	N	N	S	N			

如上图所示，对句子进行标注，然后把标注信息当成特征训练模型，就可以对新的数据提取评论标注。理论上监督学习比无监督学习处理大部分任务效果会好一些，代价就是需要的数据量太大，标注的数据太多，非常浪费时间和人力。如果数据有限或标注质量不高，很可能会导致效果还不如非监督学习。除此之外，出现一些未被标注的数据也可能导致效果不佳，而新词是层出不穷的，这一点也制约了监督方法的使用。

王巍（2015）提出了一些其他可行的方法：比如主题模型（LSA、PLSA、LDA），用主题模型挖掘属性词和评价词，最终得出评论的 tag。语义的角色标注，就是在句法依存关系分析的基础上，进行谓语（句子核心、句子可能包含多个谓语）识别、谓语语义（有多个语义，如“打电话”VS“打架”）识别、语义角色识别（成分识别、角色识别等）。最终根据语义角色抽取评论的标注。

也可以考虑采用深度学习的方法，如 RNN、LSTM、GRU 等方法，都是非常适合用来处理序列标注的模型。如果在标注的距离比较远的场景下，LSTM、GRU 等模型的效果会比 RNN 更好。大多数情况下，双向神经网络效果会比单向的好，因为单向的只能学习下文，而双向的可以学习上文和下文，会更加准确。如果数据量很大的话，可以使用 LSTM、数据如果不大的话，可以使用 GRU，数据量不大用 LSTM 有可能出现过拟合的情况。同时，多层双向 LSTM/GRU 的方法，也是一种新的研究方向。

三、观点词极性判断

抽取完观点词后要做的工作是进行观点词的极性判断。刘燕美（2011）在做了关于教育资源评论的倾向性研究之后指出，极性判断的主要的工作是利用机器自动化地提取人们对某人或事物的态度。从机器学习中分类的角度出发，极性判断的问题可以看成是判断包含情感的语料所表征出的正极性（积极情感或乐观情感）、中极性（无情感或中性情感）、负极性（负面情感或悲观情感）。将此问题看作是一个非常特殊的三分类问题，观点词代表一个语句的正负评价，如对于一个商品的评论的观点，是正面的还是负面的，对于一个股票或者一个公司的评价是偏正面还是负面的，以下介绍一些当前对于观点词的极性判断的研究成果。

（一）特征挖掘方法：

1）TF-IDF：TF-IDF 分别表示词频和逆向文本频率，这 2 个指标结合起来可以用来表示一个词语或者短语对于一篇文章的重要程度，例如，一个词语在一个语料中多次出现，而在其他语料中很少出现，那么这个词就能非常好地代表这个预料，或者说能够更好地表征这个语料。IDF 的公式如下：

IDF =log(|D|/DF)　　（2-1）

其中|D|为所有文档数。与 DF 所代表的意思相反，IDF 的值越高，表示此特征 t 对区别文档的意义越大，从通常意义上来讲，表明这个词能够非常好的区分这个语料文档和其他语料文档。最终定义：TF-IDF=TF*IDF 通过 TF-IDF 的特征挖掘方法，就可以更好的对于观点词的极性进行分类。

2）信息增益：高金莉（2010）指出，此方法来源于信息论中的信息熵的概念，主要描述了一个词语或者短语在文本中出现与否，对文本情感分类的信息熵影响。也就是说，一个词语在文本出现前后的这个文本的信息熵的变化。信息增益的公式如下：

$$IG(T) = H(C) - H(C/T) \quad (2\text{-}2)$$

$$= -\sum_i^n P(C_i|t)\log_2 P(C_i) + P(t)\sum_i^n P(C_i|\bar{t})\log_2 P(C_i|\bar{t})$$

其中，n 代表总类别数，$P(C_i)$代表第 i 类出现的概率。P(t)代表出词语 t 的文档数除以总文档数，那么 P($\bar{t}$)= 1 - P(t)。$P(C_i|t)$代表 t 出现时，C_i出现的概率。$P(C_i|\bar{t})$代表 t 不出现但属于C_i的概率，等于未出现 t 但属于C_i的总数除以未出现 t 的所有文档数。

（二）分类算法：

1）朴素贝叶斯算法：它的公式为：

$$P(C|X) = P(X|C)P(C)/P(X) \quad (2\text{-}3)$$

其中，先验概率$P(C)$是通过计算语料中属于每一个类的训练样本所占的比重。而在计算后验概率时，对于一个文档 d，在多项式模型中，只有在 d 中出现过的单词才会参与后验概率计算。朴素贝叶斯的方法是一个经典的利用概率模型进行分类的算法，通过先验概率，可以有效地推导后验概率，从而进行观点词的极性判断。

2）支持向量机模型 SVM：支持向量机对于线性可分的数据，用一个超平面来表示，公式为：

$$f(x) = w * x + b \quad (2\text{-}4)$$

将这两类多维数据分开，之后判断多维平面上的点到这个超平面的距离，从而来进行极性判别。而对于线性不可分的数据，也是可以采用支持向量机的方法的。宋营军（2010）指出其方法是，将线性不可分数据映射到一个更高维度的空间，然后在这个多维空间中，建立并寻找一个满足最大间隔的超平面。映射的关键是采用核函数，周绮凤（2007）在做了基于支持向量机的若干分类问题的研究后，总结了常用的核函数。包括：Sigmoid 核、线性核、径向基核、多项式核、高斯核等。因为观点词是线性不可分的，所以需要采用核函数映射的方式，来进行分类，从而得到观点词的极性。

四、小结

在回顾了文本情感分析的相关文献之后，可以得出如下结论：

1）股票舆情数据的情感分析和情感抽取工作，由于其语料数据的特殊性，往往与电商评论数据、电影影评等数据相比，更加难以处理，一般采用基于机器学习或者深度学习的模型去进行抽取和分析。这样可以保证分析的准确性和召回率指标。

2）基于 TF-IDF 的特征挖掘算法，因为考虑了语料数据的自身特点，如股票数据自身的常用词语标注等特点，可以起到非常好的特征词提取和极性判断的效果，非常适用于股票舆情数据的情感分析的工作。

第三节 文本相似度文献综述

一、NLP 中的相似度量指标

在自然语言处理(Natural Language Processing, NLP)中，经常会涉及到如何度量两个文本的相似度问题。在诸如对话系统和信息检索等的问题中，如何度量句子或者短语之间的相似度尤为重要。

王浩（2013）总结了如下几种相似度度量指标：

1）欧氏距离：衡量的是在一个多维空间中的两个点之间的绝对距离。计算是基于各维度特征的绝对数值，使用欧式距离进行计算的前提是多维空间的维度指标需要进行统一处理，如在 KNN 算法中需要对特征进行归一化处理。公式如下：

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (2\text{-}5)$$

2）曼哈顿距离：此衡量标准也被称为城市街区距离，主要衡量的是两个点在多维坐标系下的各维度的绝对轴距相加的总和。计算公式如下：

$$d = \sum_{i=1}^{n}|x_i - y_i| \quad (2\text{-}6)$$

3）余弦相似度：此衡量标准采用多维向量空间中两个向量在统一坐标下的夹角的余弦值，以此来衡量两个点之间的差异程度。相比前面介绍的欧式距离、曼哈顿距离等距离度量指标，余弦相似度的度量，主要注重的是两个多维向量在方向上的差异，而不是在距离或者长度上的差异。计算公式如下：

$$\cos\theta = \frac{\sum_{i=1}^{n} x_i * y_i}{\sqrt{\sum_{i=1}^{n} {x_i}^2 * \sum_{i=1}^{n} {y_i}^2}} \quad (2\text{-}7)$$

4）其他相似度指标：包括如皮尔森相关系数、一般化闵可夫斯基距离（当 p=1 时即为曼哈顿距离，当 p=2 时即为欧式距离）、汉明距离等。

二、文本相似度度量方法

沈诗嫫（2013）在对文本数据聚类算法的若干技术研究后，总结了度量文本相似度的主要几种方法：

1）基于关键词匹配的传统方法：如 N-gram 相似度。基于此模型的方法，是采用模糊匹配的方式定义的句子（字符串）的多个维度，然后通过比较多个维度之间的差异，来衡量相似度。

2）将文本映射到向量空间，再利用余弦相似度计算的方法：目前主流又三种方法进行表达词意的判断，第一种的代表是 WordNet，它提供了一种词的分类资源，但是缺少了词之间的细微区别，同时也很难计算词之间的相似度。第二种是 Discrete representation，如 One-hot representation，它的向量维度和词典的长度相同，所导致的问题时，此方法的向量维度可能非常高，同时由于向量之间本身是正交的，无法计算词与词之间的相似度；第三种就是 Distributed representation，它的基本思路就是将每个词映射为一个固定长度的短向量，将这些词用于构成词向量空间，将每一个词当作

是空间中的一个点，在这个空间引入距离的概念，根据词之间的距离来判断它们之间的相似性，代表方法如 word2vec、LDA 等。

3）基于深度学习的方法：如 DSSM、ConvNet、Siamese LSTM 等。夏长林（2019）在研究了深度学习在图像中的应用后指出，随着深度学习在各个领域的发展，近些年深度学习开始应用于自然语言处理的不同应用中。语义相似性匹配问题就是一个从人工设计特征转向分布式表达和神经网络结构相结合的方式。

三、小结

回顾了文本相似度计算的相关文献后，可以得出如下结论：

1）文本相似度不同于普通的距离相似度、形状相似度等指标特征，需要进行特定的转化如降维，使之成为常规能够度量的相似度指标。这样在进行自然语言处理的过程中，通过使用文本相似度计算的方法可以去除重复的舆情信息，提高舆情数据处理和整合的准确性。

2）在最初，文本相似度的计算的理论知识逐渐完善，但是受限与计算机硬件条件和算法，往往都是采用离线计算的方法，即对于需要计算的数据，需要跑长时间的计算任务才能得出最后的结果。但是因为股票的实时性的特点，这样的数据往往就没有应有的效果了，变成历史舆情数据，并不能实时去反映股票的实时指标的变化情况。但是随着机器学习技术的发展、硬件算力资源的丰富、云计算技术的发展，使得文本相似度实时计算成为可能，这为实时爬取舆情信息进行过滤预处理，奠定了理论基础和工程实践方法。

第四节 深度神经网络文献综述

一、深度神经网络综述

张蕾和章毅（2016）对深度神经网络做了概述，指出深度神经网络是机器学习的一个分支，是一种以人工神经网络为框架，对数据进行表征学习的算法。而表征学习的目标，是寻求一种可以更好的表示方法，同时创建更好的模型，以此来从大规模未标记数据中学习这些表示方法。与传统的机器学习的方法相比，深度学习以数据为驱动，能按照神经网络的方式，自动从数据中提取特征，对于一些半结构或者非结构化、模式多变或跨领域的大数据的分析，具有非常显著的优势，适用于基于分类的数据应用场景。

深度神经网络归根到底是一种判别模型，可以通过使用反向传播算法进行训练。权重更新可以使用下式进行随机梯度下降法求解：

$$\Delta_{w_{ij}}(t+1) = \Delta_{w_{ij}}(t) + \eta \frac{\partial C}{\partial w_{ij}} \qquad (2\text{-}8)$$

其中，η为学习率，C 为代价函数。这一函数的选择与学习的类型（例如监督学习、无监督学习、增强学习）以及激活函数相关。

周子天（2018）指出，深度神经网络目前是许多人工智能应用的基础。在语音识别、图像识别上取得了突破性的应用，使用深度神经网络的应用量有了爆发式增长。这些应用被部署到了从自动驾驶汽车、癌症检测到复杂游戏等各种应用上。在这些领域中，深度神经网络的判别准备率，能够大大超越人类的准确率。其出众的表现主要源于使用统计学习方法从原始数据中提取高阶特征，从而可以在获取大量数据的同时，能够获得数据在输入空间的有效高维表征，这与之前使用手动提取特征的方式或基于专家系统的规则的方法完全不同。

二、深度神经网络常用模型

冯伟国（2015）指出，深度神经网络的模型有很多，目前开发者最常用的深度学习模型与架构包括 CNN、DBN、RNN、RNTN、自动编码器、GAN 等，以下将对一些常用模型的原理以及适用场景进行介绍。

（一）卷积神经网络原理

卷积神经网络是一种前馈神经网络，具有，其中的人工神经元可以响应一部分覆盖范围之内的周围单元的特点。它在图像处理方面有着非常优秀的表现。它的卷积运算的过程大致为：

1）归一化：包括图像白花处理的变形、减法运算（平均去除、高通滤波器进行滤波处理）、除法运算（局部对比规范化，方差归一化）。

2）滤波器组：维度拓展，映射。

3）非线形：包含稀疏化、饱和、侧抑制、精馏、成分明智收缩、双曲正切等操作。

4）池化：空间或特征类型的聚合、最大化、LP 范数、对数概率等。

王术波（2018）指出，给一个卷积神经网络更多的训练数据，可以做更多的训练迭代，就能实现更多的权重更新，对神经网络进行更好的调参。Facebook 可以使用数以亿计的用户发发表的所有照片，将其应用到自动标准算法中，Pinterest 可以使用在其网站上的 500 亿信息将其应用到主页个性化信息流中，谷歌可以使用搜索数据将其应用到图片搜索中，亚马逊可以使用每天的产品购买数据将其应用到产品推荐服务。

（二）递归神经网络 RNN 及 LSTM 原理

递归神经网络与前馈神经网络不同，它的输入参数不但包含当前所见的输入样例，还包括了网络中上一个时刻所感知到的信息。如图所示：

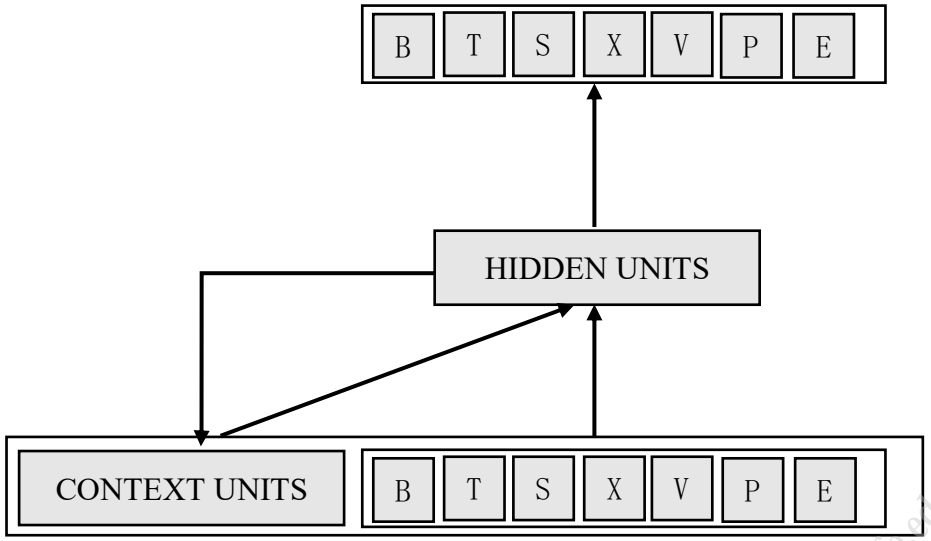

图 2-3 RNN 处理过程

其中，递归网络在第 t-1 时刻的判定，会影响其在随后第 t 个时刻的判定。所以递归网络有来自当下和不久之前时刻的两种输入，这两者的结合，决定了网络对于新数据如何反应。而这些顺序信息，则保存在递归网络隐藏状态，即中间层的神经网络中，可以不断向前层层传递，跨越许多个时刻，影响每个新样例的处理过程。其记忆向前传递的过程公式如下：

$$h_t = \phi(WX_t + Uh_{t-1}) \qquad (2\text{-}9)$$

其中，在第 t 个时刻的隐藏状态表示为h_t，他是同一时刻的输入X_t的函数，由权重矩阵W进行修正，同时，加上前一时间步的隐藏状态h_{t-1}去乘以它自身的隐藏状态。隐藏状态中包含了权重矩阵，它可以决定赋予当前的输入、过去的隐藏状态具体权重的筛选器。所产生的误差会反向传播，进行返回，用来对权重进行调整，直到达到误差允许的范围之内，不能再降低，误差调整结束。

德国研究者 Sepp Hochreiter 和 Juergen Schmidhuber（1995）提出了一种递归神经网络的变体，也就是带有所谓长短记忆单元的网络，被称为 LSTM，它的目的是用来解决梯度消失或者梯度爆炸的问题。LSTM 可以进行误差保留，这样可以用于沿时间或者层进行反向传递。LSTM 可以将误差保持在一个更为恒定的水平，使得能够进行许多个时间步的学习，从而可以建立时间序列中距离较远的数据之间的关联。

RNN 主要使用场景有：文本生成（给出前后文，预测空格中的词）、机器翻译（翻译工作也是典型的序列问题，词的顺序直接影响了翻译的结果）、语音识别（根据输入音频判断对应的文字是什么）、生成图像描述（类似看图说话，给一张图，能够描述出图片中的内容，经常使用 RNN 和 CNN 结果）。

（三）生成对抗网络 GAN

高清红（2018）指出，生成对抗网络是一种生成模型，它在某种意义上避免了如传统概率生成模型一般都需要的进行马尔可夫链式的采样和推断的学习机制，这是它

区别于传统的概率生成模型的特点。GAN 有效地避免了这个计算复杂度很高的过程，使之能够直接进行采样和推断，从而提高了 GAN 的应用效率，所以其实际应用场景也变得更加广泛，如人物头像的融合任务，将 2 个头像中的人物融合入一个中做成人物专属头像等。

GAN 的设计框架显得非常灵活，各种不同类型的损失函数都可以整合到 GAN 的模型中去。使得 GAN 可以针对不同的任务设计不同类型的损失函数，这些参数都会在 GAN 的模型进行学习和自我优化。同时，在传统基于概率的神经网络中，出现当概率密度不可计算的情况，那些依赖于数据自然性解释的生成模型就不能行学习和应用。但是 GAN 依然可以使用，这是因为它引入了内部对抗的训练机制，可以通过内部对抗的方式，逼近一些很难计算的目标函数。

GAN 模型的一个重要优点，是生成模型 G 的参数更新，不来自于数据样本本身，而来自于于判别模型 D 的一个反传梯度。当然，这样子也有缺点，其一是导致模型的可解释性非常差，就如同一个黑盒一样，输入的是一个随机变量，而输出的却是我们想要的一个数据分布。其次是，在实际应用中 GAN 模型会比较难以训练。因为 GAN 模型要交替优化生成模型和判别模型，而这两个模型之间的优化，则需要很好的同步。

沈郑燕（2017）指出，GAN 在实际中的应用有很多，如图像超分辨率的识别，将之前古老的低分辨率的模糊图像，通过 GAN 进行变换得到高分辨率的带有丰富细节的清晰图像。如数据合成（使模拟数据更加逼真，与真实图像的差异性尽量小）、图像到图像的翻译（比如将语义标注图、灰度图或边缘图作为 GAN 的输入，我们希望的结果是输出能够和输入图一致的真实图像）等。

（四）受限玻尔兹曼机 RBM

受限玻尔兹曼机具有两层：可见层（V 层）以及隐藏层（H 层），两层神经元之间是全连接的，但是每一层各自的神经元之间并没有连接。也就是说，RBM 的图结构是一种二分图。玻尔兹曼机是允许同一层之间的神经元相连的，RBM 是一种简化了的 BM 模型。RBM 中的神经元都是二值化的，也就是说只有激活和不激活两种状态 0 和 1，可见层和隐藏层之间的边的权重可以用 W 来表示，W 是一个$|V| \times |H|$大小的实数矩阵。秦浩然（2016）指出，因为 V 和 H 都是二值化的，没有连续可导函数去计算，实际中采用的 sampling 的方法来计算，比如 gibbs sampling 的方法。

受限玻尔兹曼机可以理解为是一个能量模型（Energy based model，EBM），能量模型需要做的事情是先定义一个合适的能量函数，然后基于这个函数得到变量的概率分布，最后基于概率分布去求解一个目标函数（如最大似然）。RBM 的能量函数定义为：

$$E_\theta(v,h) = -\sum_{i=1}^{n_v} a_i v_i - \sum_{j=1}^{n_h} b_j h_j - \sum_{i=1}^{n_v}\sum_{j=1}^{n_h} h_j w_{j,i} v_i \quad (2\text{-}10)$$

如果写成向量/矩阵的形式，则为：

$$E_\theta(v,h) = -a^T v - b^T h - h^T W_v \quad (2\text{-}11)$$

那么可以得到变量（v, h）的联合概率分布是：

$$P_\theta(v,h) = \frac{1}{Z_\theta} e^{-E_\theta(v,h)} \quad (2\text{-}12)$$

章浩伟等（2019）认为，受限玻尔兹曼机可以看成是一个编码和解码的过程，编码的过程就是从可见层到隐藏层，反过来，解码的过程就是从隐藏层到可见层。例如在推荐系统中使用受限玻尔兹曼机，就可以把用户对各个物品的评分，当做可见层神经元的输入。用户的数量就是训练样本的容量。因为在实际中，用户不会对所有的物品都进行评分，所以对于任意样本，会存在有些可见神经元没有值的情况出现。优化目标是希望让解码后的数据和原始输入数据尽可能的接近。在推荐场景中，可以获取到用户对物品的评分矩阵，经过 RBM 的编码-解码过程处理后，不仅得到了已有评分对应的新评分，同时对未评分的物品进行预测，并将预测分数从高到低排序就可以生成推荐列表，也就是将 RBM 应用到协同过滤中。

三、小结

在回顾了深度神经网络的概述和相关常用模型之后，我们可以得出如下结论：

1）深度学习理论的核心在是在算力足够的基础上，以基于概率的方式，反复迭代、反复计算，去训练参数，最终能够在满足误差率的基础上，得到训练参数，用于预测未来的输入变量的输出，或者用于校验已有的输入数据。

2）深度学习是许多人工智能方面的应用的基础理论支撑。如图像识别、语音识别、语义识别等方面的技术日新月异，并已经在人们的日常生活中得到了广泛的应用。同时，深度学习的理论也为文本分析、舆情分析、基于时序数据的股票指标分析等都提供了理论基础。

3）LSTM 模型是一种递归神经网络的变种，对于时序数据的处理效果非常好，由于可以沿时间进行反向训练，对于时序序列数据中相隔很远的数据，往往也能起到效果，所以用此模型来对股票的时序数据进行训练可以起到很好的效果。

第三章 模型选择、指标设计、数据来源

第一节 模型选择

一、情感词分析模型选择

情感词分析的前提是获取到充足的语料数据，本文的语料数据的来源都是来自于互联网上各平台的公开的数据，采用爬虫工具进行数据的抽取、清洗、过滤、转换、存储等操作，最终得到适用的数据。最终选择的是基于 python 语言的开源爬虫工具 scrapy，开源项目网址为：https://github.com/scrapy/scrapy，具有数据分布式爬取、网页数据格式转换、爬取流程控制、中间件控制等特点，能够满足大规模实时数据爬取的项目需求。scrapy 的功能包括：1）基于 twisted 异步框架的高性能数据爬取，效率高、异步、非阻塞；2）采取可读性更强的 xpath 代替正则；3）强大的统计和 log 系统；4）支持 shell 方式，方便独立调试；5）统一过滤器、清洗规则等中间件。

在获取到了相应的数据之后，就是对于这些文本信息的处理了。前面文章中介绍了情感词分析的一些主要方法和步骤。在此，我们按照文本处理流程、情感词收取、情感分析算法，最终分词选取了基于字符的生成模型，词性标注采用了三层的隐马尔可夫模型，情感分析采用基于分类的算法进行训练，得到具体词的情感得分，这样就能够完整地解决，抽取的数据从文本分词、词性标注、情感词抽取、情感分析的一整套处理流程，可以将使用爬虫爬取的相关社区、论坛中的对公司的论坛帖子、评论信息，按照天、小时等为维度，进行相关的分析，比如判断用户对于公司情感的增长与降低的趋势情况等。本文最终选取了基于 python 语言的开源 github nlp 项目 snownlp，开源项目网址为：https://github.com/isnowfy/snownlp，作为中文的自然处理的库，并且带有训练好的中文词典，编码使用 unicode 编码。功能包括：中文分词、词性标注、情感分析、文本分类、转换成拼音、繁体转简体、提取文本关键词、提取文本摘要、TF（词频）计算，IDF（逆向文本频率）计算、Tokenization、文本相似度计算等。

示例如下：

```
from snownlp import SnowNLP
s = SnowNLP(u'这个东西真心很赞')
s.words                # [u'这个', u'东西', u'真心',
#    u'很', u'赞']
s.tags                 # [(u'这个', u'r'), (u'东西', u'n'),
#    (u'真心', u'd'), (u'很', u'd'),
#    (u'赞', u'Vg')]
```

s.sentiments # 0.9769663402895832 positive 的概率

输入股评中用户所发表的帖子或者所发表的评论信息，就可以对这些信息进行分词、词性标注以及情感词分析等操作。根据这些信息，就可以将这些信息作为参数，来判断这些舆论信息对于股票的价格、波动率等影响。

二、文本相似度模型选择

本文采用谷歌的 simHash 算法，来进行相似文本的查重工作。simhash 算法分为 5 个步骤：分词、hash、加权、合并、降维，具体过程如下所述。

1）分词：给定一段语句，进行分词的操作，将一段语句转化为有效的特征向量，然后为每一个特征向量设置 1-5，这 5 个级别的权重。

2）Hash：使用桶的思想进行拆分，哈希的存储结构如下：

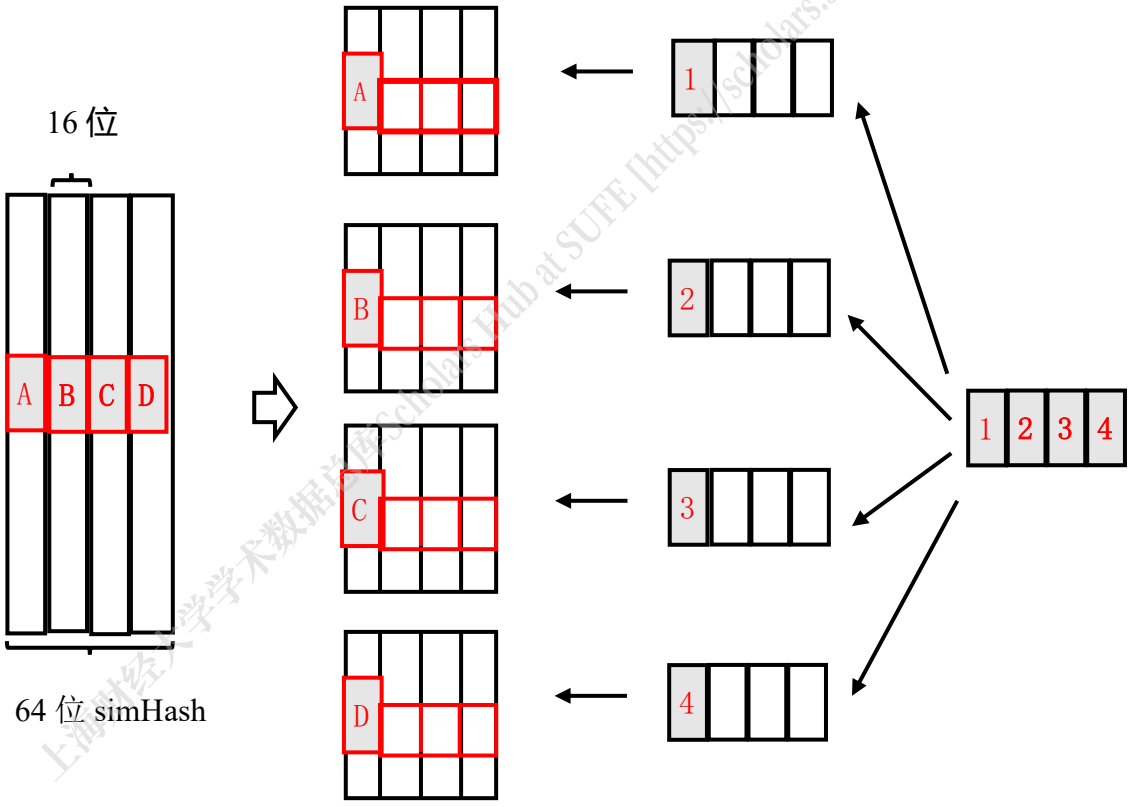

图 3-1 simHash 处理架构

3）加权：在哈希的基础上，给所有特征向量即分词后的集合进行加权，即 W = Hash * weight，规则为：为 1 时，hash 值、权值正相乘，为 0 时，hash 值、权值负相乘。

4）合并：将上述所计算出来的各个特征向量的加权结果进行累加，变成一个字节序列串。

5）降维：对于 n 比特的签名的累加结果进行降维操作，具体规则为：如果当前为大于 0 则最终结果为置 1，否则为 0，最后从而得到该语句的局部敏感哈希值，这样，便可以根据不同语句的 simhash 的不同，通过比较两者的海明距离来判断它们的相似度。

在得出文档的 simHash 签名值后，将其转换为一个数字签名，那么判断 2 个文档是否相似的问题，演变为了判断这 2 个 64 位数的汉明距离的问题。接着计算两个签名的海明距离即可。采用样本，通过对 64 位的 simHash 值进行准确率和召回率的分析，如下图所示。

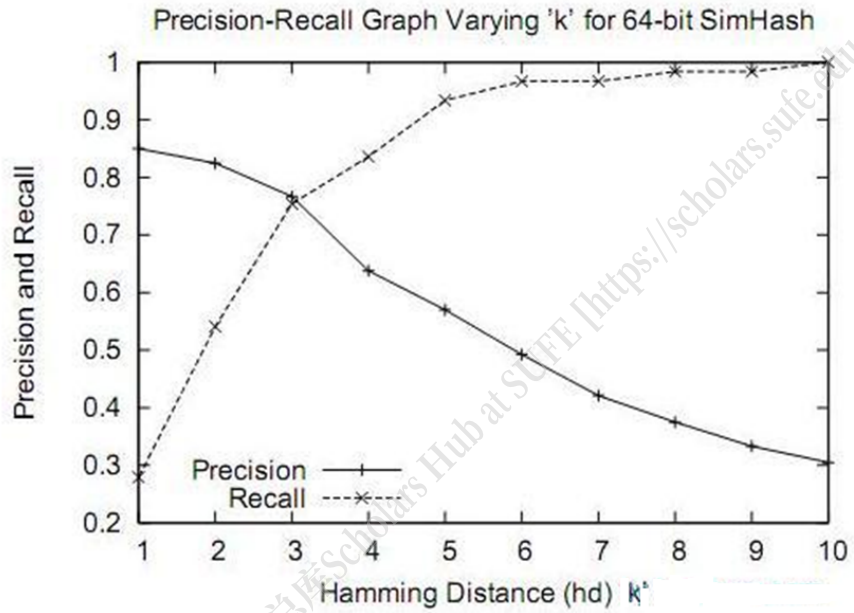

图 3-2 汉明距离取值的准确率和召回率

可以看出，一般取汉明距离为 3，可以同时满足准确率和召回率的要求。通过前面几步的分词、哈希、加权、合并、降维，以及计算汉明距离的操作，就可以进行文本相似度查重的操作了。在附录 A 详细介绍了基于 simHash 的关键代码。

三、深度神经网络模型选择

在上面一小节中介绍了数据的提取以及数据的处理，通过这些处理的步骤之后，我们就得到了后续所需要使用的数据了。具体数据在 mysql 数据库中的表结构见附录 B 中。在获取到这些数据之后，我们就可以使用统计学的一些方法进行相关的数理统计，验证这些指标和股票信息的关联关系，在得到关联习惯较大的高阶指标后，使用深度神经网络模型去训练舆情数据对于公司股价的影响因子以及影响程度。

在前面的介绍中，我们提到了 LSTM（长短期记忆网络），能够解决循环神经网络中长序列训练问题过程中梯度消失和梯度爆炸的问题。在工程方面，我们使用了谷

歌开源的基于数据流编程的、广泛运用于机器学习算法的编程实现的库：Tensorflow，可以用它来处理大量数据，快速建立数学模型，其前身是谷歌的神经网络算法库DistBelief。开源项目网址为：https://github.com/tensorflow/tensorflow。TensorFlow 是一个机器学习开源库，拥有一个全面而灵活的生态系统，其中包含各种工具、库和社区资源，可助力研究人员推动先进机器学习技术的发展，并使开发者能够轻松地构建和部署由机器学习提供支持的应用。可进行神经网络、生成对抗网络、基于注意力的神经机器翻译等大量机器学习的工作。tensorflow 的特点有如下：

1）拥有多层级结构，可部署于各类服务器、PC 终端和网页并支持 GPU 和 TPU 高性能数值计算。

2）用户向 TensorFlow 输入搭建模型所需的信息，并转化为可处理数据。

3）提供了很多的模块（函数、方法），在日常使用时进行模型搭建，用户可以调用这些模块中的功能。例如 gradient descent 梯度下降函数来求解模型的参数；如交叉熵损失 loss()函数来判断模型是否最优。

4）循环地迭代式训练和评估模型，以便确定模型中的参数。例如使用梯度下降（gradient descent）函数方法获得交叉熵损失最小化。

除了这些特点，tensorflow 同样支持 python 语言，这样就可以和之前的爬虫数据爬取、数据处理、情感识别，到模型训练计算、模型参数调优、模型效果评估使用相同的技术栈。

tensorflow 中完整支持 LSTM 神经网络模型，构建 LSTM 模型的步骤如下：初始化对象、构建输入数据（实际数据占位符，变量等）、创建输入层、构建隐藏层、构建输出层、构造损失（成本）函数、构造优化器、定义精准度评估函数、构建模型、将字符批量化、训练和保存模型。通过这些步骤，基于 tensorflow 的算法框架，就可以通过输入参数，优化参数，最后进行输出的预测和计算了。

第二节 指标设计

一、应变量指标

本文的目标是研究舆情对于公司股价的影响，具体探讨的影响指标是股价和波动率，通过结合这 2 个指标，使用股票真实波动率 ATR（average true range）这个指标作为应变量。ATR 指标是由 J.Welles Wilder 发明的，主要是用来衡量市场波动的强烈度，可以很好地量化股票的波动程度，即为了显示市场变化率的指标因此将这个指标作为目标。

ATR 这一指标对于长期持续边幅移动的时段是非常典型的，这一情况通常是发生在市场的顶部，或者是在价格巩固期间。

ATR 的计算方法为：

1）当前交易日的最高价与最低价间的波动幅度。

2）前一交易日收盘价与当个交易日最高价间的波动幅度。

3）前一交易日收盘价与当个交易日最低价间的波动幅度。

然后取这三者之间的最大值，便为真实波幅。计算公式如下：

$$TR = MAX((HIGH - LOW), ABS(REF(CLOSE, 1) - HIGH), ABS(REF(CLOSE, 1) - LOW)) \quad (3\text{-}1)$$

$$ATR = MA(TR, N) \quad (3\text{-}2)$$

二、自变量指标

对于输入自变量参数，具体的指标设计可以分为数理统计和基于深度神经网络的模型入参这两部分。

对于指标的选取，我们参考了舆情分析中常见的指标，并结合情感分析的维度进行设计和提取，主要考察时间维度、数量维度、显著维度、集中维度和意见维度这几个维度的指标，对于每个维度，精细化几个对应的二级指标。

1）时间维度：按照小时和天两个维度进行统计。

2）数量维度：按照时间维度的统计，结合爬取的多平台数据，分别统计帖子总量和回复总量，以及数量对于时间的分布。

3）显著维度：对于一些公众性的大事件，区别于常规的统计，剔除掉，按照事件分析法进行单独分析。

4）集中维度：对于一个事件的集中讨论，也需要单独关注，进行独立分析。

5）意见维度：对于大 V、股评 KOL 等用户发表的帖子，会得到更多的转发和讨论，所以可以给予单独的参数维度进行分析。

对于数理统计，我们对于给到的参数，进行一些数理统计参数的计算以及分析，如：样本均值、样式方差、中位数、极值、概率分布、假设检验等。在工程上，同样使用基于 python 的 sklearn（scikit-learn）和 scipy 的数据统计和计算的函数库，pandas 辅助数据格式转换并进行快捷运算，statsModels 进行统计建模，matplotlib 进行可视化图表展示。

对于基于深度神经网络的模型，我们将已经确定的这些指标作为输入变量，基于 LSTM 模型构建神经网络，分别构建输入层、隐藏层、输出层，并定义相应的损失函数、优化器、精准度评估函数等，经过多次迭代，最终得到满足精准度评估函数的模型优化参数。

再获取到数据之后，我们采用交叉验证法进行验证，将数据集 D 划分为 k 个容量大小相似的互斥子集，数据集 D 的划分公式如下：

$$D = D_1 \cup D_2 \cup \cdots \cup D_k, D_i \cap D_j = \emptyset \ (i \neq j) \quad (3\text{-}3)$$

为了保证每个子集D_i的数据分布一致性，从 D 集合中，通过分层采样的思想进行采集来获取。在每次处理时，使用 k-1 个子集的并集来作为训练集，余下的子集用作测试集，这样就获得了 k 组训练/测试的集合。从而可以进行 k 次训练和测试，最终返回的是这 k 个测试结果的均值。

第三节 数据来源

一、数据源获取

数据来源分为两类：舆情数据和股票交易数据。

舆情数据的来源主要来自：

1）各大类股吧：对于上市公司来说，在股吧里经常会爆出各种所谓的内幕信息，由于股吧里无价值的信息越来越多，股吧的言论舆情环境使之可能成为一些别有用心的人进行信息操控的场所，从而导致影响正常投资者的交易。还可能出现的情况下，因为负面的舆情的突然爆发，危及上市公司的股票正常架价格走势，有些严重的甚至还会影响上市公司的长期稳定发展。

2）各类财经网和社区论坛：这些网站通常是股民们日常查看股票咨询，相互交流的地方。股民们可以在这些地方，了解当前大盘行情、后期走向、投资建议等。但是由于互联网的言论的自由性，也会出现一些极端意见或者误导信息。比如，一些股民由于股市投资受损，为了进行报复，随意在财经网站上发表质疑某些上市公司存在财务造假、违规炒卖自家股票等问题，或者爆出一些所谓的内幕消息，诱导大家正常的投资者，这些行为也是引发舆情危机的重要因素。

3）微博等社交媒体：相对于股吧等网络社区来说，在微博上传播更加快速、互动性更加强，而且社会反响效果更加强烈等。一旦有用户发表关于股市的某个公司的突发信息，再经过微博民众、官方媒体、大 V 账号等进行放大传播，消息就会迅速扩散，最后导致舆情问题很难收拾。

知道了舆情数据的来源，就可以采用网络爬虫技术进行实时爬取，结合国泰安 CSMAR 数据库中股吧的历史舆情数据，进行数据合并处理。在拿到原始数据之后，需要对数据进行清洗、转换等操作，以达到满足后续的使用需求。具体的操作如下：

1）数据转换：对于爬虫爬取的 HTML 网页数据，需要转换为结构化数据，使用数据库存储，以便后续给到 pandas 数据处理的库就行处理。而对于国泰安 CSMAR 中股吧舆情数据，则需要下载对应数据，对处理文件数据，使得 pandas 数据分析类库能够使用。

2）数据清洗：对于爬虫爬取的数据，有很多无效数据，如：表情符号、停用词、无效水贴等，需要进行数据清洗操作，这些数据认为是无效数据，直接清洗掉。在清

洗之后，需要对数据进行一些整合，如计算帖子信息中情感词的评分情况，对帖子进行分词处理，计算帖子的情感得分等。

3）数据存储：采用数据库 MYSQL 进行数据的存储，按照结构化的方式，对于上市公司股票的相关舆情信息进行存储。

对于股票数据，同样选取国泰安 CSMAR 数据库中的个股交易数据，同样需要进行数据清洗，最终得到相关个股的股价增加情况和股价波动率情况的数据。

这样，就将所需要的数据进行了统一处理，获取到了最终我们需要分析的所有的数据了。

二、股票池选择

对于股票池的选择，我们选择不同的对照组进行分析，具体如下：

1）按照申万一级行业进行区分：从雪球网上爬取出各个行业分类，从计算机、非银金融、房地产、医药生物、电子、国防军工等行业中，分别选取所在行业的龙头企业作为分析的数据源，因为龙头企业关注度高，同时能够保证舆情数据的总体数量达到一定的数量，保证统计上的精确度。

2）因为某些个股的舆情信息不足，这对后续的舆情处理会带来偏差，因而设置舆情处理的阈值，在去除掉无用信息和停用词之后的舆情信息，同时经过相似文本过滤之后，达到了每天 200 条舆情数据，则进行处理，否则放弃之。

3）考虑小市值股票存在被游资炒作或者消息面的影响比较大，因而入选的标准是股票流通市值需要达到 1000 亿。

4）去除连续停牌、近两年内上市的复符合上述条件的股票。

在经过以上条件过滤之后，从 A 股三千多只股票中，筛选得到的符合条件的数据如下图：

表 3-1 申万一级行业符合要求股票

行业	股票
非银金融	中国平安、中国人寿、中国太保
农林牧渔	温氏股份、牧原股份、新希望
采掘	中国神华
化工	万华化学、荣盛石化
钢铁	宝钢股份
有色金属	紫金矿业
电子	海康威视、立讯精密、蓝思科技
家用电器	格力电器、美的集团
食品饮料	贵州茅台、五粮液、海天味业、伊利股份、泸州老窖
纺织服装	无
轻工制造	无
医药生物	智飞生物、长春高新、恒瑞医药、复星医药、迈瑞医疗
公用事业	无

交通运输	顺丰控股
房地产	万科 A、保利地产
商业贸易	无
休闲服务	中国中免
综合	无
建筑材料	海螺水泥
建筑装饰	中国建筑
电器设备	宁德时代、隆基股份
国防军工	中国重工
计算机	恒生电子、用友网络
传媒	无
通信	中兴通讯
银行	招商银行、浦发银行、工商银行
汽车	上汽集团
机械设备	三一重工

从图中可以看出，因为设定了较高的准入门槛，所以入选的都是长期在市场上有比较好表现的股票，同时股票走势大多比较平稳，这样，就可以有效剔除一些股票炒作方面的影响，从而可以进行相关的分析。

第四章 实证分析

第一节 网络舆情市场效应检验

一、舆情讨论热度因子

对于上一章所选取的股票，在采取了预处理的措施之后，得到了格式化的舆情帖子数据。我们进行简单的统计处理，以及通过分词和情感词极性抽取的方式，获取到以日为统计单位的舆情汇总数据，如每日帖子汇总数，正负情感判别等，做出简单的网页展示效果，用于展示每日舆情基本数据，如下图：

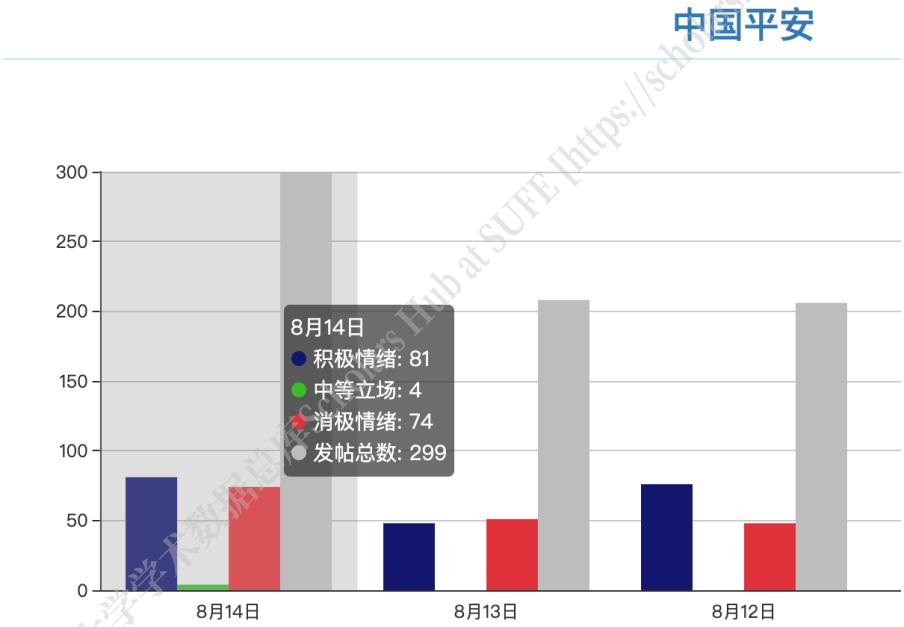

图 4-1 中国平安每日舆情信息统计展示

上图展示了中国平安这只股票在 2020.8.12～2020.8.14 之间，去重后的帖子总数，以及基于情感得分计算的情感分布情况。如在 2020.8.14 的时间，发帖总数为 299，积极情绪帖子数量为 81 条，中性情绪帖子数量为 4 条，消极情绪帖子数量为 74 条，基于这些数据可以直观判断相关数据的每日走势情况。

我们对于筛选出来的股票，计算出每个股票每日舆情帖子数量和股票真实波动率之间的关系，最后计算组合内的总体平均数。可以看出，伴随着每日的帖子发布数量的多少，也说明用户对于公司的讨论的热情的增减。并且用户每日发帖中，所展现出来的情感因素也有所不同，积极和悲观的因素交替产生。

根据每日舆情的讨论数据，结合股票的 ATR 数据，验证两者之间是否存在关系。首先是找出近一个月来的舆情数据和股票的波动数据，选取的时间点为 2020.7.15～2020.8.15，共 23 个交易日。作出这两者之间的散点图如下：

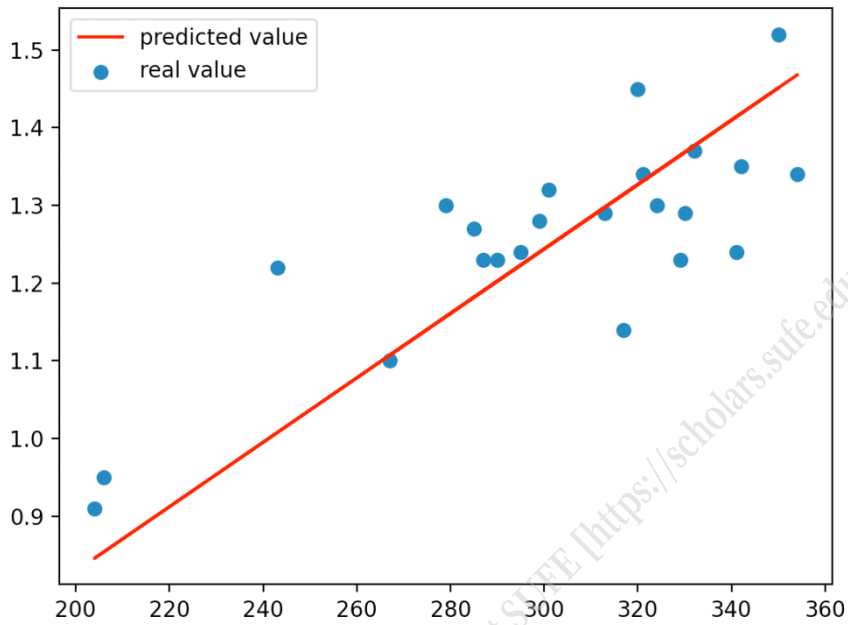

图 4-2 舆情讨论热度因子与 ATR 真实波动率散点图

从散点图中，可以舆情讨论热度因子的数据与股票的真实波动率存在着一定的联系，每日讨论的热度提高，会刺激着股民交易的热情，自然导致股票真实波动率的提高。接下来进行数理统计的分析，如图所示：

OLS Regression Results

Dep. Variable:	atr	R-squared (uncentered):	0.994
Model:	OLS	Adj. R-squared (uncentered):	0.994
Method:	Least Squares	F-statistic:	3588.
Date:	Sat, 29 Aug 2020	Prob (F-statistic):	7.26e-26
Time:	13:17:04	Log-Likelihood:	20.631
No. Observations:	23	AIC:	-39.26
Df Residuals:	22	BIC:	-38.13
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
opinion	0.0041	6.92e-05	59.902	0.000	0.004	0.004

Omnibus:	0.209	Durbin-Watson:	1.040
Prob(Omnibus):	0.901	Jarque-Bera (JB):	0.348
Skew:	-0.187	Prob(JB):	0.840
Kurtosis:	2.528	Cond. No.	1.00

图 4-3 舆情讨论热度因子与 ATR 真实波动率统计分析

从 R 平方值和 P 值可以看出，舆情讨论热度因子和股票的真实波动率之间存在线性关系，可以对股票的波动进行一定的反应。

二、情感得分因子

上一节中，介绍了获取用户每日用户讨论总量的获取和简单的正负面情感的拆分。在获取了这些帖子信息之后，就可以通过 NLP 相关技术，去计算每日用户对于公司的情感得分指标。下图展示了中国平安这个股票在百分制下的每日舆情得分情况。

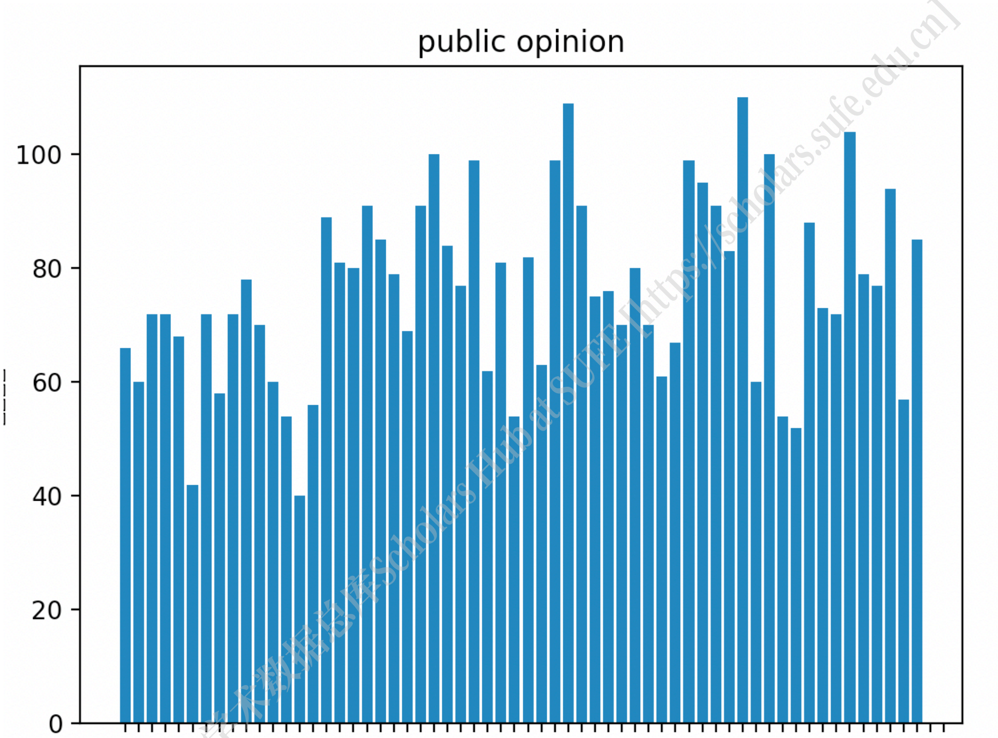

图 4-4 中国平安舆情情感得分统计

从图中可以看出，用户对于股票的每日的评论信息中，所包含的情感得分每日都在变化，在股票出现负面消息或者股价下跌的时候，会出现情感得分下降的情况，说明用户的帖子信息中，会出现偏悲观的词语出现，导致所流露出的情感偏向于负面。结合图 4-1 的每日正负面评价的统计，图 4-2 对于每日帖子的平均得分做了量化的处理。

接下来，根据每日舆情的情感指标数据，结合股票的 ATR 数据，验证两者之间是否存在关系。依然是选取的近一个月来的舆情数据和股票的波动数据，选取的时间点为 2020.7.15～2020.8.15，共 23 个交易日。作出这两者之间的散点图为：

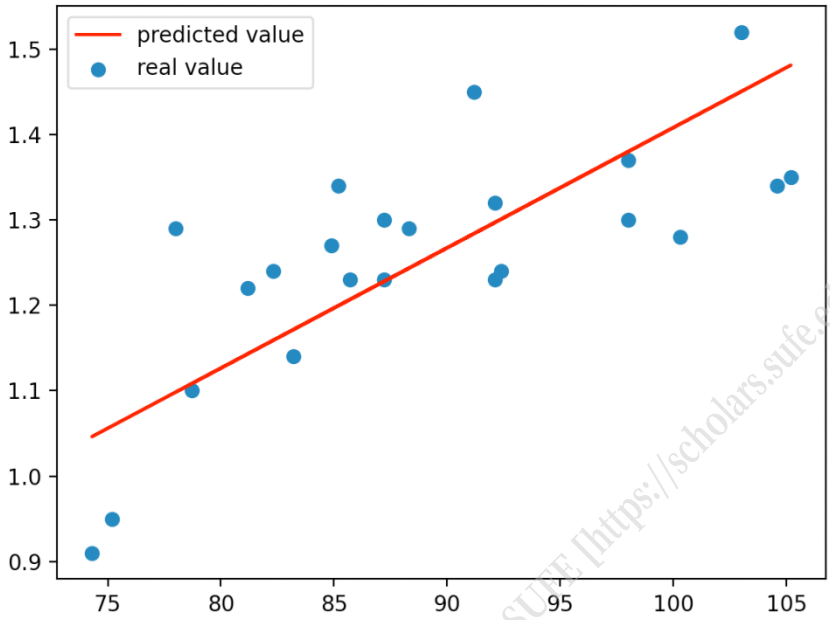

图 4-5 中国平安舆情情感得分与股票真实波动率散点图

从散点图中，可以看出舆情情感数据与股票的真实波动率存在着一定的联系，用户对于股票的正面评价提高，对股票观感变好，会刺激股民进行交易，导致股票真实波动率的提高。接下来进行数理统计的分析，如图所示：

OLS Regression Results

Dep. Variable:	atr	R-squared (uncentered):	0.994
Model:	OLS	Adj. R-squared (uncentered):	0.994
Method:	Least Squares	F-statistic:	3759.
Date:	Sat, 29 Aug 2020	Prob (F-statistic):	4.36e-26
Time:	13:41:55	Log-Likelihood:	21.163
No. Observations:	23	AIC:	-40.33
Df Residuals:	22	BIC:	-39.19
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
sensation	0.0141	0.000	61.311	0.000	0.014	0.015

Omnibus:	1.318	Durbin-Watson:	1.602
Prob(Omnibus):	0.517	Jarque-Bera (JB):	0.921
Skew:	0.140	Prob(JB):	0.631
Kurtosis:	2.061	Cond. No.	1.00

图 4-6 中国平安舆情情感得分与 ATR 真实波动率统计分析

从统计分析中 R 平方值和 P 值可知，每日舆情的情感指标数据和股票的真实波动率存在着一定的关系，可以起到一定的表征作用。

三、KOL 影响因子

接下来，我们验证热门数据对于股票价格和波动性的影响。我们统计的是雪球网所讨论得热门的股票，数据的处理思路如下：在主页获取雪球活跃用户，在活跃用户的发帖中统计股票被提及的次数，之后分析出个股讨论度与价格走势之间的关系。

首先是数据源，我们使用爬虫获取到股票名称和股票所在的行业信息，然后是获取热门数据，我们选取 2020.8.3 之前的 3 天作为数据分析，获取到如下热门用户的信息如图所示：

```
1  3079173340 银行螺丝钉
2  6195589551 铁公鸡金融
3  6146592061 持有封基
4  5819606767 DAVID自由之路
5  3727797950 草帽路飞
6  1821992043 ice_招行谷子地
7  1029319098 王剑
8  2554781328 学经济家
9  1135063033 魏斌杰
10 4117950729 投资人李木白
11 3366431401 谦和屋
12 9137056825 般若波罗蜜
13 8291461932 房杨凯-众研会
14 7995171607 薛定谔家的小猫
15 8869975766 喝牛二的战总
16 3727797950 草帽路飞
17 9226205191 处镜如初
18 6167347258 牛和智
19 2340719306 流水白菜
20 1234051357 精算盘管家-徐老师
```


图 4-7 2020.7.30 雪球热门帖子发帖人之一

21	5180243808 businesslike
22	3770558188 青侨阳光
23	9518372158 Stevevai1983
24	7355827634 微进化ing
25	7102683784 深圳零存整取
26	3727797950 草帽路飞
27	7952175174 自由老木头
28	7157351566 欢无之尘
29	6292744184 人淡若菊
30	6056806984 朱酒
31	9137056825 般若波罗蜜
32	2621452699 加班的会计
33	5296621426 刘步尘
34	8067986264 张立聪
35	1457840645 东易日盛
36	7082832549 李暮泊
37	5180243808 businesslike
38	7947659357 雨枫
39	4212501514 投资者牛小顿
40	3995301829 小狮子旺财

图 4-8 2020.7.30 雪球热门帖子发帖人之二

其中，第一列代表雪球的用户 id，第二列是用户昵称，以上两图只展示了前 40 位热门用户。

这其中也包括了一些预处理的工作，如使用分词工具进行分词，我们使用的分词工具是中科院的 ICTCLAS (Institute of Computing Technology, Chinese Lexical Analysis System)，主要功能包括中文分词；词性标注；命名实体识别；新词识别；另外，也支持用户自定义词典的功能，从而可以自己扩充领域词库。从 ICTCLAS3.0 的数据来看，分词速度使用单机（机器 4 核 8G 配置）达到 1MB/s 左右，分词精度超过了 98%，API 的容量不超过 200KB，各种词典数据，再经过压缩算法压缩后容量不到 3M。

分词之后，统计评论中的词语提及次数，把频率较高的当做股票领域词语，然后根据语料库，提取特征值。使用 snownlp 分类评论的正面和负面情绪，来判断所在公司的热门词汇以及最终的得分情况。首先是获取的具体讨论的相关行业以及被谈及的次数，如图所示：

1	卫生%1
2	旅游环境%1
3	采矿业%1
4	租赁和商%1
5	食品制造%1
6	农林牧渔%1
7	汽车制造%1
8	批发零售%1
9	交通物流%1
10	橡胶塑料%1
11	建筑业%1
12	医药制造%2
13	非金属制%2
14	金融业%2
15	茶酒饮料%3
16	房地产业%3
17	信息技术%4
18	通信设备%4

图 4-9 雪球热门帖子涉及行业

可以看出，在热门帖子中，提到了如下相关的行业，我们再选取相关行业当天的走势图来进行比较得出,信息技术和通信设备这 2 个行业，都能够获取超出大盘的超额收益率。

图 4-10 雪球热门帖子涉及行业超额收益率

在此，本文分析了舆情中舆情数量的趋势、舆情评分的提高、以及 KOL（关键意见领袖）对于股票的影响，从分析中可以看到，这些指标对于股票都是有一定的相关性的。会极大地影响股票的价格和波动率。并且可能有一定的自相关性，伴随着股价的上涨和舆情的高涨，进一步强化了自身的量级。

四、多元回归分析

最后，我们分析前面的舆情讨论热度因子、情感得分因子、KOL 影响因子对于股票真实波动率的影响情况。我们采用多元线性回归模型进行分析，并且对于 KOL 影响因子，以 0 和 1 分别代表当日是否存在首页传播、大 V 传播的因素影响情况。得出的多元线性回归的统计结果如图所示：

OLS Regression Results

Dep. Variable:	atr	R-squared (uncentered):	0.997
Model:	OLS	Adj. R-squared (uncentered):	0.996
Method:	Least Squares	F-statistic:	1998.
Date:	Sat, 29 Aug 2020	Prob (F-statistic):	6.11e-25
Time:	14:09:20	Log-Likelihood:	27.597
No. Observations:	23	AIC:	-49.19
Df Residuals:	20	BIC:	-45.79
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
opinion	0.0020	0.001	3.575	0.002	0.001	0.003
sensation	0.0075	0.002	3.963	0.001	0.004	0.011
kol	-0.0597	0.045	-1.339	0.196	-0.153	0.033

Omnibus:	0.870	Durbin-Watson:	1.617
Prob(Omnibus):	0.647	Jarque-Bera (JB):	0.756
Skew:	0.122	Prob(JB):	0.685
Kurtosis:	2.146	Cond. No.	866.

图 4-11 多元线性回归统计分析

从调整 R 平方值和 P 值参数可以看出，用这 3 个高阶舆情指标去拟合股票的真实波动率指标，起到了很高的拟合效果，证明了舆情指标可以很好地说明股票的真实波动率情况。

第二节 舆情对股价波动分析

在这里，我们使用基于 LSTM 的模型，对于公司舆情指数加入指标，来得出一套系统，分析舆情指标对于验证股市的相关引导作用。舆情相关的指标为：每日有效帖子发帖数与回复数加权得分、企业每日舆情得分、以及 KOL 对于公司的影响因子这 3 个，通过这 3 个参数，以及估计公司股票的历史股价以及股票波动率，来对未来的公司股价和波动率进行预测。

股票风险一般可用波动率来衡量，传统方法利用历史数据来构造模型预测未来波动率，如 ARCH 模型和 GARCH 模型。这一方法假设金融数据是稳态随机过程， 因而未来风险与历史风险在统计意义上一致。这一假设显然过于粗糙，因为不论是市场环境还是股票发行者的经营状况都会随着时间发生显著变化，用历史数据对未来进行预测本身具有极大风险。为提高风险预测的准确性和实时性，需要在基于历史数据建模的同时，快速学习当前市场和标的股票的动态特性。因而提出利用长短期记忆循环神经网络来实现这一方案。

循环神经网络(Recurrent Neural Network, RNN)是处理序列数据的有力工具。传统神经网络模型假设输入到网络中的各个数据片段是互相独立的，因而无法对时间序列数据进行建模。在循环神经网络中，一个序列当前的输出不仅依赖于当前输入，同时也依赖前一时刻的网络状态，这意味着这一网络可以对历史输入信息和系统状态信息进行记忆，并基于当前网络所处的状态计算当前输出。循环神经网络通过在隐藏层节点间加入反馈回路来实现。利用 LSTM 对历史信息的记忆能力，用来学习股票数据中长记忆性特征，并将 其应用于股票数据的波动率预测中。使用 LSTM 在线学习的能力，通过每天输入最新数据更新 LSTM 模型，使模型遗忘过久的历史信息，反应市场和标的股票的当前状态，提高波动率预测准确性。利用 LSTM 多因子建模能力，解决了传统波动率预测模型中仅考虑往期波动率这一种因子的局限性。

基于 LSTM 的模型搭建有如下几步：

1）引入相关库：如 numpy、pandas、sklearn、tensorflow、keras、math 等。

2）引入参数。

3）构建模型：获取相关高阶舆情数据和股票的金融数据。

4）数据正则化：scikit-learn 中的 MinMaxScaler()对数据进行了正则化。

5）搭建神经网络：在基于 LSTM 的 RNN 中，每个隐层节点由一个 LSTM 构成。每个 LSTM 接收一个输入，给出一个输出，并在一个记忆单元中记住当前系统状态。

不同于传统 RNN，LSTM 引入了三个门变量，如下所示:

1）输入门:表示是否允许信息加入到记忆单元中。如果该值为 1（门开），则允许输入，如果为 0（门关），则不允许，这样就可以摒弃掉一些无用的输入信息。

2）遗忘门:表示是否保存当前隐层节点存储的历史信息，如果该值设置为 1 则保留，如果设置为 0 则清空当前节点所存储的历史信息。

3）输出门:表示是否将当前节点输出值输出给下一层（隐层或者输出层），如果该值设置为 1 则当前节点的输出值将作用于下一层，如果设置为 0 则丢弃该信息。

系统选择五个输入结点，分别对前一天收盘时该股票的波动率、收益率、当天舆情数量、当天企业舆情评分、KOL 影响指数, 输出对应下一天该股票波动率的预测值。中间隐藏层包括 20 个 LSTM 单元。

系统流程图如下：

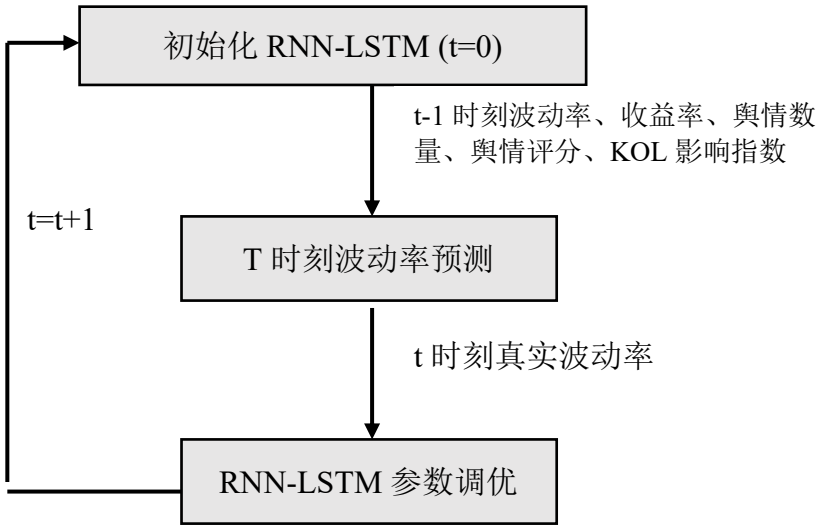

图 4-12 RNN-LSTM 训练框架

其中的核心思想还是梯度下降。采用 LSTM 模型，设置好 loss 损耗函数，最终就可以训练出模型了。训练的参数和原始数据拟合的结果如下，可以看出，通过舆情相关的参数以及股票股价和股票波动率等参数，可以很好地对股票的波动率进行预测。

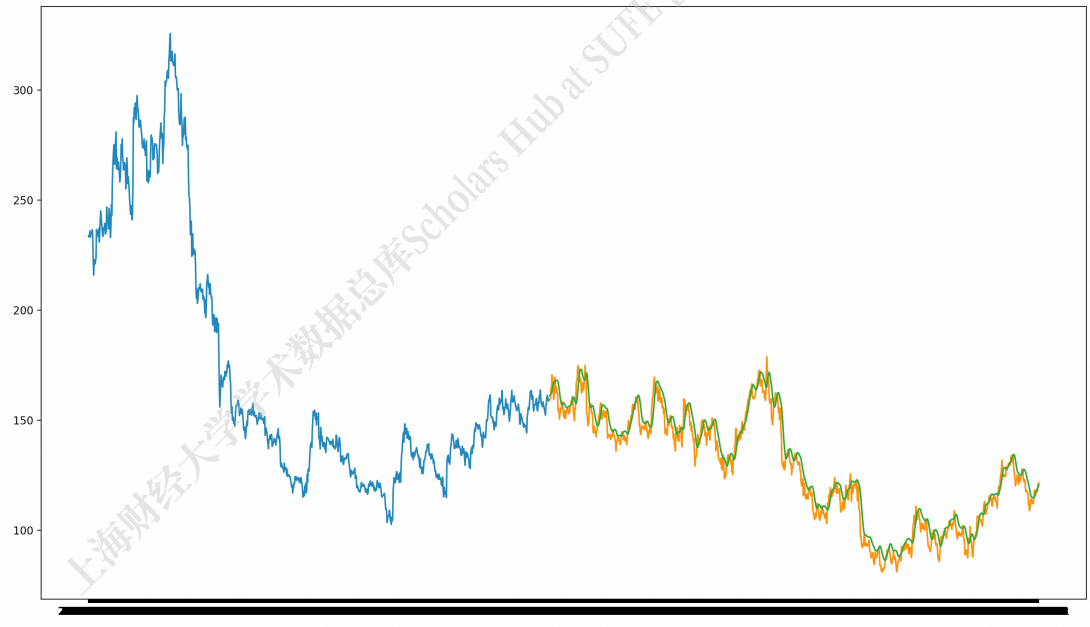

图 4-13 RNN-LSTM 数据拟合结果展示

图中统计了两年时间内，公司舆情的数量、情感得分、以及热门帖子信息对于公司的真实波动率的影响情况。后面一半为基于 LSTM 模型的预测效果，可以看到，基于舆情的高阶指标，能够很好的进行公司股票波动率的预测。经过 LSTM 模型训练之后，基于 adam 优化器，最后得出的平方差误差为：3.8%。那么可以说明，在使用深度神经网络给予时序数据的分析处理上，网络舆情的高阶指标，可以很好的表征公司股票波动率的变化。

第三节 实证结果进一步说明

在本章的前两节中，分别在数理统计和深度神经网络的角度，去验证了舆情数据对于公司股票收益率、波动率等的影响。第一节是是先获取到可以统计的几个高阶舆情指标：加权舆情量、企业舆情情感评分和 KOL（key opinion leader）关键意见领袖影响因子这 3 个参数，先验证这几个参数和企业股价之间的关联性。在验证好了关联性之后，再通过深度神经网络，通过舆情数据来预测公司股价的波动率指标。对于分析的结果，有如下几点说明：

1）在选取股票指标时，没有选用股价或者收益率指标。因为在热门事件发生时，公司可能股价直线下降或者之间上升或者波动，但无一例外，公司股票的波动率都是更增大，另外一点则是，股票经常会出现拆股的现象，计算工程中，需要进行复权的操作，对于回溯的数据处理会很麻烦，而且并不直观。所以采用股票波动率的指标能够更好的反应舆情数据对于股票的影响情况。

2）在选取这几个舆情的指标之后，通过构建基于 LSTM 的深度神经网络，设置损耗函数，来进行模型的训练，最终可以获取到拟合数据非常好的效果。模型可以对各种不同参数进行参数调优，如增加 dropout 值、改变 LSTM 层数或增加训练迭代轮数（epoch）数。但 LSTM 的预测并不足以确定股票价格将上涨还是下跌。另外，股价受到公司新闻和其他因素的影响，如公司的非货币化或合并、分拆，另外还有一些无形的因素，往往无法事先预测。

3）所采用的深度学习模型，本质是一个概率的事件集，即使是像 alphaGo 这种，在前期也是屡次失败的，通过不断学习，最终取得了胜利。因为按照深度学习的模型，alphaGo 每走一步棋，所代表的统计含义，是代表走完这一步之后的棋盘局面会是相对最好的概率。受限于算力、硬件资源等，深度学习不可能计算每一种可能性，只能是采取局部最优的方式，如采用牛顿下降法、改进梯度下降法等，让我们达到局部最优，而非全局最优。即使达到了全局最优，深度学习模型，依然只是给出每一种结果成功的概率，概率只是可能性，并不确定，依然代表了模糊正确性。

4）由于个股数据量较少，LSTM 模型的可扩展程度和复杂度受到很大制约，特征量的选择也受到限制（若 input 的特征量太多，而数据量较少的话，会使一部分特征量不能发挥出应有的作用，也极易造成过拟合），所以需要数据量足够大，才能起到泛化的作用，而一旦系统失去了泛化能力，那么就有必要从头开始训练模型并重新进行测试。同时，实验发现，LSTM 模型的运行原理中，会根据上下连接的数据切片修正自己的长短记忆内容，也就是具备一定的推理能力，在使用这个模型时，需要给与足够的数据，才能让模型能够进行推理。

将来，则希望能在个股/指数的小时或分钟数据上测试 LSTM 的性能。另外，将探究 LSTM 模型能否将属于一个行业的所有股票数据一起处理也是一个可选的方向。

第五章 结论与思考

第一节 研究结论

本文所做的研究，是希望在之前已有的网络舆情对于股价的影响的数理统计层面相关研究的基础之上，再进一步进行研究，探究网络舆情相关的高阶量化指标对于股票的走势例如股价、股票的波动率等因素的具体影响情况，本文采用了数理统计结合自然语言处理和深度神经网络相关的技术的研究方法。

获取了网络舆情得分以及网络舆情的传播度相关参数，另外则陈述了每日雪球热门话题对于股票的影响情况。从统计学的角度论证了，这些情感因素确实和股票的走势存在着某种相关性。舆情的快速传播和帖子中对于一个股票的不断高唱赞歌，或者不停的负面报道，都严重地影响了股民的正确判断，并且随着舆情的快速传播，股票也产生了剧烈的波动，成交量迅速扩大，在一片赞歌中，股票涨了许多，或者一片唱衰声中，股票下跌了许多。这反映了市场有效性理论在目前互联网技术高度发达的今天，可能存在一定的局限性，因为外部的消息太多，并且会不停推送到用户的手机、电脑中，无孔不入，使得股民非常容易受到这些消息以及评论的影响，做出错误的操作决定，而不是做为一个理性人，去研究公司的实际内在价值。在做了相关的数理逻辑的论证之后，尝试使用了深度神经网络的方法，通过舆情信息的高阶指标，建立模型，去预测其对于股票的价格和波动率的走势关系。因为 LSTM 在时序数据上很好的特性，所以采用 LSTM 的模型去训练数据，最终获取了一定的效果。

所以经过实验论证，可以认为，在当前的 A 股市场，网络舆情数据对股票的影响是非常重要的，在一些突发性事件爆发或者知名大 V 的一些受众面广、传播度广的文章，都会或多或少地影响公司股票的价格和波动率。这也是因为我国的股票市场是一个散户为主的市场，而散户的特点就是对相关的公司的研究没有机构那么深入和细致，容易受到外部因素的干扰，作出错误的判断，特别是在一些热门股上，非常容易受到情绪的影响。因而导致了市场上一些股票的暴涨暴跌的情况时有发生，被某些人利用，因为通常散户都可能因为看到消息和评论信息，追高买进去，但最终被套牢在高点。

同时本文使用了网络爬虫、自然语言处理、深度神经网络等相关技术，能够对公司的股票数据和网络帖子和评论进行实时的数据爬取和计算，对于某些股票的股价和波动率起到了一定的预警的作用，并能够预测相关股票的未来走势情况。

股票作为金融行业的典型代表，在新的形势下面临着严峻的挑战，股票声誉风险与舆情关系到股票的品牌价值、经营成效和发展进步，一旦遭遇舆论危机，不仅会直

接损害股票的价格，影响股民的业绩，甚至危及其生存，影响社会稳定。因此，在当下，上市企业应该审时度势，需要采用多种方式，和股民多在网上沟通，同时尝试建立一些切换机制与对接机制，可以从容应对舆情。

第二节 不足与展望

本文尝试解决网络舆情对于股价影响方面相关的问题，但是这其中还是涉及了很多不足的地方。如对于舆情数据和股票关系这部分，可能会有另外的解释，例如，有人是认为是因为本身股票价格的走高，带来了更多的赚钱效应，因而吸引了更多人的目光，自然会在相关股票讨论区、论坛等地方畅所欲言、各抒己见，这种人气效应，会吸引更多的其他人的关注，这就是索罗斯所提出的“反身性”理论，被人们的信念改变的现实又会影响人们的信念，从而使人们的信念更加强烈，进而再次影响和改变现实。而舆情信息的日益增多，对股票的无休止的追捧，不正式这一理论在现实中的写照吗？同时，对于舆情信息的处理是一件非常繁琐并且专业的事情，因为对于特定语义下的句子，是会有不同的含义的，对于公司相关的舆情信息，更是如此，需要有专家意见，进行人工标注，训练数据，才能够正确的识别相关语句的主观情感是偏悲观还是乐观。还有，对于数据清洗，停用词的删除，无用信息的过滤，也是一个非常专业的工作，需要专家意见进行指导展开工作才行。这些工作因为时间和条件的限制，并不能完美的实现。再次，这样庞大的网络舆情数据的存储和计算，也必然会消耗大量的计算机硬件资源，可能出现因为太大的数据量，实时数据计算缓慢，最终并不能达到实时的效果，这些都是工程实践中可能遇到的问题。

对于舆情相关的工作，个人觉得非常具有理论意义和现实价值。之前人们对于舆情数据的利用主要还停留在理论研究方面，或者对于历史数据的分析，来论证一些结论，但是并没有使用舆情信息来真正指导实时的股票交易。在股票的实际操作中，使用的还是非常有限，很多基金经理还是使用传统的公司基本面研究的方法进行选股的操作。而随着技术的进步，云计算的普遍应用以及人工智能相关理论的发展，能够将看上去不可量化的网络舆情信息，通过统计学或者深度神经网络的算法，来进行高阶指标的训练和提取。假如后续，能够对于大盘信息或者基于个股的信息的网络舆情实时数据，通过网络爬虫的爬取技术，对于各大主流股票平台或者股票社区、论坛，做到能够分钟级别的数据获取，再基于实时流数据的处理，以及在线计算和离线计算的算法、模型的训练，一方面对于大盘的舆情信息做一个量化的展示，另一方面，自动化地基于舆情信息来进行股票的买卖操作，可以预想，这样自然会获得不菲的超额收益。在此展望下这部分的内容能够在量化基金方向得到长足的使用，并取得不错的收益情况。

参考文献

[1]戴媛，姚飞，2008，《基于网络舆情安全的信息挖掘及评估指标体系研究》，《情报理论与实践》，第 873-876 页。

[2]谈国新，方一，2010，《突发公共实践网络舆情监测指标研究》，《华中师范大学学报》，第 3 期，第 66-70 页。

[3]张栋凯，齐佳音，2015，《基于微博的企业突发危机事件网络舆情的股价冲击效应》，《情报杂志》，第 3 期，第 132-137 页。

[4]史青春，徐露莹，2014，《负面舆情对上市公司股价波动影响的实证研究》，《中央财经大学学报》，第 10 期，第 54-62 页。

[5]张世军，程国胜，蔡吉花，杨建伟，2013，《基于网络舆情支持向量机的股票价格预测研究》，《数学的实践与认识》，第 24 期，第 33-40 页。

[6]朱健，2015，《网络舆情与上市公司股价的相关性研究》，《税收经济研究》，第 1 期，第 85-95 页。

[7]赵雪，2019，《深度学习在量化交易中的应用》，北方工业大学硕士论文。

[8]武建新，王仕宏，华一，2015，《基于计量分析的负面网络舆情对上市公司价值的影响》，《当代会计》，第 7 期，第 3-4 页。

[9]郭建峰，刘樱，陈有为，温景岗，2017，《大数据网络舆情对证券投资收益与风险影响研究》，《经济研究导刊》，第 35 期，第 127-129 页。

[10]吴鹏、刘恒旺、沈思，2017，《基于深度学习和 OCC 情感规则的网络舆情情感识别研究》，《情报学报》，第 9 期，第 972-980 页。

[11]徐琳，2013，《网络舆情对股价波动影响的实证研究》，西南财经大学硕士论文。

[12]朱昶胜，孙欣，冯文芳，2018，《基于 R 语言的网络舆情对股市影响研究》，《兰州理工大学学报》，第 4 期，第 103-108 页。

[13]宋锐，洪莉，林鸿飞，2009，《基于的观点持有者识别及其在观点摘要中的应用》，小型微型计算机系统，第 7 期，第 1462-1466 页。

[14]董静，孙乐，冯元勇，黄瑞红，2007，《中文实体关系抽取中的特征选抒研究》，中文信息学报，第 7 期，第 81-85 页。

[15]白凯敏，2016，《神经网络和深度学习在量化投资中的应用》，山东大学硕士论文。

[16]许林杰，2003，《中文文本分词研究》，山东师范大学硕士论文。

[17]安子建，2017，《基于 Scrapy 框架的网络爬虫实现与数据抓取分析》，吉林大学。

[18]李海燕，2014，《网络舆情爬虫系统的设计与实现》，厦门大学硕士论文。

[19]徐宁，2015，《主题爬虫搜索策略及关键技术研究》，重庆大学硕士论文。

[20]户保田，2016，《基于深度神经网络的文本表示及其应用》，哈尔滨工业大学硕士论文。

[21]李然，2015，《基于深度学习的短文本情感倾向性研究》，北京理工大学硕士论文。

[22]包姣，2017，《基于深度神经网络的回归模型及其应用研究》，电子科技大学硕士论文。

[23]赵晨，2014，《动态神经网络在量化投资预测中的应用》，复旦大学硕士论文。

[24]邹振华，2013，《基于文本挖掘的量化投资系统》，华南理工大学硕士论文。

[25]朱博雅，2012，《一种基于数据挖掘的量化投资系统的设计与实现》，复旦大学硕士论文。

[26]李聪，杨德平，孙海涛，2012，《基于ＥＭＤ与ＢＰ神经网络的中国股票指数期货价格预测》，《青岛大学学报》，第 25 期，第 78-83 页。

[27]李鑫欣，关菁华，2019，《基于 Python 的豆瓣读书网站用户信息采集》，《电脑知识与技术》，第 8 期，第 4-6 页。

[28]陶晨，杨剑平，苏淼，鲁佳亮，2020，《基于语料大数据的“潮”文化特征计算与分类》，《丝绸》，第 2 期，第 41-46 页。

[29]欧丽粤，毛红霞，赵春，熊浩宇，李荟，2019，《基于豆瓣音乐网的数据采集与清洗》，《信息与电脑（理论版）》，第 18 期，第 151-152 页。

[30]Fama,E.F., 1965, “The behavior of Stock-Market Prices”, The Journal of Business, vol.38, no.3, pp.34-105.

[31]Fama,E.F., 1970, “Efficient Capital Markets: A Review of Theory and Empirical Work”, The Journal of Finance, vol.25, no.2, pp.383-417.

[32]Fama,E.F., 1991, “Efficient Capital Markets II”, The Journal of Finance, vol.46, no.5, pp.1575-1617.

[33]Bing,L., Wynne,H., Yiming,M., 1998, “Integrating Classification and Association Rule Mining”, KDD-98.

[34]Appelt,D., Israel,D., 1999, “Introduction to Information Extraction Technology”, AI Communications, vol.12, no.3, pp.161-172.

[35]Rui-xing,C., Yi,C.,Chun-yuan,D., Di,M., 2018, “Design of Data Capture Program Based on Web Crawler Technology”, Proceedings of 2018 International Conference on Information, Electronic and Communication Engineering(IECE 2018).

[36]Jia-dong,L.,Yu-yi,O., 2018, “An Improved XSS Vulnerability Detection Method Based on Attack Vector”, Proceedings of 2018 International Conference on Modeling, Simulation and Analysis(ICMSA 2018).

[37]Rui,S., Yu-qiang,G., Xian-hua,F., Shi-ming,Z., 2016, “Practice Research on the Subject Knowledge Platform Based on the Knowledge Architecture”，Proceedings of 2nd Information Technology and Mechatronics Engineering Conference(ITOEC 2016).

[38]Jeffrey,D., Sanjay,G., 2008, “MapReduce”, Communications of the ACM.

[39]Allan,H., Marc,N., 1999, “Mercator: A scalable, extensible Web crawler”, World Wide Web, vol.4, no.2, pp.219-229.

[40]Chakrabarti,S., 2002, “Mining the Web:Discovering Knowledge from Hypertext Data”, Journal of Women s Health, vol.55, no.3, pp.275-276.

附录

附录 A 基于 golang 语言的 simHash 相似度查重的核心代码

```
package controllers
import (
    "fmt"
    "runtime"
    "commlib"
    "path"
    "errors"
    "strconv"
    "sync"
    "time"
    "github.com/yanyiwu/gosimhash"
    "github.com/astaxie/beego"
    "application/models"
)

type (
    idHash struct {
        contentId     string
        leftSimHash string
    }
)

var (
    sMux                    sync.RWMutex
    contentSimHashMap = make(map[string]string, 10000)
    simHashBlock1         = make(map[string][]idHash, 10000)
    simHashBlock2         = make(map[string][]idHash, 10000)
    simHashBlock3         = make(map[string][]idHash, 10000)
    simHashBlock4         = make(map[string][]idHash, 10000)
)
```

```
const (
    CONTENT_SIMHASH_PAGESIZE = 1000
)

func init() {
    go generateContentSimHash()
    go getIncrementalContentSimHash()
}

type SimHash struct {
    AbstractController
}

func (s *SimHash) Post() {
    content := s.GetString("content")
    contentId := s.GetString("contentId")
    result := s.PostHandler(content, contentId)
    iRet, _ := result["iRet"].(int)
    sMsg, _ := result["sMsg"].(error)
    s.WriteResponseJson(iRet, sMsg, result["jData"])
}

func (s *SimHash) PostHandler(content string, contentId string) map[string]interface{} {
    fingerPrint, err := s.getSimHash(content)
    if err != nil {
        //s.WriteResponseJson(models.TS_GET_SIMHASH_ERR, err, nil)
        return map[string]interface{}{
            "iRet":    models.TS_GET_SIMHASH_ERR,
            "sMsg":    err,
            "jData": nil,
        }
    }
    currentFp := fmt.Sprintf("%x", fingerPrint)

    go models.UpdateContentSimHash(contentId, currentFp)
    isSimilar, similarContentId := s.isSimilarBlock(currentFp, contentId)
```

```
    return map[string]interface{}{
        "iRet": 0,
        "sMsg": models.ERR_OK,
        "jData": map[string]interface{}{
            "fp":              fmt.Sprintf("%x", fingerPrint),
            "isSimilar": isSimilar,
            "contentId": similarContentId,
        },
    }
}

/*
** 获取 simHash（局部敏感 hash）
 */
func (s *SimHash) getSimHash(content string) (uint64, error) {
    _, fullFilename, _, ok := runtime.Caller(0)
    if !ok {
        return 0, errors.New("get fileName fail")
    }
    var baseDir string
    ip := commlib.GetLocalIP()
    baseDir = path.Join(path.Dir(fullFilename), "../conf/dict/")
    hasher := gosimhash.New(path.Join(baseDir, "jieba.dict.utf8"), path.Join(baseDir,
"hmm_model.utf8"), path.Join(baseDir, "idf.utf8"), path.Join(baseDir, "stop_words.utf8"))
    defer hasher.Free()
    fingerprint := hasher.MakeSimhash(content, 5)
    return fingerprint, nil
}

func (s *SimHash) similarScan(simHash uint64, contentId string) (bool, string) {
    sMux.RLock()
    for k, v := range contentSimHashMap {
        simHashB, _ := strconv.ParseUint(v, 16, 64)
        if contentId != k && s.isSimilar(simHash, uint64(simHashB)) {
            return true, k
```

```
            }
        }
        sMux.RUnlock()
        return false, ""
}

/*
** 计算 hamming distance，一般认为《=3 的为相似
 */
func (s *SimHash) isSimilar(simHashA, simHashB uint64) bool {
        cnt := 0
        n := 3
        simHashA ^= simHashB
        for simHashA != 0 && cnt <= n {
            simHashA &= simHashA - 1
            cnt++
        }
        if cnt <= n {
            return true
        }
        return false
}

/**
 * 根据抽屉理论，判断 simhash 是否相似，计算 hamming distance，一般认为《=3 的为相似
 */
func (s *SimHash) isSimilarBlock(simHash string, contentId string) (bool, string) {
        sMux.RLock()
        simHashA, _ := strconv.ParseUint(simHash, 16, 64)

        if v, ok := simHashBlock1[simHash[0:4]]; ok {
            for _, v1 := range v {
                simHashB, _ := strconv.ParseUint(v1.leftSimHash, 16, 64)
                if contentId != v1.contentId && s.isSimilar(simHashA, simHashB) {
```

```
                return true, v1.contentId
            }
        }
        return false, ""
    }

    if v, ok := simHashBlock2[simHash[4:8]]; ok {
        for _, v2 := range v {
            simHashB, _ := strconv.ParseUint(v2.leftSimHash, 16, 64)
            if contentId != v2.contentId && s.isSimilar(simHashA, simHashB) {
                return true, v2.contentId
            }
        }
        return false, ""
    }

    if v, ok := simHashBlock3[simHash[8:12]]; ok {
        for _, v3 := range v {
            simHashB, _ := strconv.ParseUint(v3.leftSimHash, 16, 64)
            if contentId != v3.contentId && s.isSimilar(simHashA, simHashB) {
                return true, v3.contentId
            }
        }
        return false, ""
    }

    if v, ok := simHashBlock4[simHash[12:]]; ok {
        for _, v4 := range v {
            simHashB, _ := strconv.ParseUint(v4.leftSimHash, 16, 64)
            if contentId != v4.contentId && s.isSimilar(simHashA, simHashB) {
                return true, v4.contentId
            }
            return false, ""
        }
    }
```

```
    sMux.RUnlock()
    return false, ""
}

/*
** 全量获取帖子 id 以及 simHash 值
 */
func generateContentSimHash() {
    countResult, err := models.GetContentTextCount(1)
    if err != nil {
        beego.Error("get cms text category count failed:", err)
        return
    }
    count, _ := strconv.ParseInt(countResult[0]["count"], 10, 64)
    pageCount := count / CONTENT_SIMHASH_PAGESIZE
    var i int64 = 0
    for ; i <= pageCount; i++ {
        arr, err := models.GetCmsContentSimHash(1,
i*CONTENT_SIMHASH_PAGESIZE, CONTENT_SIMHASH_PAGESIZE)
        if err != nil {
            beego.Error("get cms data failed:", err)
            continue
        }

        //改为抽屉理论的方式存储，分为 4 个区块
        makeBlock(arr)
        time.Sleep(time.Second)
    }
}

/*
** 增量获取帖子 id 以及 simHash 值
 */
func getIncrementalContentSimHash() {
```

```
    incrementalInterval, _ :=
beego.AppConfig.Int("simhash_metric::incremental_interval")
    ticker := time.NewTicker(time.Hour * time.Duration(incrementalInterval))
    for range ticker.C {
        result, err := models.GetIncrementalCmsContentSimHash(1,
time.Now().Add(-time.Hour * time.Duration(incrementalInterval)).Format("2006-01-02
15:04:05"))
        if err != nil {
            beego.Error("get incremental cms content simHash failed: ", err)
            continue
        }
        makeBlock(result)
    }
}

/**
 * 将 simHash 分成 4 个区块
 */
func makeBlock(arr []map[string]string) {
    sMux.Lock()
    for _, v := range arr {
        if len(v) == 16 {
            if _, isExist := simHashBlock1[v["sParam3"][0:4]]; !isExist {
                simHashBlock1[v["sParam3"][0:4]] = []idHash{}
            }
            simHashBlock1[v["sParam3"][0:4]] =
append(simHashBlock1[v["sParam3"][0:4]], idHash{
                contentId:     v["iContentId"],
                leftSimHash: v["sParam3"][4:],
            })

            if _, isExist := simHashBlock2[v["sParam3"][4:8]]; !isExist {
                simHashBlock2[v["sParam3"][4:8]] = []idHash{}
            }
```

```
            simHashBlock2[v["sParam3"][4:8]] =
append(simHashBlock2[v["sParam3"][4:8]], idHash{
                contentId:     v["iContentId"],
                leftSimHash: v["sParam3"][0:4] + v["sParam3"][8:],
            })

            if _, isExist := simHashBlock3[v["sParam3"][8:12]]; !isExist {
                simHashBlock3[v["sParam3"][8:12]] = []idHash{}
            }
            simHashBlock3[v["sParam3"][8:12]] =
append(simHashBlock3[v["sParam3"][8:12]], idHash{
                contentId:     v["iContentId"],
                leftSimHash: v["sParam3"][0:8] + v["sParam3"][12:],
            })

            if _, isExist := simHashBlock4[v["sParam3"][12:]]; !isExist {
                simHashBlock4[v["sParam3"][12:]] = []idHash{}
            }
            simHashBlock4[v["sParam3"][12:]] =
append(simHashBlock4[v["sParam3"][12:]], idHash{
                contentId:     v["iContentId"],
                leftSimHash: v["sParam3"][0:12],
            })
        }
    }
    sMux.Unlock()
}
```

附录 B 网络爬虫爬取舆情 MYSQL 表数据结构

```
CREATE TABLE `post` (
    `id` BIGINT(12) unsigned NOT NULL COMMENT '自增 id',
    `stock_id` int(12) unsigned NOT NULL COMMENT '股票编号',
    `date` date NOT NULL COMMENT '日期',
```

```
    `post_id` BIGINT unsigned NOT NULL COMMENT '帖子 id',
    `post_title` varchar(255) COMMENT '帖子标题',
    `post_content` blob COMMENT '帖子内容',
    `post_time` datetime COMMENT '帖子发表时间',
    `is_hot` tinyint(3) unsigned COMMENT '是否热门帖',
    `watch_count` bigint unsigned COMMENT '浏览数',
    `reply_count` bigint unsigned COMMENT '回复数',
    `url` varchar(255) COMMENT '帖子链接',
    `user_id` varchar(255) COMMENT '用户 id',
    `user_nickname` varchar(255) COMMENT '用户昵称',
    `source_id` tinyint(3) unsigned COMMENT '帖子来源 id',
    PRIMARY KEY (`id`) USING BTREE
) ENGINE=InnoDB DEFAULT CHARSET=gbk COMMENT='帖子信息表'
```

附录 C 基于 LSTM 深度神经网络的股票 ATR 预测的核心代码

```
from keras.models import Sequential
from keras.layers import Dense, Dropout, LSTM
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pylab import rcParams
from sklearn.preprocessing import MinMaxScaler

rcParams['figure.figsize'] = 20, 10
df = pd.read_csv('stock.csv')
stock_data = df.sort_index(ascending=True, axis=0)
stock_opinion_data = pd.DataFrame(index=range(0, len(df)), columns=['Date', 'atr'])

for i in range(0, len(stock_data)):
    stock_opinion_data['Date'][i] = stock_data['Date'][i]
    stock_opinion_data['atr'][i] = stock_data['Close'][i]

stock_opinion_data.index = stock_opinion_data.Date
stock_opinion_data.drop('Date', axis=1, inplace=True)
```

```
dataset = stock_opinion_data.values
train = dataset[0:987, :]
valid = dataset[987:, :]

scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(dataset)

x_train, y_train = [], []

for i in range(60, len(train)):
    x_train.append(scaled_data[i - 60:i, 0])
    y_train.append(scaled_data[i, 0])

x_train, y_train = np.array(x_train), np.array(y_train)
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))

model = Sequential()
model.add(LSTM(units=50, return_sequences=True, input_shape=(x_train.shape[1], 1)))
model.add(LSTM(units=50))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(x_train, y_train, epochs=1, batch_size=1, verbose=2)
inputs = stock_opinion_data[len(stock_opinion_data) - len(valid) - 60:].values
inputs = inputs.reshape(-1, 1)
inputs = scaler.transform(inputs)
X_test = []
for i in range(60, inputs.shape[0]):
    X_test.append(inputs[i - 60:i, 0])
X_test = np.array(X_test)
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
closing_price = model.predict(X_test)
closing_price = scaler.inverse_transform(closing_price)
rms = np.sqrt(np.mean(np.power((valid - closing_price), 2)))
train = stock_opinion_data[:250]
valid = stock_opinion_data[250:]
```

```
valid['Predictions'] = closing_price
plt.plot(train['atr'])
plt.plot(valid[['atr', 'Predictions']])
plt.show()
```

致谢

人生是一个不断探索、寻找的过程。在工作四年之后，我选择跳出自己的舒适区及以往所学的专业领域，继续探索新的未知领域的知识，超越自己，活出自己的人生价值，是我的最大的目标与追求。如何使自己理工科的思维和商学的思维进行结合，在刚开始学习的时候我一直是非常迷茫的。非常庆幸的是，我遇到了一位能够让我受益终生的良师益友--陈艳教授。两年多的学习和研究生涯并不算漫长，但在理论科研盲目探索时，在对生活的茫然无措时，有陈艳教授教导着我。使我对对生活满怀希望，对人生充满憧憬，能够在科学探索道理上继续执着前进。

本论文的工作是我在导师陈艳教授的耐心指导下所完成的。导师科学的工作方法和严谨的治学态度都给我留下了深刻的印象，对我的论文写作起到了很大的帮助作用。在论文写作和探讨过程中，老师看待问题的思路、独特的视角、严密的逻辑，总是发人深省悟。老师敏锐的学术洞察力、严谨的治学态度和对科研始终如一的精神让我肃然起敬。在此，衷心感谢老师这几年来对我的指导和关心。

在此还要感谢导师门下同届的共同奋斗的各位师兄弟们。几年来，我们一起做学问，一起探讨课题，探讨人生。在相互交流中，所绽放出来的点点闪光就是我论文写作的创新点。大家可能是来自不同专业领域，但每个人看问题的不同的视角，不同的经验积累，都让我拓展了视野，少走了许多弯路。

在这几年的求学路上，父母长辈们也一直支持着我，鼓励着我，为我所取得的进步高兴，也为我所遇到的困难担忧。感谢我的长辈们，他们以丰富的人生阅历和已经对事物的充分认知，深深影响着我的世界观、人生观和价值观，也间接地促进了我的学术观点和学术研究！也同时感激我同辈的兄弟们、姐妹们，你们对我无条件的理解、包容、鼓励和支持。

值此，谨向曾经关心和支持过我的老师、父母、朋友、同学致以最衷心的感谢和最深厚的祝福！