四川农业大学

本科毕业论文（设计）

（ 2023 届）

题　　目：基于在线评论的商品情感倾向研究

学　　院：商旅学院

专　　业：电子商务

学生姓名：林岸　　学　　201709578

导　　师：彭卫　　职　　副教授

完成日期：　　年　月　日

论文原创性声明

本人郑重声明：所呈交的学位论文是我个人在导师指导下进行研究工作所取得的成果。尽我所知，除了文中特别加以标注和致谢的地方外，学位论文中不包含其他个人或集体已经发表或撰写过的研究成果，也不包含为获得四川农业大学或其它教育机构的学位或证书所使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

本科生签名：林岩　　年　月　日

论文版权使用授权书

本人完全了解四川农业大学有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。同意四川农业大学可以用不同方式在不同媒体上发表、传播学位论文的全部或部分内容。

□ 本论文延迟____年后公开，到期后适用本授权。

（限制级及涉密学位论文请在□内划“√”，并填写延迟公开时限。不勾选此项，默认为公开）

本科生签名：林岩　　年　月　日

导师签名：　　年　月　日

目　录

基于在线评论的商品情感倾向研究

专业：电子商务　姓名：林岸
导师：彭卫　副教授

摘要：随着互联网技术的发展，越来越多的消费者依赖在线评论来获取商品信息和评价。本研究旨在探讨如何通过在线评论数据，以研究用户情感倾向为目标，提高用户满意度。为了实现这一目标，本文以电商平台的在线评论作为基本数据集合，提出了一种基于 LDA 主题模型和 HowNet 情感词典的优化商品词条撰写的方法。首先，使用爬取的电商评论数据集合进行预处理；其次，将预处理后的数据输入到 LDA 主题模型中对在线评论进行主题及主题下关键词抽取，将抽取后不同主题下的关键词利用 HowNet 情感词典，进行极性判别，情感分析，权重赋予；在最后阶段，利用 HowNet 情感词典对各个评价维度的看法进行综合评分计算，从而对三种不同品类的电子产品在各维度的整体评价进行对比。经过与普通线性模型的比较分析，证明了该方法的可行性。该方法将消费者关注的主题与情感需求相结合，以更贴近消费者的心理预期。经过优化的商品不仅能更好地满足消费者的需求，还能提高用户满意度。

关键词：在线评论;LDA 主题模型;情感词典;情感分析

Research on Commodity Sentiment Tendency Based on Online Reviews

An Lin

Abstract: With the development of Internet technology, more and more consumers rely on online reviews to obtain product information and evaluation. This study aims to explore how to use online review data to study user emotional tendencies and improve user satisfaction. In order to achieve this goal, this paper takes the online

reviews of the e-commerce platform as the basic data set, and proposes a method based on the LDA topic model and the HowNet sentiment dictionary to optimize the writing of commodity entries. First, use the crawled e-commerce review data set for preprocessing; second, input the preprocessed data into the LDA topic model to extract the topics and keywords under the topics of online reviews, and extract keywords under different topics Using the HowNet Sentiment Dictionary, conduct polarity discrimination, sentiment analysis, and weight assignment; In the final stage, the HowNet sentiment dictionary is used to calculate the comprehensive score calculation of the opinions of each evaluation dimension, so as to compare the overall evaluation of three different categories of electronic products in each dimension. The feasibility of this method is proved by the comparative analysis with the ordinary linear model. The validity of the method is verified by crawling the online reviews of Jingdong platform. This method combines the topics that consumers pay attention to with emotional needs, so as to be closer to consumers' psychological expectations. Enhanced products are capable of not only addressing consumer requirements more effectively, but also elevating user contentment.

Key words: online reviews; LDA topic model; sentiment dictionary; sentiment analysis

1 绪论

1.1 研究背景

近年来，随着互联网技术的迅速发展和智能手机的普及，电子商务已成为人们日常生活中不可或缺的一部分[1]。由于人口基数的巨大，中国自然也是全球最大的电子商务市场，该行业在中国的蓬勃发展尤为明显。随着消费者对在线购物的依赖日益加深，基于在线评论考虑用户满意度的商品词条撰写优化研究显得尤为重要[2]。中国电子商务市场近年来呈现出快速增长的趋势。众多电商平台的崛起，如阿里巴巴、京东、拼多多等，使得消费者可以足不出户便能购买到各种商

品。智能手机的普及使得在线购物变得更加便捷，消费者可以随时随地挑选商品、查看评价和下单。此外，电子商务的发展还带动了快递行业、支付行业等相关产业的繁荣，整个产业链得到了更好的推动。然而，电子商务的迅猛发展背后也暴露出一些问题。首先，商品信息的真实性和准确性成为消费者关注的焦点[3]。在竞争激烈的市场环境下，一些商家可能会使用夸张的广告语、虚假的评论等手段来吸引消费者。这不仅损害了消费者的权益，也影响了整个行业的信誉。其次，大量在线评论使消费者在挑选商品时面临信息过载的问题。如何从众多评论中筛选出有价值的信息，成为消费者在购物决策过程中的一大难题。

为了解决上述问题，本研究以用户满意度为核心，探讨如何通过优化商品词条撰写来提高基于在线评论的用户满意度。首先，通过对大量在线评论进行分析，筛选出消费者关注的主题和需求。接下来，通过情感分析技术对评论进行情感倾向评估，从而了解顾客对产品的满意度。最终，根据顾客需求和情感偏好，对产品词条进行调整和优化，使其更符合消费者的心理期望，从而提升用户满意程度。本研究旨在为电子商务行业提供一种新颖的商品词条撰写优化方法。通过分析在线评论，深入挖掘消费者的需求和关注点，以用户满意度为导向对商品词条进行优化[4]。这样的方法不仅有助于提高消费者在购物过程中的满意度，还能为商家提供有针对性的改进建议，进一步提升商家在电子商务平台的竞争力。

1.2 研究意义

在当前电子商务市场竞争激烈的背景下，提高用户满意度已成为商家关注的核心问题。本研究通过分析在线评论、挖掘消费者需求并优化商品词条撰写，从而提高用户满意度，具有重要的理论意义和现实意义。

（1）理论意义

丰富和拓展了电子商务领域的研究内容。本研究从商品词条撰写的角度出发，关注基于在线评论的用户满意度，为电子商务领域的研究提供了新的视角和思路。这有助于拓宽电子商务领域的研究领域，丰富相关理论体系。此外，还提供了一种新的商品词条优化方法，基于在线评论考虑用户满意度，提出了一种商品词条撰写优化方法。这一方法将消费者关注的主题与情感需求相结合，为商品词条优化提供了新的理论支持。同时，这一方法有助于理解消费者在购物过程中的需求

和情感倾向，为后续研究提供了理论基础。

（2）现实意义

提高电子商务平台的用户满意度。通过本研究提出的商品词条撰写优化方法，商家可以更准确地把握消费者的需求和情感倾向，从而提高用户在购物过程中的满意度。高用户满意度将有助于提升消费者对电子商务平台的忠诚度，进一步提高平台的竞争力。为商家提供有针对性的改进建议。通过分析在线评论，本研究可以帮助商家深入了解消费者的需求和关注点，从而为商家提供有针对性的改进措施。这不仅有助于提高商品的吸引力，还能促使商家在经营过程中不断优化和提升，最终实现可持续发展。促进诚信经营，提升行业整体信誉。本研究关注商品词条的真实性和准确性，有助于提高商家对诚信经营的重视。当商家更加注重商品信息的真实性和准确性时，将有助于提高整个电子商务行业的信誉，为消费者提供更为可靠的购物环境。

1.3 研究内容

本研究以在线商品评论为基础，目的是通过分析顾客在评论中所表达的信息来评估他们对电子产品行业提供的商品的态度，从而为产品或服务的改进提供依据和建议。本研究主要关注商品特征提取和满意度应用的深入探讨。商品具有多个方面的特征，如“品质”、“售后”等，因此本研究旨在开发一种从在线评论中提取影响顾客的商品特征的方法，以便更细致地分析影响顾客满意度的因素。在传统研究方法中，通过计算顾客对各种商品特征的情感倾向，可以了解顾客对商品的满意程度。本研究将情感倾向更加细分例如加入了：正负面评价词语，程度副词，以进一步拓展基于在线评论研究的应用领域。本研究主要工作如下图 1 所示：

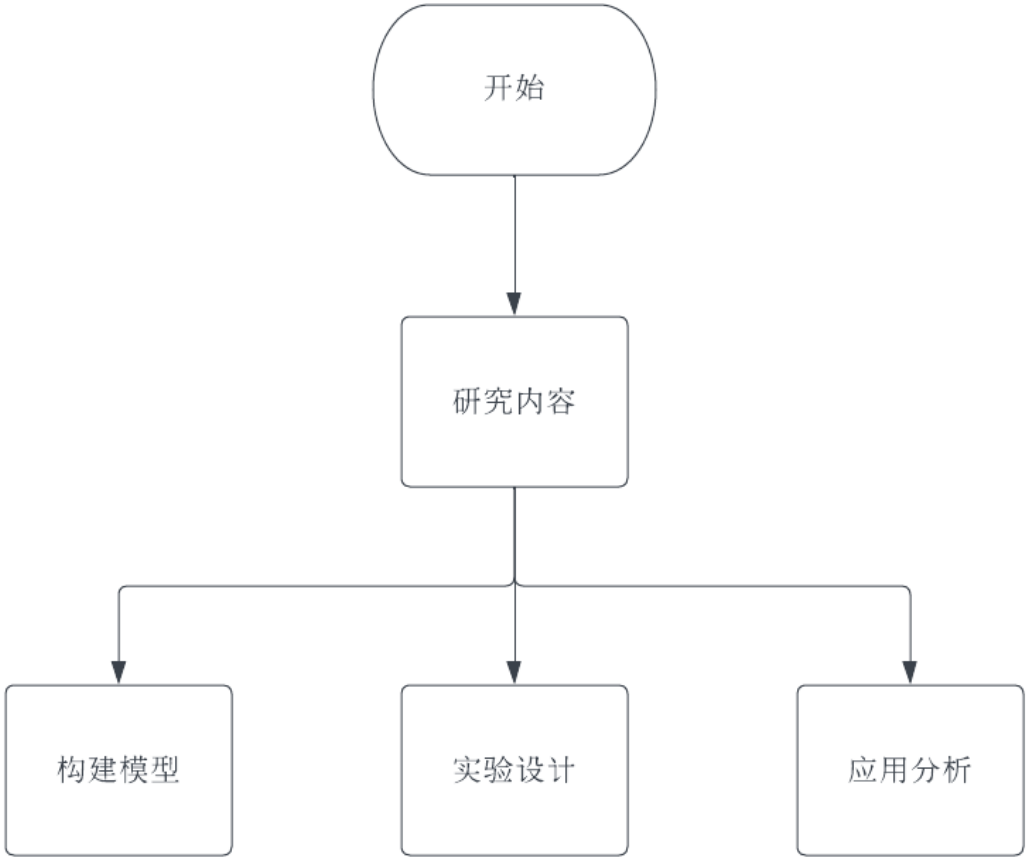

图 1 主要研究内容

（1）模型构建

通过大量阅读相关文献，了解各种提取方法的优劣，最后选取利用情感词典判别情感倾向作为本文的研究重点。该方法的模型通过预处理获得无标点，无停用词以及其他与试验结果不相关的因素的能够输入模型的数据，通过 LDA 主题模型生成相应主题与关键词，在通过 HowNet 情感词典设置权重，设置参数后，通过相应的编程语言以及编程语言中现有的模块实现模型。

（2）实验设置

在京东电子商务平台的电子产品区域，运用网络爬虫搜集评论数据，并通过连续试验来确认最后的模型参数。此外，构建专门的情感词汇库，对经处理的评论进行情感剖析，紧接着为各项数据分配权重并进行深入分析。

（3）应用探讨

对模型处理后的成果与实际情况相融合，进行深入分析，使得结果更接近实际应用需求。这有助于协助商家提升产品和服务质量，从而进一步增强用户忠实度、黏着度以及客户满意程度。

1.4 研究方法

（1）文献梳理方法

在中国知网和万方数据库中，利用“网络评论”、“LDA 主题建模”、“属性抽取”和“情感评估”作为关键词，查阅相关文献并对现有研究进行概括和理解。关注半监督机器学习、文本分类、情感评估方法以及顾客满意度相关领域的研究发展。

（2）数据采集策略

通过网络爬虫工具获取京东电子商务平台在线评论作为数据源，并对数据执行去除重复、降低噪声、清除停用词等预处理任务，以使处理后的词语更符合模型标准。

（3）数值分析方法

对处理后的数据进行数值分析，得出评论的情感评估结果。量化客户对商品各方面的满意程度，为后续利用情感评估结果作好铺垫。

（4）试验方法

针对涉及多个相关参数的模型，开展一系列实验，以确定最优模型参数并确保模型性能达到最佳状态。

1.5 论文结构

论文将用五个部分对研究的背景、领域现状、挖掘测度方法、实验分析和结论五个部分进行阐述。如下图 2 所示

第一章是绪论，引出本文研究的内容，其中包括：研究背景、研究意义、研究内容和研究方法。

第二章将通过综述相关文献的研究方法、结论和影响等内容，分别界定相关概念，总结归纳相关领域的研究进展，并陈述本文选择相关方法与理论的原因。

第三章给出我们的模型，而后通过公式、模型架构图等方式陈述实验的具体步骤，包括数据预处理阶段的各项任务及其原因，相关模型的设计、训练、测试

等。

第四章将按照前期实验设计的步骤，首先介绍实验过程中用到的数据集及其规格，而后展现评论文本情感极性的实验结果，并对其他实验数据进行预测，分析消费者的评论情感倾向。最后描述基于实体的情感分析结果，预测评论情感倾向。

第五章会基于实验结果分析消费者存在这些偏好的原因，引用相关理论进行解释，并提出对应的策略，将总结文章的结论，陈述研究的创新点与相关贡献，分析不足之处并引出未来研究的局限以及未来的改进方向。

图 2 论文组织架构

2 文献综述

2.1 有关 LDA 主题模型的分析方法

潜在狄利克雷分布主题模型简称：LDA 主题模型。它在 GPT 文本生成以及分析领域中得到了许多应用。LDA 主题模型是一种无监督的生成概率模型，可以自动地从大规模文本集合中挖掘出潜在主题，并为每篇文档分配主题概率分布[5]。首先，LDA 模型在社交媒体文本分析中表现出较好的性能。例如，Nguyen 等人[6]应用 LDA 模型对推特数据进行主题建模，有效地发现了与特定事件相关的热点话题。这一方法可以帮助研究者和决策者了解社交媒体上的舆论动态，为政策制定提供数据支持。其次，潜在狄利克雷主题分布模型在情感分析中获得了不少成果。Habbat 等人[7]结合 LDA 主题模型和情感词典对在线评论进行情感分析，提高了情感分类的准确性。通过挖掘评论中的潜在主题，可以更好地理解消费者对商品或服务的看法和需求。此外，LDA 模型还在多领域知识融合中发挥了重要作用。Huang 等人[8]提出了一种基于 LDA 模型的跨领域主题融合方法，用于实现知识的有效迁移。这种方法可以为多领域学习提供新的思路，提高模型的泛化能力。

2.2 文本情感分析方法

赵妍妍等人认为[9]:文本情感分析是指通过自然语言处理技术(Natural Language Processing)对带有主观性色彩的文本进行情感挖掘，主要包含情感信息抽取、情感信息分类和搜索等任务。在最近几年，情感分析已经成为自然语言处理领域的关注焦点。人们在社交媒体上分享日常生活、表达对社会事件的观点，微博和 Twitter 等平台已成为实时的公众舆论源。同时，购物网站上的商品评价为消费者选购产品提供了极大便利。对非结构化文本的情感倾向进行分析以解决特定问题对信息决策者具有重要意义。赵常煜认为[10]情感分析又称为情感倾向性分析或意见挖掘，是从用户意见中提取信息的过程通过对文本、音频和图像等进行分析以获取人们的观点、看法、态度 和情感等。钟佳娃等人认为[11]通过阅读相关文献和基于已有的研究，学者们将文本情感分析大志分为以下 4 种: 基于情感词典与规则的方法、基于传统机器学习的方法、基于深度学

习的方法和多策略混合的方法。

2.2.1 基于情感词典与规则的方法

基于情感词典与规则的方法在情感分析领域中得到了广泛的应用。这种方法主要包括两个方面：情感词典的构建和情感分析规则的设计。本文对该方法进行了文献梳理，旨在为相关研究提供理论依据。情感词库是情感分析的核心，主要涵盖情感词语、强度词语、否定词语等。情感词典的构建方法有很多，如基于语料库的方法、基于词义相似度的方法、基于知识图谱的方法等[12]。这些方法旨在挖掘词汇在不同情境下的情感倾向，从而为情感分析提供依据。如何构建高质量、全面的情感词典是研究者关注的重点。近年来，多语种、领域特定的情感词典逐渐受到关注[13]。情感分析规则的设计是基于情感词典与规则方法的另一个关键环节。通过合理的规则设计，研究者可以准确地抽取文本中的情感信息。规则设计主要包括词汇加权、情感传播、情感聚合等方面。词汇加权主要针对程度词汇、否定词汇等词汇对情感倾向的调整作用[14]；情感传播则关注情感词汇在句子和篇章层面的传播规律；情感聚合主要研究如何将抽取到的情感信息整合为一个统一的情感分数。规则设计的难点在于考虑多种因素的影响，如词汇间的关系、句子结构、篇章结构等。总的来说，基于情感词典与规则的方法在情感分析领域中具有较高的实用价值。这一方法关注情感词典的构建和情感分析规则的设计，旨在通过综合考虑词汇、句子、篇章等多个层面的信息，准确地抽取文本中的情感倾向。

2.2.2 基于机器学习的方法

在文本情感分析领域，基于传统机器学习的方法得到了广泛应用。这些方法通常包括支持向量机（SVM）、朴素贝叶斯（NB）、决策树（DT）等算法，它们在处理大规模、高维度的文本数据方面具有较好的性能。首先，基于特征选择的方法是传统机器学习在文本情感分析领域的一个关键应用。特征选择主要包括词袋模型（BOW）、词频-逆文档频率（TF-IDF）等方法，这些方法可以有效地提取文本中的关键信息，为后续的情感分类提供依据[15]。其次，基于模型融合的方法在情感分析中也得到了广泛关注。通过组合多个基于传统机器学习的分类器，研究者可以进一步提高情感分类的准确性[16]。一种常见的融合策略是集成学习方

法，如随机森林、AdaBoost 等。最终，依赖迁移学习的策略在文本情感分析领域也获得了一定程度的突破。这种方法主要借助源领域的知识为目标领域的情感分析提供帮助[17]。通过对源领域与目标领域数据进行适应性调整，迁移学习手段能够增强目标领域的分类表现。

2.2.3 基于深度学习神经网络的方法

近期，依托深度学习技术的文本情感分析在研究领域获得了显着的成果。这些技术一般包含但不限于卷积神经网络（CNN）、循环神经网络（RNN）和长短时记忆网络（LSTM）等算法，它们在处理文本数据时具有较高的表现力和准确性。首先，卷积神经网络（CNN）在文本情感分析领域得广泛应用。CNN 通过局部感知和权值共享的特性，能够自动学习文本中的局部特征，从而提高情感分类的准确性[18]。此外，循环神经网络（RNN）和长短时记忆网络（LSTM）同样在文本情感分析领域表现出较好的性能。它们可以有效地捕捉文本中的长距离依赖关系，提高情感分类的准确性。近期，预训练语言模型如 BERT（Bidirectional Encoder Representations from Transformers）和 GPT（Generative Pre-trained Transformer）在文本情感分析领域取得了重要突破。这些模型通过在大规模语料库上进行预训练，可以学习到丰富的语义知识，为情感分类提供强大的特征表示能力[19]。此外，基于多任务学习的方法在情感分析中也获得了广泛关注。通过并行学习多个类似的任务目标，模型的普遍泛化能力将得到一定程度的优化，随后能够在目标任务中实现性能提升。

3. 研究模型

本研究的核心内容为：LDA 主题模型抽取和情感分析，因此本研究将关注这两个方面的内容。下面简要描述了本文研究内容的框架流程，如图 3 所示。

1.设计 oython 程序提取生鲜电商平台的用户评论数据，并对数据执行预处理操作，包括：去除重复、分词、清除停用词等，从而提高原始数据品质

2.使用 LDA 主题模型构建数个基于评论的主题，并且生成 30 个在不同主题下强相关的关键词。

3. 依据 HowNet 情感词典创建领域情感词库，然后根据构建的词库进行文本中情感极性划分和情感倾向预测。

4. 根据前一步的情感极性划分结果进行权重分配，情感倾向趋势预测，最终得到模型输出结果。

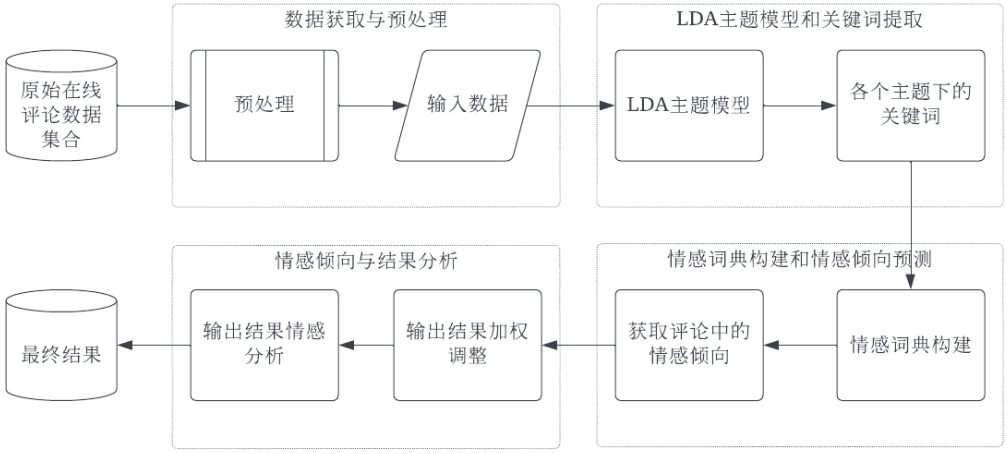

图 3 研究框架流程图

3.1 数据获取与预处理

顾客在电商平台购买商品后，会根据自身的消费售后体验撰写在线评价。通过设计 Python 动态爬虫程序，模拟用户操作界面，从而提取网页评论中对应评论的 HTML 程序部分，实现将用户评价在爬虫界面下批量自动采集。如下图 4 所示


```
[{"id":19018476212,"guid":"f51f401c57d81d4f6f5f3fa14126246f","content":"外形外观：手机外观时尚，尺寸大小中规中矩，到手握着也不费力\n屏幕音效：屏幕清晰，且不刺眼，6.1寸大小刚刚合适，音效也非常好，清晰浑厚没有杂音\n拍照效果：拍出来的效果就是实物，很清晰\n运行速度：丝滑流畅，没有卡顿\n待机时间：可以轻松玩一天","vcontent":"{$%&外形外观：&%$}手机外观时尚，尺寸大小中规中矩，到手握着也不费力\n{$%&屏幕音效：&%$}屏幕清晰，且不刺眼，6.1寸大小刚刚合适，音效也非常好，清晰浑厚没有杂音\n{$%&拍照效果：&%$}拍出来的效果就是实物，很清晰\n{$%&运行速度：&%$}丝滑流畅，没有卡顿\n{$%&待机时间：&%$}可以轻松玩一天","creationTime":"2023-04-13
```


图 4 爬虫界面下的客户评论

在线评论本质是一种非结构化的文本，根据爬取到的在线评论我们就可以进行预处理，以待进一步使用。内容包括，1.去重：去除重复的评论。2.分词：将句子划分为词语，以词语为单位对数据进行处理。3.去停用词：停用词是指文本中无实际意义的词、比如语气词以及一些连词或者一些标点，如“的”、

“啊”、“？”等，去除后节省空间且不影响数据信息。下图 5 是预处理后的评论文本数据，所生成有关京东平台上面手机条目下的词云。

图 5 手机评论词云

3.2 评论 LDA 主题提取与关键词提取

基于规则的商品属性提取方法需要大量的语法、句法等的先验知识；基于传统深度学习方法如 CNN、RNN 需要大量训练数据，同时 CNN 基于拓扑结构的数据处理方式与 RNN 基于序列结构的数据处理方式无法捕捉广泛的评论语义信息[20]。

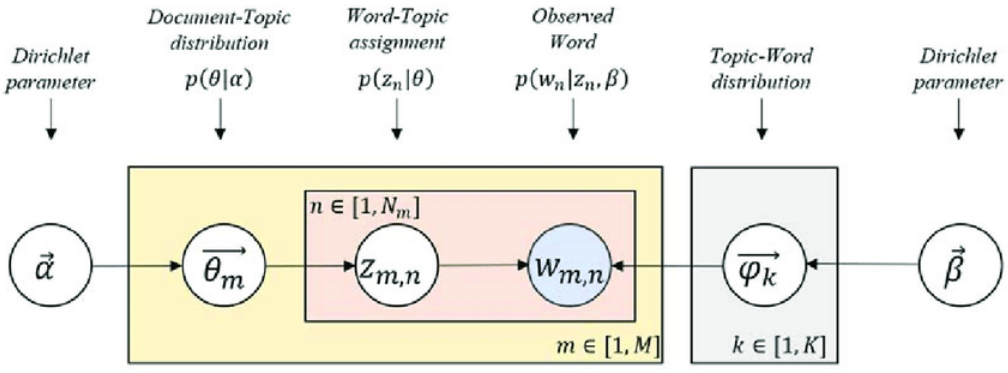

图 6 LDA 向量空间中工作原理及参数

进而本研究提出将从网络电子商务平台也就是京东平台获取的评论数据集合构建为 LDA 主题模型要求维数的数据格式，采用半监督的算法提取相应的主题及关键词。首先构建基于评论的 LDA 向量空间（Vector Space of LDA）如上图 6 所示。

表 1　LDA 向量空间中的参数

parameters	description
α	controls per-document topic distribution
β	controls per topic word distribution
M, m	is the total documents in the corpus
N, n	is the number of words in the document
w	is the word in a document
z	is the latent topic assigned to a word
θ	is the topic distribution
φ	is the topic-word distribution

LDA 主题模型中参数含义如上表 1 所示。随后构建了相应的 LDA 向量空间后，在代码块中设置相应的参数，就能够输出许多根据评论文本相应的主题。算法流程如下图 7 所示。

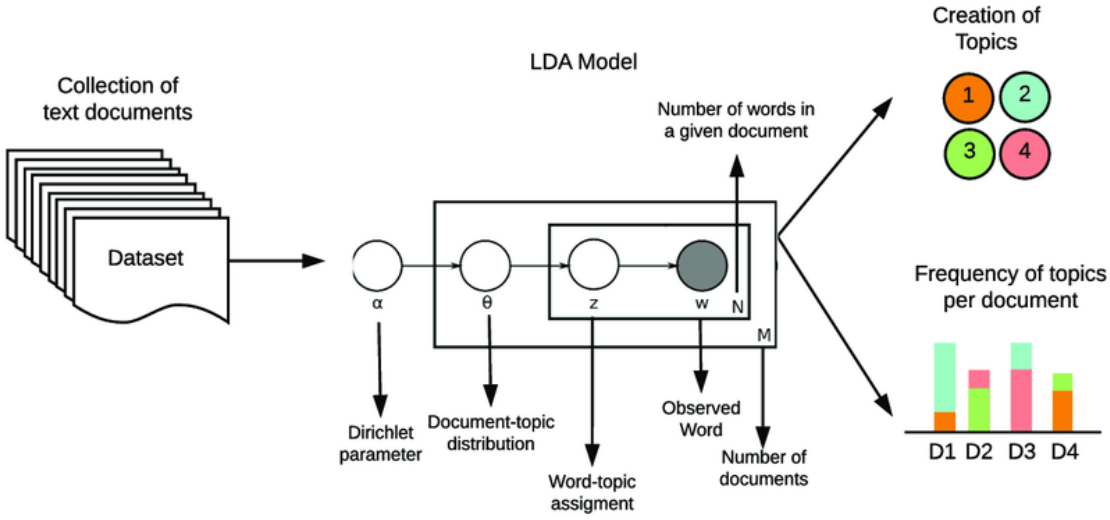

图 7 LDA 主题模型总流程

最后通过整个模型生成的主题结果以及主题下的关键词如下图 8 所示。（文本中对于相同单词有不同的文本指向符合实际情况，例如一个主题为夏天的文本和一个主题为冰淇淋的文本很可能出现相同的词语、比如“热”、“冰淇淋”）

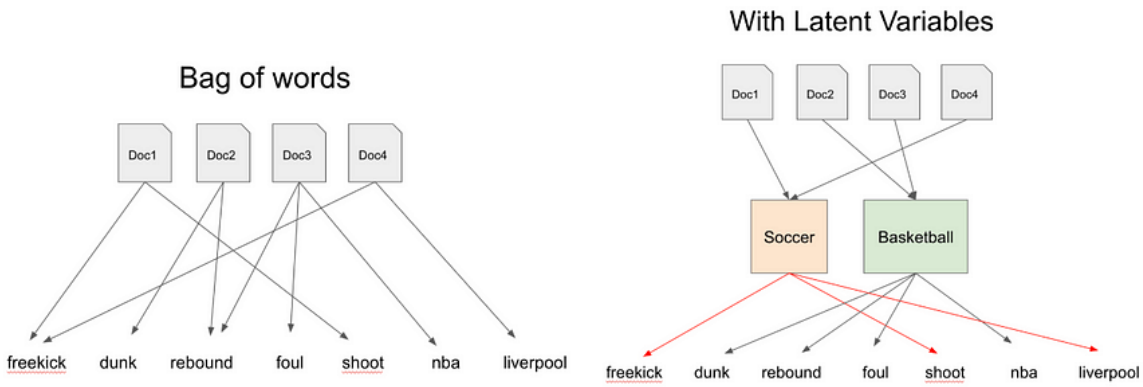

图 8 LDA 主题模型生成的结果

3.2.1 数据输入

因此本文获取通过预处理后的数据，首先将其转换为 CSV 格式，共计得到评论 10000 条。手机 2000 条、笔记本电脑 3000 条、显示器 5000 条。将 CSV 文件列为两列，第一列表示评论的序数，第二列表示评论的内容。读取全部数据集的文本数据构建为一个长度为 10000 的，纬度为 2 维，元素类别为字符串，类型为 python 中可变长度的列表。

3.2.2 主题困惑度(perplexity)

在 LDA（潜在狄利克雷分配）主题模型中，“Perplexity”（困惑度）是一种评估主题模型性能的指标[21]。困惑度衡量了模型对未知数据（例如测试集）的预测能力。数学上，困惑度可以解释为给定模型生成观察到的测试数据的概率的倒数[22]。换句话说，困惑度反映了模型对实际数据分布的逼近程度。一个好的主题模型应该能较准确地预测测试数据，因此具有较低的困惑度。但是困惑度并不是越低越好，因为随着主题数量的增加，虽然困惑度越来越低但是生成的模型往往会过拟合。然而，需要注意的是，困惑度并非唯一评估主题模型性能的指标，其他指标如主题一致性（Coherence）也在某些情况下被认为更具解释性[23]。因此，在评估和选择主题模型时，通常需要综合考虑多个指标。下图 9 所示为 iPhone14 评论

集合中用 LDA 主题模型拟合选取不同主题数目下的困惑度，此处选取主题数目等于 10 作为过拟合的阈值。

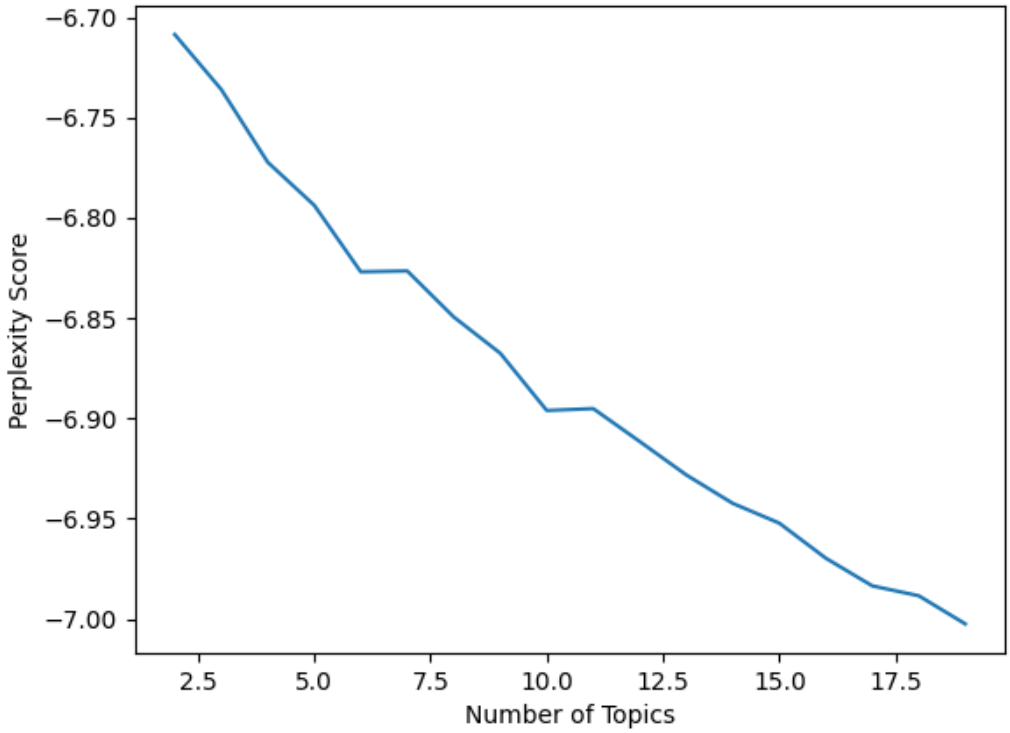

图 9 LDA 主题困惑度(Perplexity)折线图

3.2.3 主题一致性(coherence)

在 LDA（潜在狄利克雷分配）模型中，“Coherence”（主题一致性）是衡量主题模型性能的另一个重要指标。主题一致性衡量了一个主题内部的关键词在语义上的相关性，即一个主题内部关键词之间的紧密程度。较高的一致性表示主题内部的关键词更加紧密相关，表明该主题能更好地捕捉到实际文本中的语义结构[24]。

主题一致性与困惑度（Perplexity）这种基于概率分布的评价指标有所不同，它更侧重于评估主题的可解释性和可理解性[25]。在实际应用中，主题一致性可以作为补充困惑度的评价指标，帮助研究者和实践者更全面地了解主题模型的性能。下图 10 所示为手机评论集合中用 LDA 主题模型拟合选取不同主题数目下的一致性。

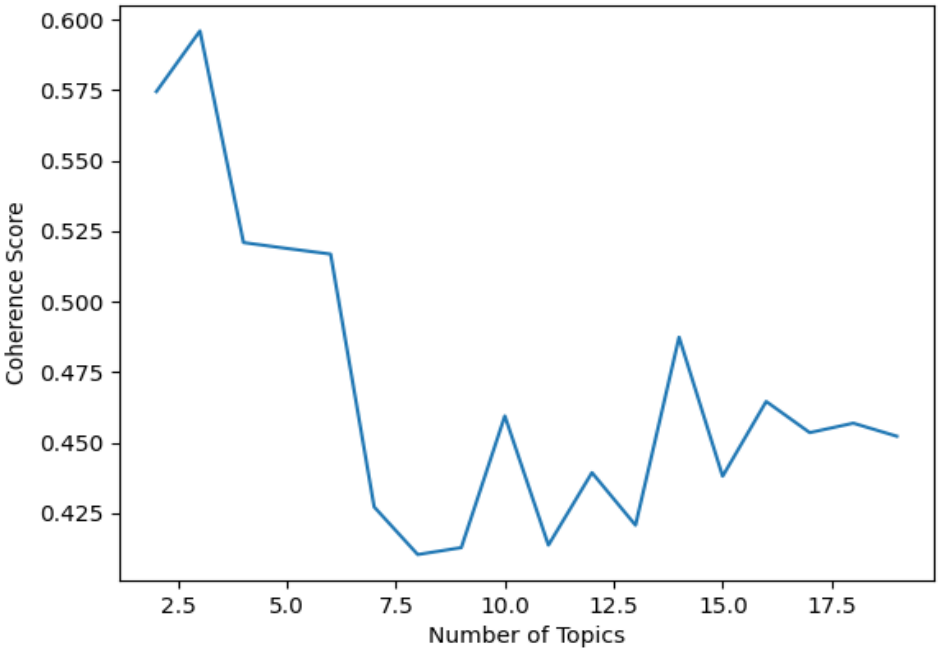

图 10 LDA 主题一致性(coherence)折线图

3.2.4 主题个数选取

根据困惑度和主题一致性的结果，选择一个综合表现最优的主题个数。常用的方法是绘制两个指标的折线图，观察它们的变化趋势。如上两图所示，综合困惑度范围以及在困惑度范围下一致性的最大值点的横坐标作为最终训练的主题数目。此处在手机条目下选取主题数目为 num_topics=3

3.2.5 关键词选取

根据选取最优的主题个数，通过 LDA 主题模型训练后的数据生成在不同主题下的 15-30 个关键词，可视化结果如下图 11 所示。参考过去的文献以及根据本文研究，所示 $\lambda = 0.6$ 时结果最为显著[26]

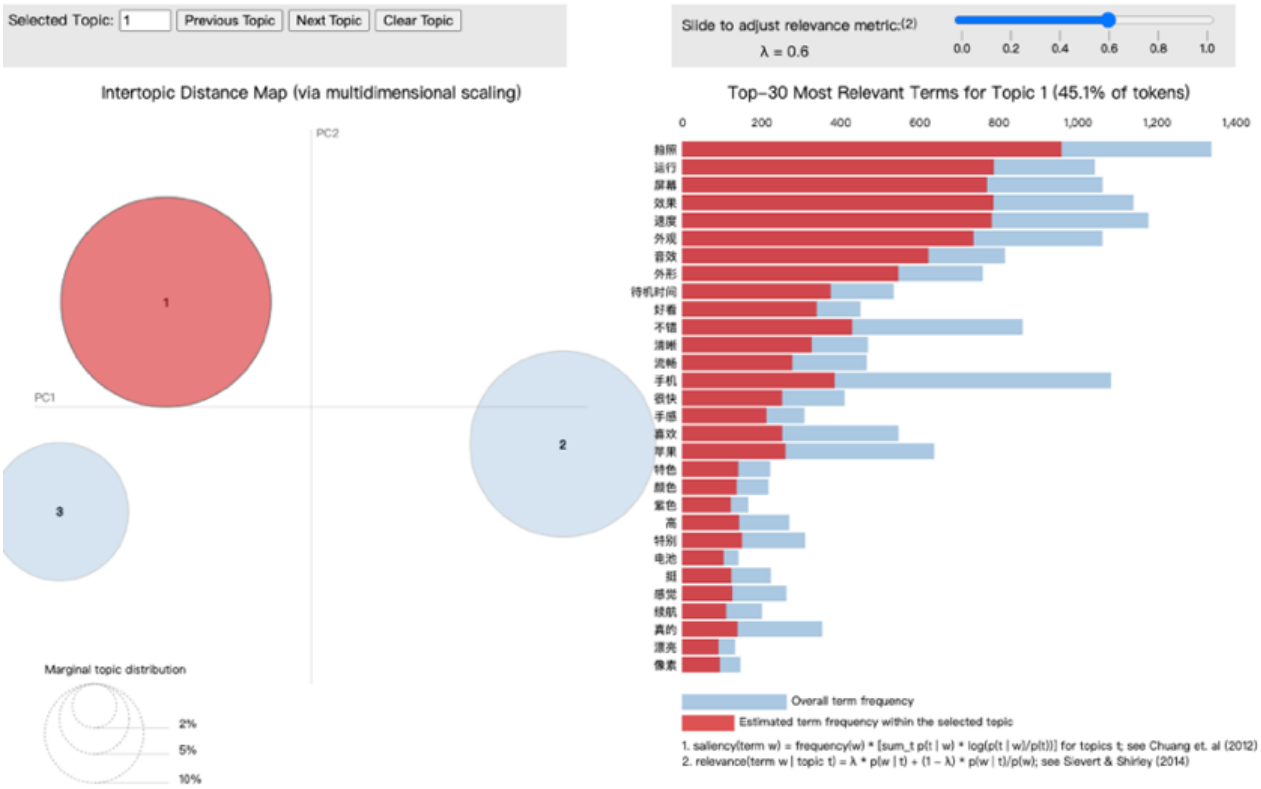

图 11 LDAvis 结果渲染图

3.3 领域情感词典构建

通过知网上所公开的文献下载 HowNet 情感词典，编码方式为 GB18030 其中 HowNet 情感词典主要包括：正负面评价词语、正负面情感词语、程度级别词语。其中，程度级别词语又包括：超(most)，非常(very)，较(more)，稍(ish)，不足（insufficiently）等不同程度的程度副词。如下表 2 和下表 3 所示。

表 2 正负面评价词语

正负面评价词	词组（仅展示 10 个）
正面	安，安生，安顿，安定，安分，安静，安康，安宁，安全
负面	不好，碍眼，碍难，暗，爱答不理，爱理不理，矮，昏，骗

表 3 程度副词词典及权重

程度级别	举例	分值
超	超，十足，超级，最，极，过，过度	2
非常	很，颇，颇为，实在，太，特，特别，非常	1.5
较	较，更，比较，比较严重，大量，几乎	1.25
稍	稍，略，略微，有点，偶尔，偶然，毫无，稍微，少许，些微	0.5
不足	半点，不大，不丁点儿，不甚，轻度，弱，丝毫，微，相对	0.25

3.4 评论情感极性判别

商品评价情感分析是通过在线评价来判断评价者表达的情感偏好。文本数据的情感分析技术已日渐完善，并在各行业中得到广泛应用。主要的情感判断技术包括基于情感词汇表、基于机器学习以及基于深度学习的方法。其中，基于情感词汇表的方法无需训练数据，可以直接对文本的情感偏好进行评估，具有低成本、高迁移性等优点。因此，本文通过情感词典的方法来对评论的情感倾向进行判别[18]。

本研究通过情感词典中的正负面评价词语来判断 LDA 模型训练后某个主题下所有有关的关键词所构成极性，具体判别方式如下。$keyword_i^N$ 表示生成的第 N 个主题下第 i 个关键词，其中 N 和 i 的取值范围为：$\{N = 1,2,\dots\dots num\_topics\}$ $\{i = 0,1,\dots\dots 29\}$； $counter_{pos}$表示积极评价的个数，其取值范围为$\{counter_{pos} = 1,\dots\dots 30\}$；$counter_{neg}$表示消极评价的个数，其取值范围为$\{counter_{neg} = 1,\dots\dots 30\}$。通过遍历如上图领域词典中的正负评价词语 posElement，negElement 其中 posElement, negElement 取值为：(0, 正面评价词语的个数-1)和(0, 负面评价词语的个数-1)得到$counter_{pos}$以及$counter_{neg}$。通过如下公式获得每个主题的分数

$$counter_{pos} = counter_{pos} + 1,\ if\ keyword_i^N = \text{posElement} \quad (1)$$

$$counter_{neg} = counter_{neg} + 1,\ if\ keyword_i^N = \text{negElement} \quad (2)$$

$$score = \begin{cases} 2, & if \quad counter_{pos} > counter_{neg} \\ 0, & if \quad counter_{pos} = counter_{neg} \\ -2, & if \quad counter_{pos} < counter_{neg} \end{cases} \quad (3)$$

本研究通过以上几个公式就得到某个主题下情感极性，其中大于零代表积极，等于零代表中性，小于零代表消极。在此基础上，我们利用正负情感词典对该主题下的加强情感进行判别。我们首先引入一个均衡函数$f_{balance}(x)$如下面公式所示。其中$counter_{emo} = counter_{emo_{pos}}, counter_{emo_{neg}}$，其中$counter_{emo_{pos}}$及$counter_{emo_{neg}}$的取值范围为{value = 1,2, … … ,30}。通过遍历如上图领域词典中的正负情感词语 posEmoElement，negEmoElement 其中 posEmoElement, negEmoElement 取值为(0, 正面情感词语的个数-1)和(0, 负面情感词语的个数-1)得到$counter_{pos}$以及$counter_{neg}$。根据季旺等人的研究，log(x)函数作为分母同时引入调节参数对于函数设计具有很显著的效果[27]，所以本研究借鉴季旺等人的研究通过设计如下公式得到加强情感分数$score_{emo}$，δ为均衡因子，其范围大致为(0, 1]其作用有：防止分母为零或者防止结果过大结果出现异常，保持均衡函数$f_{balance}(x)$在其定义域内为一直为缓慢单调递增函数以便符合实际要求。

$$f_{balance}(x) = \frac{x}{ln(x+\delta)}, \quad x = counter_{emo} \quad if \quad counter_{emo} \quad is \quad not \quad null \quad (4)$$

$$score_{emo} = \begin{cases} score + \frac{1}{2} \times f_{balance}(x) - \frac{1}{2} \times f_{balance}(x), & x = counter_{emo_{pos}} \quad if \quad score \geq 0 \\ score - \frac{1}{2} \times f_{balance}(x) + \frac{1}{2} \times f_{balance}(x), & x = counter_{emo_{neg}} \quad if \quad score < 0 \end{cases} \quad (5)$$

最后我们引入程度副词情感词典，由于程度副词词典数量较少且部分还与停用词集合有交集，所以很可能在通过 5 个不同的计数器($counter_{most}$, $counter_{very}$, $counter_{more}$, $counter_{ish}$, $counter_{insuff}$)分别记录 5 个不同程度的程度副词后，5 个记数器都为 0 个。所以我们简单设计如下法则：

$$score_{final} = \begin{cases} score_{emo}, & if \quad \sum_{n=1}^{5} counter_{adv} = 0 \\ score_{emo} = counter_{\{adv=5\}} \cdot \{2, 1.5, 1.25, 0.5, 0.25\}, & if \quad \sum_{n=1}^{5} counter_{adv} \neq 0 \end{cases} \tag{6}$$

3.5 主题情感倾向评价

通过 3.5 整个步骤就能过获得不同主题下的情感倾向，结果以实数表示，结果的范围不超(-6.6,+6.6)。如果所获取的结果绝对值越大代表情感倾向越强烈。如果是正实数且绝对值越大代表该主题下，客户在线评论对这个主题，大部分表现为较为积极的态度。如果是负实数且绝对值越大代表该主题下，客户在线评论对这个主题，大部分表现为较为消极的态度。

4. 应用分析

4.1 数据获取与预处理

设计并编写网页爬虫程序抓取京东电子商务网站上手机、笔记本电脑、显示器、三个品类商品的在线评论，共得到手机评论 2000 条、笔记本电脑评论 3000 条和显示器评论 5000 条。对评论数据进行分词、去停用词和词性标注等预处理提高数据质量。

4.2 模型主题选取及关键词

通过上述步骤 3.2.5 模型选取手机，笔记本电脑，显示器三个不同条目下的不同主题及关键词。如下表 4、5、6、7 所示

表 4 三个条目下商品的影响因素

条目	条目下的主题（影响因素）
手机	手机外观体验，购物物流与质量，手机性能与媒体体验
笔记本电脑	电脑性能与设计体验，电脑购买与使用体验，购物物流与质量
显示器	购物物流与性价比，显示器性能与视觉体验

表 5 手机条目下的影响因素

影响因素	关键词（前 15 个）
手机外观体验	拍照，运行，屏幕，效果，速度，外观，音效，外形，待机时间，好看不错，清晰，流畅，手机，很快，手感
购物物流与质量	手机，京东，买，满意，质量，苹果，物流，不错，收到，购买，喜欢，值得，价格，包装，购物
手机性能与媒体体验	效果，拍照，屏幕，外观，外形，运行，速度，音效，待机时间，特色，清晰，视频，拍，流畅，模式

表 6 笔记本电脑条目下的影响因素

影响因素	关键词（前 15 个）
电脑性能与设计体验	运行，速度，外观，屏幕，轻薄，效果，散热，性能，外形，程度，很快，不错，清晰，好看，特色，薄
电脑购买与使用体验	华为，电脑，买，办公，笔记本，手机，开机，系统，不错，第一次，款，流畅，键盘，感觉，颜值，配置
购物物流与质量	京东，质量，物流，满意，购物，值得，不错，包装，购买，收到，价格，发货，产品，信赖，品牌

表 7 显示器条目下的影响因素

影响因素	关键词（前 15 个）
购物物流与性价比	不错，买，显示器，京东，质量，满意，性价比，高，价格，包装，很快，物流，购买，真的，值得
显示器性能与视觉体验	效果，游戏，不错，外观，尺寸，刷新率，大小，屏幕，显示器，外形，寸，显色，高，显示

4.3 评论情感倾向预测

通过公式（3）计算情感极性，通过公式（4）（5）计算加强情感倾向，通过公式（6）计算程度副词下的影响因素。通过调节均衡函数$f_{balance}(x)$中的δ均衡因子获取不同情感倾向预测，不同结果如下表 8、9、10 所示。

表 8 $\delta = 1.00$时情感倾向预测结果

条目	影响因素及预测结果		
手机	手机外观体验	购物物流与质量	性能与媒体体验
	+4.365	+5.442	+2.721
电脑	性能与设计体验	购买与使用体验	购物物流与质量
	+4.365	+5.820	+5.821
显示器	购物物流与性价比	性能与视觉体验	
	+4.366	+2.721	

表 9 $\delta = 0.75$时情感倾向预测结果

条目	影响因素及预测结果		
手机	手机外观体验	购物物流与质量	性能与媒体体验
	+4.483	+5.787	+2.893
电脑	性能与设计体验	购买与使用体验	购物物流与质量
	+4.483	+5.978	+5.977
显示器	购物物流与性价比	性能与视觉体验	
	+4.482	+2.893	

表 10 $\delta = 0.5$时情感倾向预测结果

条目	影响因素及预测结果		
手机	手机外观体验	购物物流与质量	性能与媒体体验
	+4.638	+6.466	+3.233
电脑	性能与设计体验	购买与使用体验	购物物流与质量
	+4.637	+6.183	+6.183
显示器	购物物流与性价比	性能与视觉体验	
	+4.637	+3.233	

4.4 各影响因素下情感倾向预测评价

根据公式（4）中均衡函数当中均衡因子δ须保证总体均衡函数呈缓慢单调增的特性，通过多次对比实验发现$\delta = 0.75$附近效果最为显著。分析结果如下表 11 所示

表 11 $\delta = 0.75$时各影响因素情感倾向预测值

条目	各影响因素下情感倾向预测值排序
手机	购物物流与质量>手机外观体验>性能与媒体体验
电脑	购买与使用体验≥购物物流与质量>性能与设计体验
显示器	购物物流与性价比>性能与视觉体验

在该案例中我们发现，手机，笔记本电脑，显示器三个条目中，对于商品来说，影响消费者情感偏好的因素并非完全相同，但部分因素具有共性。这种现象源于这四个品类的商品虽属于电子产品类别，但仍存在差异，如手机消费者关注的“购物的物流及手机质量”，显示器消费者购买用户会更在意“购物物流与性价比”。

本研究通过多次实验设计后确定均衡因子为$\delta = 0.75$时符合公式规律以及结果预期。通过表 发现对于电子产品条目下的商品客户对于购物物流始终放在第一位。其中影响因素除了可能是电子产品本身所具有的属性还存在，数据获取的平台是京东电子商务平台，该平台在市场竞争中始终以优质物流作为吸引顾客点；手机对于外观体验的情感倾向大于性能。通过分析可能是因为在大部分顾客购买手机后在评论时，由于手机是全新包装，外观精美相比于手机性能在这时候更关注于外观体验。在一段时间后可能顾客对于外观体验，相比于手机性能来说，就不会过分关注。

综合上面的影响因素对于手机商家，需要在“性能与媒体体验”上面多加入一些满足顾客情感倾向的因素，例如，在商家新商品的时候时撰写词条时多加入一些有关“拍照效果”、“待机时间”等等在“性能与媒体体验”主题下的关键词。又例如，在撰写有关显示器词条的时候多加入一些有关“显示器性能与视觉体验”的关键词例如从该模型中挖掘出来的“刷新率”、“游戏”等等。对于通过LDA 主题模型挖掘出来相对优秀的部分，也可以在关键词下寻找一些偶尔出现的关键词作为下一次撰写商品词条的新文本。对于相对欠缺的部分应当及时改进。

4.5 模型评价与比较

本研究为了验证模型具有提升效果，采取人工情感词标注以及打分的规则，来验证模型的有效性。为此我们选取 20 个独立互不影响的同学，对模型提取下的关键词如表 4、5、6 所示的关键词，然后进行情感标注及打分，具体规则如下；对于名词设计为 0 分，形容词、副词等参与打分，如果认为该词为积极那么打+1 分，如果认为消极则打-1 分。对于提取下的关键词越靠前越有权重，我们则设计加入权重具体为：（15-词语位置+1）* 0.01。结果如下图 12、13、14 所示。

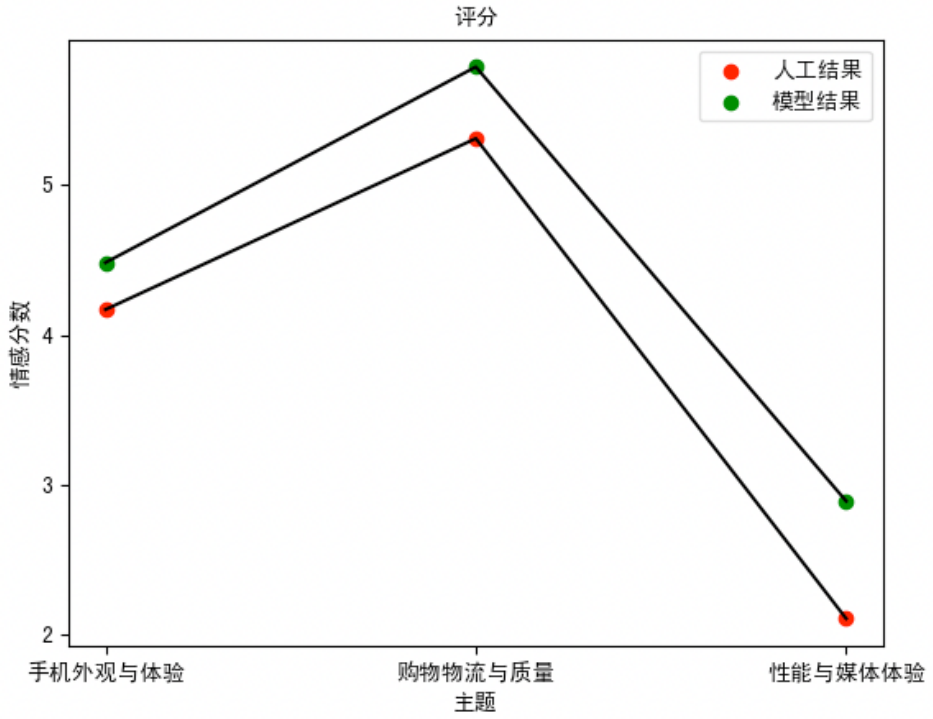

图 12 手机各主题人工与模型结果比较

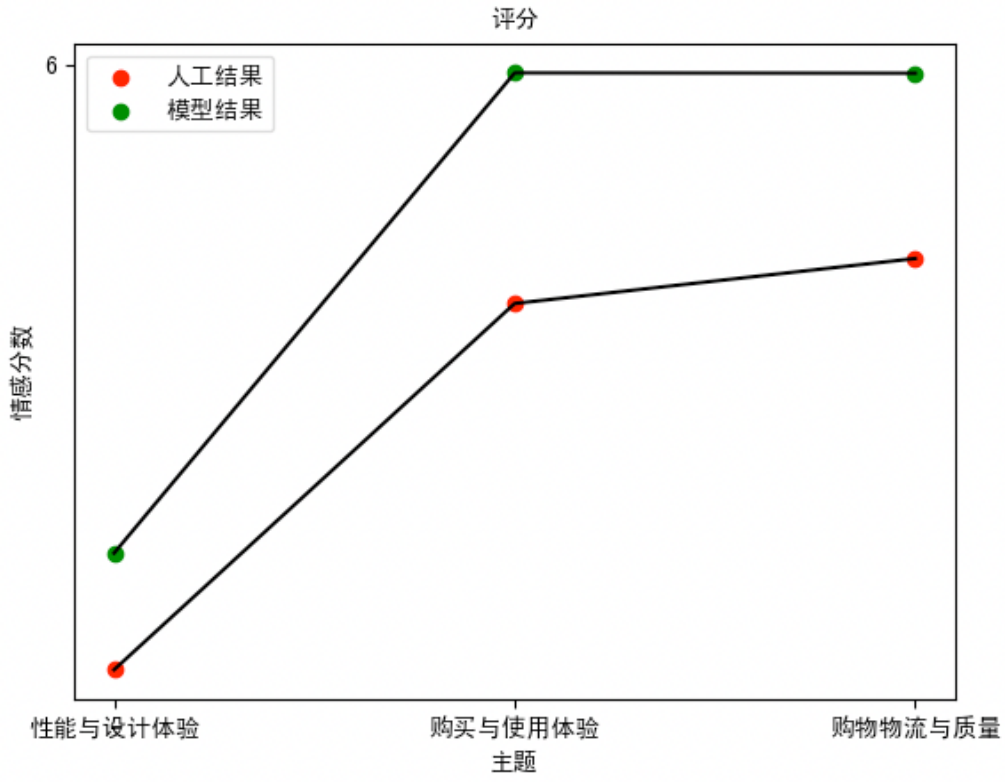

图 13 笔记本电脑各主题人工与模型结果比较

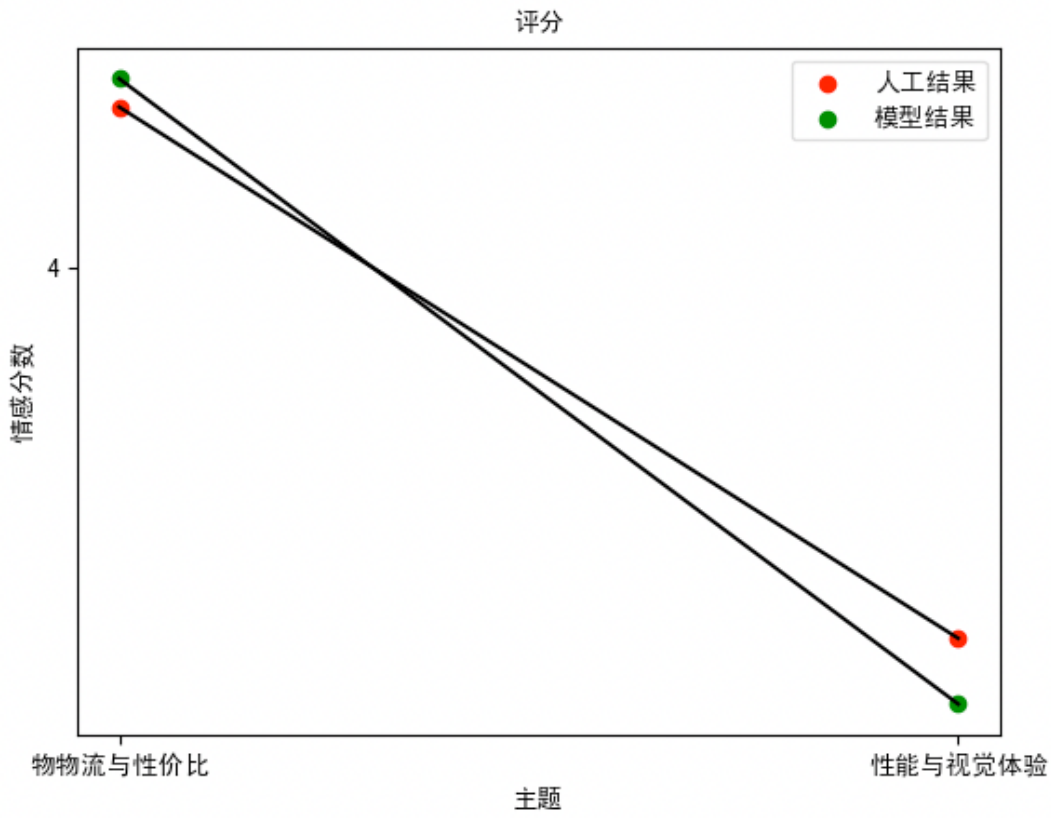

图 14 显示器各主题人工与模型结果比较

由图 12、13、14 可以看出，模型结果相对于人工情感标注较为更出色，由此可以验证本模型的有效性。

5. 研究局限与未来展望

本研究的局限大致有两点：第一点在本文的研究案例中，本文的虽然引入了情感词典来判别情感倾向，但是该方法略显粗糙，因为情感词典并不是动态更新，所以很多在电子商务平台在新出现的评论，很可能被当作噪声被去除了。这方面可通过基于深度学习的神经网络改善，但受到技术、理论方面不足的限制，本文未能引入改技术。第二点在本文的研究当中，每句评论被机械性的去处停用词，后分词拆开，虽然去除了大部分噪声，但是由于缺乏一些句法规则的约束也间接导致了部分数据的模糊，比如对于形容词“快”，可能是形容“物流”也可能是形容“手机运行”，同时生成的关键词也很难定位到原始文本位置，这也是未来研究和改进的一个重要方向。除了以上局限部分本研究还有部

分基于现有基础的创新，查阅领域情感词典有关的文献，大部分都是对一段文本进行情感分析，或者基于某些规则对于文本进行情感分析。但是本文着重研究点是通过在 LDA 主题模型下通过生成的关键词判别，整个主题（影响因素）的情感倾向。这点可作为部分创新。对于未来研究文本情感分析大致可以从动态情感词典的构建进行改进，或者引入多神经元节点的深度学习神经网络也可以构建动态情感词典。

参考文献

[1] Li, Hongxiu, and Reima Suomi. "E-commerce development in China: opportunities or challenges." Proceedings of the IADIS International Conference on E-commerce (Krishnamurthy, S. & Isaias, P. Ed.). 2006.

[2] Jiaxiu He, Xin Wang, Mark B. Vandenbosch, Barrie R. Nault,Revealed preference in online reviews: Purchase verification in the tablet market,Decision Support Systems,Volume 132, 2020, 113281

[3] 崔耕, 庄梦舟, 彭玲. 莫让网评变为 “罔评”: 故意操纵网络产品评论对消费者的影响[J]. 营销科学学报, 2014, 10(1): 21-34.

[4] 李贺, 谷莹, 刘嘉宇. 数据驱动下基于语义相似性的产品需求识别研究[J]. 情报理论与实践, 2022, 45(5): 99-106.

[5] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. Journal of machine Learning research, 2003, 3(Jan): 993-1022.

[6] Nguyen D, Doğruöz A S, Rosé C P, et al. Computational sociolinguistics: A survey[J]. Computational linguistics, 2016, 42(3): 537-593.

[7] Habbat N, Anoun H, Hassouni L. Topic Modeling and Sentiment Analysis with LDA and NMF on Moroccan Tweets[C]//Innovations in Smart Cities Applications Volume 4: The Proceedings of the 5th International Conferenc e on Smart City Applications. Springer International Publishing, 2021: 147-161.

[8] Huang X, Rao Y, Xie H, et al. Cross-domain sentiment classification via t opic-related TrAdaBoost[C]//Proceedings of the AAAI Conference on Artifi cial Intelligence. 2017, 31(1).

[9] 赵妍妍,秦　兵,刘　挺.文本情感分析.软件学报,2010,21(8):1834-1848

[10] 赵常煜, 吴亚平, 王继民. “一带一路” 倡议下的 Twitter 文本主题挖掘和情感分析[J]. 图书情报工作, 2019, 63(19): 119.

[11] 钟佳娃,刘巍,王思丽,杨恒.文本情感分析方法及应用综述[J].数据分析与知识发现,2021,5(06):1-13.

[12] Chen Y, Skiena S. Building sentiment lexicons for all major languages[C]//Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2014: 383-389.

[13] Zhang S, Wei Z, Wang Y, et al. Sentiment analysis of Chinese micro-blog text based on extended sentiment dictionary[J]. Future Generation Computer Systems, 2018, 81: 395-403.

[14] Kaewpitakkun Y, Shirai K, Mohd M. Sentiment lexicon interpolation and polarity estimation of objective and out-of-vocabulary words to improve sentiment classification on microblogging[C]//Proceedings of the 28th Pacific Asia conference on language, information and computing. 2014: 204-213.

[15] Liu R, Shi Y, Ji C, et al. A survey of sentiment analysis based on transfer learning[J]. IEEE access, 2019, 7: 85401-85412.

[16] Ray B, Garain A, Sarkar R. An ensemble-based hotel recommender system using sentiment analysis and aspect categorization of hotel reviews[J]. Applied Soft Computing, 2021, 98: 106935.

[17] 刘慧清, 郭延哺, 李维华. 基于贝叶斯网的跨领域情感分析方法[J]. 计算机应用与软件, 2020, 37: 12.

[18] Chen Y. Convolutional neural network for sentence classification[D]. University of Waterloo, 2015.

[19] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[20] Linmei, Hu, Yang, Tianchi, et al. Heterogeneous Graph Attention Networks for Semi-supervised Short Text Classification[C]. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).2019:4821-4830

[21] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. Journal of machine Learning research, 2003, 3(Jan): 993-1022.

[22] DiMaggio P, Nag M, Blei D. Exploiting affinities between topic modeling and the sociological perspective on culture: Application to newspaper coverage of US government arts funding[J]. Poetics, 2013, 41(6): 570-606.

[23] Syed S, Spruit M. Full-text or abstract? examining topic coherence scores using latent dirichlet allocation[C]//2017 IEEE International conference on data science and advanced analytics (DSAA). IEEE, 2017: 165-174.

[24] Röder M, Both A, Hinneburg A. Exploring the space of topic coherence measures[C]//Proceedings of the eighth ACM international conference on Web search and data mining. 2015: 399-408.

[25] Newman D, Lau J H, Grieser K, et al. Automatic evaluation of topic coherence[C]//Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics. 2010: 100-108.

[26] Sievert C, Shirley K. LDAvis: A method for visualizing and interpreting topics[C]//Proceedings of the workshop on interactive language learning, visualization, and interfaces. 2014: 63-70.

[27] 季旺,夏振宇.基于改进 TFIDF 算法的情感分析模型研究[J].计算机与数字工程,2022,50(08):1671-1675.

致 谢

本文不仅体现了本人几年来的学习与研究成果，同时它也更多地凝结了老师、同学、朋友和加热的支持与帮助。在这里我向他们表示由衷的感谢。

我的导师对我论文的完成进行了全面而细致的指导，从论文的选题、拟定提纲、研究工作的展开一直到论文的完成都倾注了彭老师的大量心血。虽然平时工作比较繁忙，但他经常抽出时间利用电话、QQ 等方式对我进行悉心的指导和支持。帮助我解决了我在课题研究中产生的问题、遭遇的困难，使我在日常工作学习中少走了很多弯路。

另外，同届同学从本文开题阶段开始一直到论文全部完成都提供了很大帮助，至今仍经常询问我的工作、生活以及论文完成情况，并根据自己的实践经验提供合理的意见和建议。大学同学以及论文指导老师也经常关系我论文的进展情况，并有针对性地提出意见和建议，也经常提醒我对论文要报以认真态度对待，时刻不能马虎，修改一定要谨慎。还有我的家人，他们在生活上在工作上都给予我无微不至的关心和支持，才有了我现在的论文研究成果。

最后，向在我这几年在学习和生活中曾给予我支持和教导、扶持和帮助的老师和同学表示深深的谢意。

附录一：爬虫代码附录

```
# -*- coding:utf-8 -*-
import requests
import re
import time
import csv

# https://api.m.jd.com/?appid=item-v3&functionId=pc_club_productPageComments&client=pc&clientVersion=1.0.0&t=1682625231933&loginType=3&uuid=122270672.1037635570.1681383230.1681450514.1682624917.4&productId=100009554947&score=0&sortType=5&page=0&pageSize=10&isShadowSku=0&fold=1
# https://api.m.jd.com/?appid=item-v3&functionId=pc_club_productPageComments&client=pc&clientVersion=1.0.0&t=1682625594709&loginType=3&uuid=122270672.1037635570.1681383230.1681450514.1682624917.4&productId=100043588742&score=0&sortType=5&page=1&pageSize=10&isShadowSku=0&fold=1
# https://api.m.jd.com/?appid=item-v3&functionId=pc_club_productPageComments&client=pc&clientVersion=1.0.0&t=1682625699171&loginType=3&uuid=122270672.1037635570.1681383230.1681450514.1682624917.4&productId=100032149194&score=0&sortType=5&page=0&pageSize=10&isShadowSku=0&fold=1

#读取 csv 文件
csv_header =['numbers','comments']
with open('/Users/ankew/Desktop/data2.csv','w',encoding='utf-8',newline='') as file:
        writer = csv.writer(file)
        writer.writerow(csv_header)

#  网址
prefix_url ="https://api.m.jd.com/?appid=item-v3&functionId=pc_club_productPageComments&client=pc&clientVersion=1.0.0&t=1682625231933&loginType=3&uuid=122270672.1037635570.1681383230.1681450514.1682624917.4&productId=100009554947&score=0&sortType=5&page="
suffix_url ="&pageSize=10&isShadowSku=0&fold=1"

prefix_url_two =    "https://api.m.jd.com/?appid=item-v3&functionId=pc_club_productPageComments&client=pc&clientVersion=1.0.0&t=
```

```
1682625594709&loginType=3&uuid=122270672.1037635570.1681383230.1681450
514.1682624917.4&productId=100043588742&score=0&sortType=5&page="
prefix_url_three= "https://api.m.jd.com/?appid=item-
v3&functionId=pc_club_productPageComments&client=pc&clientVersion=1.0.0&t=
1682625594709&loginType=3&uuid=122270672.1037635570.1681383230.1681450
514.1682624917.4&productId=100032149194&score=0&sortType=5&page="
# 伪装
headers = {"user-agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36"}

for i in range(0,100): # 最大页数 100 页
    url = prefix_url + str(i) + suffix_url

    # 发送请求
    response = requests.get(url=url, headers=headers)
    # print(response)# 打印响应状态 <Response [200]>：表示已经响应成功了

    # 获取数据
    if response.content:
        json_data = response.text

    # 解析数据 列表
    comments = json_data

    # 正则筛选
    res = r'"guid":"[a-zA-Z0-9]{32}"\s*,\s*"content":"(.*?)"'
    content = re.findall(res, comments)

    # 替换掉\n
    contents =[]
    for k in range(0,len(content)): # temp 为临时变量
        temp =
content[k].replace("\\n","").replace("&ldquo;","").replace("&rdquo;","") # cotent 去
掉\n 变为 contents
        contents.append(temp)

    # print(content)
    for index in range(0,len(contents)):
        csv_content =[index+1+i*10,contents[index]]
        with open('/Users/ankew/Desktop/data2.csv','a',encoding='utf-8',newline='')
as file:
            writer = csv.writer(file)
```

```
                writer.writerow(csv_content)
            print("第{num1}条评论  {content}\n".format(num1=10 * i + index +
1,content=contents[index]) )

        # 休息 20 秒在进行
        time.sleep(3)

for i in range(0,100): # 最大页数 100 页
        url = prefix_url_two + str(i) + suffix_url

        # 发送请求
        response = requests.get(url=url, headers=headers)
        #print(response)# 打印响应状态 <Response [200]>：表示已经响应成功了

        # 获取数据
        if response.content:
            json_data = response.text

        # 解析数据 列表
        comments = json_data

        # 正则筛选
        res = r'"guid":"[a-zA-Z0-9]{32}"\s*,\s*"content":"(.*?)"'
        content = re.findall(res, comments)

        # 替换掉\n
        contents =[]
        for k in range(0,len(content)): # temp 为临时变量
            temp =
content[k].replace("\\n","").replace("&ldquo;","").replace("&rdquo;","") # cotent 去
掉\n 变为 contents
            contents.append(temp)

        #print(content)
        for index in range(0,len(contents)):
            csv_content =[1000+index+1+i*10,contents[index]]
            with open('/Users/ankew/Desktop/data2.csv','a',encoding='utf-8',newline='')
as file:
                writer = csv.writer(file)
                writer.writerow(csv_content)
            print("第{num1}条评论  {content}\n".format(num1=1000+10 * i + index +
1,content=contents[index]) )
```

```
    # 休息 20 秒在进行
    time.sleep(3)

for i in range(0,100): # 最大页数 100 页
    url = prefix_url_three + str(i) + suffix_url

    # 发送请求
    response = requests.get(url=url, headers=headers)
    # print(response)# 打印响应状态 <Response [200]>：表示已经响应成功了

    # 获取数据
    if response.content:
        json_data = response.text

    # 解析数据 列表
    comments = json_data

    # 正则筛选
    res = r'"guid":"[a-zA-Z0-9]{32}"\s*,\s*"content":"(.*?)"'
    content = re.findall(res, comments)

    # 替换掉\n
    contents =[]
    for k in range(0,len(content)): # temp 为临时变量
        temp =
content[k].replace("\\n","").replace("&ldquo;","").replace("&rdquo;","") # cotent 去
掉\n 变为 contents
        contents.append(temp)

    # print(content)
    for index in range(0,len(contents)):
        csv_content =[2000+index+1+i*10,contents[index]]
        with open('/Users/ankew/Desktop/data2.csv','a',encoding='utf-8',newline='')
as file:
            writer = csv.writer(file)
            writer.writerow(csv_content)
        print("第{num1}条评论 {content}\n".format(num1=2000+10 * i + index +
1,content=contents[index]) )

    # 休息 20 秒在进行
    time.sleep(3)
```

附录二：词云代码附录

```
import pandas as pd
import jieba
from matplotlib.font_manager import FontProperties
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
import matplotlib.pyplot as plt

# 读取 csv 文件
df = pd.read_csv(r"/Users/ankew/Desktop/DataAna/data.csv")

# 将评论拼接成一个字符串
text = ' '.join(df['comments'].tolist())

# 中文分词
seg_list = jieba.cut(text)

# 去除停用词
stopwords = set(STOPWORDS)
with open(r"/Users/ankew/Desktop/stopwords.txt", 'r', encoding='utf-8') as f:
    for word in f.readlines():
        stopwords.add(word.strip())

seg_list = [i for i in seg_list if i not in stopwords and i != ' ']

# 统计词频
word_counts = {}
for word in seg_list:
    word_counts[word] = word_counts.get(word, 0) + 1

# 全局中文字体
plt.rcParams['font.sans-serif'] = ['SimHei']

# 生成词云
wc = WordCloud(background_color="white", max_words=2000,
               font_path='/Users/ankew/opt/anaconda3/pkgs/matplotlib-base-
3.5.1-py39hfb0c5b7_1'+
                   '/lib/python3.9/site-packages/matplotlib/mpl-
data/fonts/ttf/SimHei.ttf',
               width=400,height=400,scale=2)
wc.generate_from_frequencies(word_counts)
```

```
# 显示词云图
plt.imshow(wc, interpolation='bilinear')
plt.axis("off")
plt.show()
```

附录三：LDA 主题模型及情感分析目录

```
import pandas as pd
import jieba
from gensim import corpora, models
import pyLDAvis.gensim_models as gensimvis
import matplotlib.pyplot as plt
import pyLDAvis.gensim_models
import multiprocessing
import datetime
import math as ma

# 开始时间
start_time = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')
print("开始的时间是{}".format(start_time))

# 设置多进程启动方法
multiprocessing.set_start_method('fork')

# 读取数据集
df = pd.read_csv(r"/Users/ankew/Desktop/DataAna/data.csv")

# 读取情感词典集合以及副词集合
pos_eval_data = pd.read_csv(r"/Users/ankew/Desktop/HowNet-
Encoding=GB18030/pos_eval.csv")
neg_eval_data = pd.read_csv(r"/Users/ankew/Desktop/HowNet-
Encoding=GB18030/neg_eval.csv")

pos_emo_data = pd.read_csv(r"/Users/ankew/Desktop/HowNet-
Encoding=GB18030/pos_emo.csv")
neg_emo_data = pd.read_csv(r"/Users/ankew/Desktop/HowNet-
Encoding=GB18030/neg_emo.csv")

most_data = pd.read_csv(r"/Users/ankew/Desktop/HowNet-
Encoding=GB18030/most.csv")   # 2
very_data = pd.read_csv(r"/Users/ankew/Desktop/HowNet-
Encoding=GB18030/very.csv")   # 1.5
```

```
more_data = pd.read_csv(r"/Users/ankew/Desktop/HowNet-
Encoding=GB18030/more.csv")    # 1.25
ish_data = pd.read_csv(r"/Users/ankew/Desktop/HowNet-
Encoding=GB18030/ish.csv")    # 0.5
insufficiently_data = pd.read_csv(r"/Users/ankew/Desktop/HowNet-
Encoding=GB18030/insufficiently.csv")    # 0.25

# 获取正负评价文档列表并去除后缀 r"\xa0"
pos_eval_list = pos_eval_data["contents"].tolist()
for i in range(0, len(pos_eval_list)):
        pos_eval_list[i] = "".join(pos_eval_list[i].split())
neg_eval_list = neg_eval_data["contents"].tolist()
for i in range(0, len(neg_eval_list)):
        neg_eval_list[i] = "".join(neg_eval_list[i].split())

# 获取正负情感文档列表并去除后缀 r"\xa0"
pos_emo_list = pos_emo_data["contents"].tolist()
for i in range(0, len(pos_emo_list)):
        pos_emo_list[i] = "".join(pos_emo_list[i].split())
neg_emo_list = neg_emo_data["contents"].tolist()
for i in range(0, len(neg_emo_list)):
        neg_emo_list[i] = "".join(neg_emo_list[i].split())

# 获取程度副词文档列表并去除后缀 r"\xa0"
most_list = most_data["contents"].tolist()
for i in range(0, len(most_list)):
        most_list[i] = "".join(most_list[i].split())
very_list = very_data["contents"].tolist()
for i in range(0, len(very_list)):
        very_list[i] = "".join(very_list[i].split())
more_list = more_data["contents"].tolist()
for i in range(0, len(more_list)):
        more_list[i] = "".join(more_list[i].split())
ish_list = ish_data["contents"].tolist()
for i in range(0, len(ish_list)):
        ish_list[i] = "".join(ish_list[i].split())
insufficiently_list = insufficiently_data["contents"].tolist()
for i in range(0, len(insufficiently_list)):
        insufficiently_list[i] = "".join(insufficiently_list[i].split())

# 中文分词
def chinese_word_cut(mytext):
```

```
    return " ".join(jieba.cut(mytext))

# 去除停用词和标点符号
stopwords = [line.strip() for line in open(r"/Users/ankew/Desktop/stopwords.txt", 'r',
encoding='utf-8').readlines()]
df['content_cutted'] = df['comments'].apply(chinese_word_cut)
df['content_cutted'] = df['content_cutted'].apply(
    lambda x: ' '.join([word for word in x.split() if word not in stopwords]))

# 构建语料库
texts = [doc.split() for doc in df['content_cutted']]
dictionary = corpora.Dictionary(texts)
corpus = [dictionary.doc2bow(text) for text in texts]

# 绘制困惑度曲线和主题一致性曲线
coherence_scores = []
perplexity_scores = []
for num_topics in range(2, 20):
    lda_model = models.ldamodel.LdaModel(corpus=corpus,
                                         id2word=dictionary,
                                         num_topics=num_topics,
                                         iterations=1000,
                                         passes=10,
                                         random_state=1)
    coherence_model_lda = models.CoherenceModel(model=lda_model, texts=texts,
dictionary=dictionary, coherence='c_v')
    coherence_lda = coherence_model_lda.get_coherence()
    coherence_scores.append(coherence_lda)
    perplexity_scores.append(lda_model.log_perplexity(corpus))

# 绘制困惑度曲线
x = range(2, 20)
plt.plot(x, perplexity_scores)
plt.xlabel('Number of Topics')
plt.ylabel('Perplexity Score')
plt.show()

# 绘制主题一致性曲线
plt.plot(x, coherence_scores)
plt.xlabel('Number of Topics')
plt.ylabel('Coherence Score')
plt.show()
```

```
# 选择合适的主题数量
optimal_num_topics = coherence_scores.index(max(coherence_scores)) + 2
print("Optimal Number of Topics:", optimal_num_topics)

# 训练 LDA 模型
lda_model = models.ldamodel.LdaModel(corpus=corpus,
                                     id2word=dictionary,
                                     num_topics=optimal_num_topics,
                                     iterations=1000,
                                     passes=10,
                                     random_state=1)

# 可视化主题
vis = gensimvis.prepare(lda_model, corpus, dictionary)
pyLDAvis.display(vis)
pyLDAvis.save_html(vis, 'screen_comments_result_random_two.html')  # 将结果保存为该 html 文件

# pycharm 单独设置显示所有列的结果
pd.set_option('display.max_columns', None)  # 显示所有列
pd.set_option('display.max_rows', None)  # 显示所有行

# 用表格展示结果
all_topics = {}
num_terms = 10  # Adjust number of words to represent each topic
lambd = 0.6  # Adjust this accordingly based on tuning above lambda constant
for i in range(1, optimal_num_topics + 1):
    topic = vis.topic_info[vis.topic_info.Category == 'Topic' + str(i)].copy()
    topic['relevance'] = topic['loglift'] * (1 - lambd) + topic['logprob'] * lambd
    all_topics['Topic ' + str(i)] = topic.sort_values(by='relevance',
ascending=False).Term[:num_terms].values
print(pd.DataFrame(all_topics).T + "\n")

# 提取每个主题下的前 30 个关键词
topics = lda_model.show_topics(num_topics=optimal_num_topics, num_words=30,
formatted=False)

# 把关键词存进去
topics_words = []
for topic in range(0, optimal_num_topics):  # num_topics = optimal_num_topics
    topics_words.append([topic])
```

```
for each_topic_num in range(0, optimal_num_topics): # num_topics = optimal_num_topics
    for element_content in range(0, len(topics[each_topic_num][1])):

topics_words[each_topic_num].append(topics[each_topic_num][1][element_content][0])

# 弹出首位数字
for num in range(0, len(topics_words)):
    topics_words[num].pop(0)  # 得到纯文本链表

# 运行算法

# 计算基础评价分数
pos_counter = 0
neg_counter = 0
basis_topics_scores = []

for i in range(0, optimal_num_topics):  # num_topics = optimal_num_topics

    # 把正负指针置零
    pos_counter = 0
    neg_counter = 0

    for content in topics_words[i]:

        # 统计正，负面评价词数
        if content in pos_eval_list:
            pos_counter = pos_counter + 1

        elif content in neg_eval_list:
            neg_counter = neg_counter + 1

    # 一共 6 轮，每轮正负评价词数统计完毕后进行比较赋值

    if pos_counter > neg_counter:
        # 如果 pos_counter > neg_counter, 该主题下的关键词是积极的, 句子基本值设定为 +2
        basis_topics_scores.append(+2)

    elif pos_counter < neg_counter:
        # 如果 pos_counter < neg_counter, 该主题下的关键词是消极的, 句子基
```

```
本值设定为 -2
            basis_topics_scores.append(-2)
        else:
            # 如果 pos_counter = neg_counter, 该主题下的关键词是中性的, 句子基本值设定为 0
            basis_topics_scores.append(0)

# print(basis_topics_scores) # 打印基本情感分数

# 在基础评价分数上计算临时情感分数
pos_emo_counter = 0
neg_emo_counter = 0
temp_topics_scores = []
log_temp_pos = 0
log_temp_neg = 0

for i in range(0, optimal_num_topics):    # num_topics = 6

    # 把正负指针置零
    pos_emo_counter = 0
    neg_emo_counter = 0

    # 统计正，负面情感词数
    for content in topics_words[i]:
        if content in pos_emo_list:
            pos_emo_counter = pos_emo_counter + 1
        elif content in neg_emo_list:
            neg_emo_counter = neg_emo_counter + 1

    #liyong
    if pos_emo_counter >= 1 :
        log_temp_pos = pos_emo_counter / ma.log(pos_emo_counter + 1)
    elif neg_emo_counter >= 1 :
        log_temp_neg = neg_emo_counter / ma.log(neg_emo_counter + 1)
    # elif pos_emo_counter == 1:
    #     log_temp_pos = 1
    # elif neg_emo_counter == 1:
    #     log_temp_neg = 1
    elif pos_emo_counter == 0:
        log_temp_pos = 0
    elif neg_emo_counter == 0:
        log_temp_neg = 0
```

```
        # 基本分数加上情感词分数
        if basis_topics_scores[i] > 0:
            temp_topics_scores.append(basis_topics_scores[i] + log_temp_pos * 0.5 -
log_temp_neg * 0.5)
        elif basis_topics_scores[i] < 0:
            temp_topics_scores.append(basis_topics_scores[i] - log_temp_pos * 0.5 +
log_temp_neg * 0.5)
        else:
            temp_topics_scores.append(basis_topics_scores[i])

# 在临时情感分数上计算最终得分
most_counter = 0
very_counter = 0
more_counter = 0
ish_counter = 0
insufficiently_counter = 0
final_topics_scores = []

for i in range(0, optimal_num_topics):    # num_topics = 6

        # 各类指针置零
        most_counter = 0
        very_counter = 0
        more_counter = 0
        ish_counter = 0
        insufficiently_counter = 0

        # 统计各类词的个数
        for content in topics_words[i]:
            if content in most_list:
                most_counter = most_counter + 1
            elif content in very_list:
                very_counter = very_counter + 1
            elif content in more_list:
                more_counter = more_counter + 1
            elif content in ish_list:
                ish_counter = ish_counter + 1
            elif content in insufficiently_list:
                insufficiently_counter = insufficiently_counter + 1
        # 防止程度副词为 0 个
        if
```

```
most_counter+very_counter+more_counter+ish_counter+insufficiently_counter == 0:
            final_topics_scores.append(temp_topics_scores[i])
        else:
            final_topics_scores.append(temp_topics_scores[i] * (most_counter * 2 +
very_counter * 1.5 +

more_counter * 1.25 + ish_counter * 0.5 +

insufficiently_counter * 0.25))
        # 最终结果
print(final_topics_scores)

# 结束时间
end_time = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')
print("结束的时间是{}".format(end_time))
```