分类号： 单位代码： 10033

密 级： 学 号： 153520085211015



硕士学位论文



**中文论文题目： 物品词语聚类方法研究与实现**

**英文论文题目： Research and Implementation of Items Words Clustering**

申请人姓名： **程宇芬**

指导教师： **尚文倩**

专业名称：  **计算机技术**

研究方向： **数字娱乐与动画技术**

所在学院： **计算机学院**

**论文提交日期**

中国传媒大学研究生学位论文独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 **中国传媒大学** 或其他教育机构的学位或证书而使用过的材料。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谢意。

学位论文作者签名： 签字日期： 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解 **中国传媒大学** 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磁盘，允许论文被查阅和借阅。本人授权**中国传媒大学**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。

（保密的学位论文在解密后适用本授权书）

学位论文作者签名： 导师签名：

签字日期： 年 月 日 签字日期： 年 月 日

# 致 谢

时光飞逝，在研究生入学的第二年便要匆匆离校，在这段时间的学习生活中，作为一名研究生，我在学术上拓宽了自己的眼界，了解了现如今科学前沿的技术，对如今发展迅猛的大数据技术进行了研究，接触到了可视化技术，对于计算机专业的自己，感受到了探索新知识的乐趣。生活中认识了许多优秀的同学，品学兼优的师哥师姐，师弟师妹。与优秀的人在一起总是不断地让自己不知不觉有所成长，被周围优秀的人所影响，让自己的时间分配，学习习惯也有所改变，我坚信这将是我在日后人生中的巨大财富。

在这段丰富的研究生生活中，我要对所有帮助我，支持我的老师，同学，亲人们表示诚挚的感谢。

首先要感谢我的导师，尚文倩老师，尚老师是一个十分关心学生的老师，对学生的个人发展，学生的生活烦恼都十分关注，并且尽自己的力量帮助学生排解难处。每当自己在学术研究上不知如何下手时，会经常找导师帮忙解答困惑，老师都会耐心解答，让自己的学术研究更顺利的进展。生活中，前一段时间找工作的压力难以排解，当有工作机会时，老师总是及时通知，帮助我尽早找到合心的工作。研究生期间，我觉得十分幸运遇到了这位导师，让我的学习生涯更加顺利。除了我的导师，还要感谢朱立谷老师，十分有幸加入了朱立谷老师的团队，朱立谷老师对学生要求严格，以实际项目为导向，来培养学生的实际操作能力。生活中注重团队的团结，经常组织团队活动，使实验室的关系十分融恰。朱老师十分关心学生，注重学生的个人成长，会经常给学生提出相关的建议，发现学生的问题时也会及时的提醒学生，研究生期间在朱老师带领的这个团队里，使自己可以得到更多的成长。研究生期间还有所有接触的其他老师，在此表示诚挚的谢意，感谢诸位老师的栽培。

然后要感谢实验室的同学，师哥师姐，师弟师妹们，实验室学术氛围很浓，而且不失有趣，因为实验室的这种氛围，使我研究生期间十分喜欢在实验室学习，工作。实验室同学在学习上和生活上时时刻刻的为我提供帮助，非常感谢。还有要感谢室友和班级同学，寝室关系十分和谐，都是源于室友间的相互包容与磨合，生活中有着同学们的鼓励与帮助，才能使自己的求学生活更加充满动力与乐趣。在此祝福实验室毕业的师哥师姐和同学，预祝此后工作顺利，一切顺心。

最后要感谢我的父母和家人，能够走到研究生毕业，离不开他们，是父母背后多年来的默默支持令我安心读书，使我的工作和学习更有动力。我向父母，向家人，表示深深的感谢与祝福。

# 摘 要

近几年来，电子商务已经在很大程度上改变了人们的消费习惯。人们如今大部分的消费通过电子交易的方式实现。对于各个电商企业，倘若通过电子交易产生的大量数据，从中选取交易量较大的客户，分析其不同日期所购买的物品及其物品的相关种类，可以便于进行商业推荐，为提高收益而制定销售计划，制定高质量的商业决策。

如今大数据技术的发展已经十分成熟，在电子商务中，大数据技术也已经有了相关的应用。电子交易所产生的数据往往数据量巨大，并且增长速度非常快。结合大数据技术，对各个客户的交易信息进行物品种类的分析，可以保证处理方法的科学性与处理过程的高效性。

针对上述想法，本文针对物品词语，根据大数据技术的相关理论与相关技术，设计并实现了相应的词语聚类流程。在系统环境上，选择了支持大数据分析的Hadoop与Spark集群；在数据处理上，针对所获得的数据，进行数据预处理，将数据处理为后续工作所需数据格式与内容，之后选择了IK分词工具对预处理后的数据进行分词；在模型实现上，使用了google研发的word2vec词向量转化工具与Kmeans聚类算法相结合的方法，实现对词语的聚类。测试中，通过测试本次研究结果的精确度与运行效率，判断出此词语聚类方法是可行的。

本文总体流程中所涉及的理论知识与相关方法主要包括数据预处理，hadoop与spark集群，中文分词，词语聚类等。文中对这些技术的理论与应用进行了详细介绍。

关键词：电子商务 大数据 Spark HDFS词语聚类

**Research and Implementation of Items Words Clustering**

# ABSTRACT

In recent years, e-commerce has changed people's consumptive habits to a great extent. Nowadays, most of the consumption is realized by electronic trading. For each ecommerce company, if select some customers who has high volume, analyze the items and their types purchased at different date. Through the data enterprises can make sales plan for increase profitability, make high-quality business decisions easier.

The development of big data technology has been very mature. In e-commerce, big data technologies also already had related applications. Data generated by the electronic exchange is often enormously and rapidly. Combined with the big data technology, to analyze the types of items through transaction information for each customer, can ensure the scientific of the processing method and the efficiency of the process.

According to above-mentioned idea, the thesis aims at items words, according to the related theories and technologies of big data technology, I design and implement the corresponding word clustering process. For the system environment, choose Spark and Hadoop cluster which support the analysis of big data. For data processing, according to the obtained data, carry out data preprocess, process data as useful content and data formats, then use IK word segmentation tool on the data which through preprocessed. For the realization of model, use the method of combine Word2vec word vector transformation tool and Kmeans clustering algorithm to realize word clustering. At last, by test the accuracy and efficiency of the results, proves that the clustering method is feasible.

In the research, the theoretical knowledge and related methods involved in the overall process include data preprocessing, Hadoop and spark cluster, Chinese word segmentation, word clustering and so on. In the thesis, the theory and application of these technologies are introduced in detail.

**Keywords:** Electronic Commerce, Big Data, Spark, HDFS, Word clustering

# 目录

[致 谢 I](#_Toc480485918)

[摘 要 II](#_Toc480485919)

[ABSTRACT III](#_Toc480485920)

[目录 IV](#_Toc480485921)

[1 绪论 1](#_Toc480485922)

[1.1 课题背景 1](#_Toc480485923)

[1.2 国内外研究现状 2](#_Toc480485924)

[1.3 研究意义及其目标 3](#_Toc480485925)

[1.4 研究内容与文章结构 4](#_Toc480485926)

[1.4.1 研究内容 4](#_Toc480485927)

[1.4.2 文章结构 4](#_Toc480485928)

[2 底层架构 5](#_Toc480485929)

[2.1 Hadoop 5](#_Toc480485930)

[2.1.1 HDFS架构 7](#_Toc480485931)

[2.1.2 Map Reduce模型 10](#_Toc480485932)

[2.2 Spark框架 11](#_Toc480485933)

[2.2.1 Spark模块 12](#_Toc480485934)

[2.2.2 Spark运算模型 13](#_Toc480485935)

[2.3 本章小结 13](#_Toc480485936)

[3 关键技术介绍 14](#_Toc480485937)

[3.1 数据预处理 14](#_Toc480485938)

[3.1.1 数据清洗 14](#_Toc480485939)

[3.1.2 数据变换 15](#_Toc480485940)

[3.1.3 数据集成 16](#_Toc480485941)

[3.1.4 数据归约 17](#_Toc480485942)

[3.2 中文分词工具 18](#_Toc480485943)

[3.3 词语聚类方法 20](#_Toc480485944)

[4 词语聚类模型设计 22](#_Toc480485945)

[4.1 需求设计 22](#_Toc480485946)

[4.1.1 性能需求 22](#_Toc480485947)

[4.1.2 功能需求 22](#_Toc480485948)

[4.2 架构设计 23](#_Toc480485949)

[4.2.1 硬件架构 23](#_Toc480485950)

[4.2.2 软件架构 24](#_Toc480485951)

[4.3 流程设计 25](#_Toc480485952)

[4.4 本章小结 26](#_Toc480485953)

[5 物品词语聚类实现 27](#_Toc480485954)

[5.1 数据采集模块 27](#_Toc480485955)

[5.2 数据处理模块 27](#_Toc480485956)

[5.2.1 数据预处理 28](#_Toc480485957)

[5.2.2 分词处理 29](#_Toc480485958)

[5.2.3 词语聚类 29](#_Toc480485959)

[5.3 本章小结 32](#_Toc480485960)

[6 物品词语聚类结果分析 33](#_Toc480485961)

[6.1 系统部署 33](#_Toc480485962)

[6.2 测试与分析 34](#_Toc480485963)

[6.2.1 测试数据集 34](#_Toc480485964)

[6.2.2 分词测试 35](#_Toc480485965)

[6.2.3 词语聚类测试 36](#_Toc480485966)

[6.2.4 结果分析 37](#_Toc480485967)

[6.3 本章小结 38](#_Toc480485968)

[7 总结与展望 39](#_Toc480485969)

[7.1 论文总结 39](#_Toc480485970)

[7.2 展望与后续工作 39](#_Toc480485971)

[参 考 文 献 41](#_Toc480485972)

[攻读硕士学位期间取得的学术成果 44](#_Toc480485973)

# 绪论

## 1.1 课题背景

现如今，随着互联网越来越迅速的发展，网络上的信息增长速度也越来越快，在这信息爆炸的时代，大数据技术应运而生并且随之发展。

计算机的应用越来越广泛地情况下，人们希望与计算机可以利用更加便捷的方式进行交流，智能化人机接口得以发展，而自然语言处理技术为智能化人机接口的核心技术之一。与此同时，在人们通过计算机大量获取并处理消息时，所获取与处理的信息大部分为文本信息。此情况下，人们需要更加科学快捷的方式令机器对文本信息进行分析处理，文本信息的处理离不开自然语言处理技术。自然语言处理学科在此类需求下产生。自然语言处理表示通过计算机对自然语言进行转化，存储到计算机之中进行分析处理的科学，是综合性科学，与数学、语言学、统计学等学科都有所联系[1]。

聚类为机器学习中的无监督学习方法，自动化程度较高，应用于文本分类，信息检索等方面十分广泛。聚类中的词语聚类即根据词语的语义、语用、语法词汇进行聚类，使得同一个类中的词语相似度尽可能高，不同类别的词语相似度尽可能低。词语聚类的研究于自然语言处理方面有着重要的意义，于国内外均有着广泛的关注，现如今也已经取得了一定的研究成果[2]。

电子商务是在网络环境下的经济活动，在经济全球化的发展中，电子商务也随之发展。电子商务的发展改变了人们日常的购物方式，使购物不受时间、地点的限制，人们生活更加便捷[3]。电子商务中主要承载着物品交易，电子商务发展的热潮，也推动了对电子商务信息处理研究的不断推进。基于此背景，倘若对所交易的物品类别情况进行分析，通过对大量数据进行分析，清晰的展示市场需求与市场趋势，为商家提供设计商业计划的依据，设计科学的销售计划，避免决策失误而造成的商业损失[4]。

电子商务中，进行交易所流通的物品数量巨大，且种类繁多，大量商品交易信息需要得到及时的处理，对此想法进行设计实验时，不仅需要考虑各功能的实现方法，也需要对运算性能进行设计分析。在此情况下，利用大数据开发技术进行设计与实现，可以通过计算机技术来对物品进行快速并且较为准确的分类。既能使此功能实现更科学合理，更加满足了对运行性能的需求。

基于上诉背景与思想，本文设计实现了利用自然语言处理技术中词语聚类方法实现物品词语聚类功能。文中介绍了基于物品数据进行词语聚类的方法分析、设计与实现。总体流程中主要包括数据获取，数据预处理方法，之后选择了基于上下文词语相似度的方法计算物品词语的相似度，最后利用与K-means聚类算法相结合的方法对物品词语进行聚类实现。

本次研究中由于需要加强对自然语言处理知识的了解，并且深入了解两种算法的原理并且用代码进行实现与部分改进。

## 1.2 国内外研究现状

1） 大数据研究现状

互联网时代，每个人都十分依赖从网络获取并且分享信息。互联网的发展促使着物联网，移动互联网，云计算等技术的迅速发展。信息的产生速度和产生量与日俱增，各个企业的数据量已达到PB级。此情况下对数据的处理不能仅仅使用传统的方式，一般的技术已经无法满足对功能且性能的要求。从各个企业数据中可以获得商业信息，根据数据的分析可以进行商业决策。这些数据资产对于各个企业都是十分重要的商业机密，在此情况下，大数据应运而生[5]。

大数据最早提出是在《第三次浪潮》一书中，其作者阿尔文.托夫勒是著名的未来学家，他在本书中赞颂“大数据”为“第三次浪潮的华丽乐章”。目前国内外对大数据的概念尚未统一，唯一统一的便是大数据的数据规模，只有在PB级及以上的数据量基础上，才可以称之为大数据。国内外从各个特征对大数据进行了定义，其中IBM指出了大数据的4V特征，多样性variety，体量volume，价值密度value，速度velocity。数据包括不同类别的数据，如文本数据，音频数据，视频数据。Variety表示数据类型具有多样性和异构性。Volume表示数据量巨大，并且增长速度快。所获取的数据中，可能海量数据中只有一小部分的数据是真正有用的，比如所获取的大量淘宝交易信息中，若进行物品分析，则仅仅需要物品名称。Value表示的是在大数据中数据的价值密度较低。Velocity表示需要对数据进行实时分析，处理速度要快[6]。

对于大数据，国内外的研究在大数据技术方向比对其概念的研究更为重视。在国内，各个高校已经开展了专门的大数据研究中心，各个企业也已经成立了相关的大数据处理部分，对各企业的信息进行挖掘分析。目前国内对大数据的研究应用主要包括，分布式处理，数据挖掘，大数据可视化，云计算等方面。在国外，各个政府对大数据技术的研究提供资金资金支持，如美国政府于2012年3月启动“大数据研究和发展计划”，投资基金为2亿美元。目前多个跨国IT企业已经专门开展大数据分析工作，对大数据领域进行商业研发探索。在国外各个知名高校也已经专门开设数据科学课程，为培养下一代大数据人才提供良好的平台，使数据科学的研究不断壮大。大数据技术的发展也越发迅速，如Hadoop，Spark，MapReduce等技术发展，为进行大数据研究工作提供更加方便的平台[7]。

大数据的研究需要对多个学科的知识都有所了解，如计算机学科，统计学科，信息学科等。如今国内外对大数据的研究与应用十分重视，大数据的4V特征使大数据研究与实现方面都有着巨大的挑战。国内外的学者通过长期的研究工作，在大数据的潮流中，在大数据处理，大数据分析与挖掘，大数据采集，大数据存储，大数据展现与大数据应用技术方面都不断有着突破性的发展与研究结果[8]。

2） 词语聚类研究现状

词语聚类是自然语言处理领域的一个研究方向，聚类是指将不同数据聚合成不同的类别，并且各个类别之间的相似度尽可能的小，而同一个类别中的数据的相似度尽可能的大。而词语聚类表示进行聚类的数据为词语。1963年，《Principles of Numerical Taxonomy》一书促进了聚类的发展和研究工作，本书是有Robert Sokal和Peter Sneath合著的。多年前的聚类主要是基于距离的聚类，如AutoClass方法，当时聚类为统计学的一个分支。而如今，聚类理论与技术都得到了发展，现在，在众多学者的努力下，聚类已经有了多种不同的算法，用于应用与不同的数据与情景。现在聚类方法已成为知识发现的重要手段[9]。

聚类中的词语聚类现如今的研究与应用也十分广泛。词汇是一个相对开放的系统，若要建立语言模型，由于无法确定一种语言有多少词汇，并且语言模型的建立往往难以确定模型时所需的众多参数，由于所获取的语料库往往无法包括所有的语言现象[10]。为了解决这一问题，便考虑在语言模型中引入对已经进行聚类处理的词语所获得的类条件概率。词语聚类便是为解决这一问题被提出。词类是一个相对于封闭的系统，其数量较小，在此情况下，语言模型的构建便更加方便[11]。词语聚类结合了语言学，统计学等知识，应用于智能化检索，文本挖掘等领域。

## 1.3 研究意义及其目标

电子商务使人们的生活变得更加高效便捷，购物不再受到时间和空间的限制，物品具有更低廉的价格，及更多可同时选择的样式，随着时代的进步，更符合用户的需求。电子商务的热潮，既推动我国物流业的发展，也令商务信息的处理研究不断推进[12]。对电子商务的信息进行挖掘分析，可以为用户科学的提供商业建议，使用户更加科学合理的制定商业决策，提高收益。电子商务中主要承载的是物品交易，针对交易的物品，若对商业运营中的物品进行分类分析，可以对不同的客户收发货物品类别进行分析统计，指导下一步销售计划，同时根据历史消息，监测异常情况。

大数据技术于电子商务中的应用越来越广泛，并且技术已经较为成熟，各大交易平台，如京东、天猫等，利用大数据技术实现了针对不同用户，推荐相应商品的功能。在大数据现如今的理论与应用资源十分充足的情况下，可以利用大数据技术实现针对电商物品进行分类的需求[12]。依靠大数据技术，对电子商务交易中的物品进行类别划分，以科学高效的方法，实现物品的分类。

本次研究中，利用大数据应用开发技术，大数据分析技术，设计对电子商务中的物品信息进行统计分析，根据物品名称进行类别划分的方法。考虑到经过处理后的物品数据均为词语，因此计划使用机器学习中词语聚类的相关方法实现区分类别的目标。

## 1.4 研究内容与文章结构

### 1.4.1 研究内容

本文的研究内容包括：电子商务中物品词语的收集，对所收集到的数据进行预处理过程，词语分词的实现，词语聚类的实现与词语聚类精确度与速度测试。

本文解决的问题包括：词语聚类的实现与测试，分布式集群系统的使用。

### 1.4.2 文章结构

本文主体内容分为六章，每一章的内容安排如下：

第一章 引出了本次研究的背景，介绍了国内外研究现状，然后分析了该研究的相关内容及其意义。

第二章 重点分析研究了词语聚类的使用的大数据底层架构。主要是对Hadoop集群与Spark集群进行了介绍。

第三章 对本次研究的关键技术进行了详细介绍，介绍了需要使用的数据预处理技术，中文分词工具和词语聚类的方法。

第四章 设计了本次实验的词语聚类模型，提出其功能和性能方面的需求。根据需求选择对应的词语聚类流程和件架构。

第五章 对设计的各个模块进行实现，包括数据采集模块和数据处理模块。

第六章 进行实验测试。对测试的结果进行分析。

第七章 总结本次词语聚类的研究，提出研究可以改进的部分与可以扩展到的应用。

# 底层架构

高效稳定的底层系统环境可以令研究更具有可靠性，为实验的成功提供保障。本次研究选择的底层架构为大数据处理分析技术中的，Hadoop与Spark集群架构。此架构满足了大数据的处理环境，处理效率的需求，同时也满足了对于大数据的存储技术的需求。若结合二者为一体，可以充分运用二者优势，提高读取数据，处理数据的效率[13]。在此章节中，将对Hadoop架构与Spark架构的工作原理，工作流程进行详细介绍。对于Hadoop架构将主要介绍其中的HDFS架构。

## 2.1 Hadoop

Hadoop是一个分布式系统基础架构，使用java进行开发的一个开源软件框架。可以实现海量数据的处理运行。利用hadoop架构，用户可以在不了解底层架构的情况之下，直接使用hadoop进行数据的分析处理[14]。

在hadoop框架之中，HDFS和MapReduce是最为核心的设计。HDFS为海量数据提供存储，MapReduce为海量数据提供处理方法。但是随着大数据的迅速发展，Hbase与Hive也变得越来越重要。Hadoop的整体架构与各个部分的功能如图2.1所示：



图2.1 Hadoop整体框架

接下来对各个部分进行详细介绍：

1）HDFS

HDFS表示hadoop分布式文件系统。HDFS为整个Hadoop框架提供了分布式数据存储的底层支持[15]。是hadoop框架中十分重要的一部分。之后，会在下文中，对HDFS的原理和读取文件的流程进行详细的介绍。

2）Map/Reduce

MapReduce是由集群中的多个节点共同组成的一个计算框架，用于大数据量的处理运行[16]。MapReduce在下文中也将有详细的介绍。

3）Hbase

Hbase是一个数据库，全称为hadoop database，为开源架构。Hbase与mySQL数据库，oracle数据库不同，Hbase基于列式存储，而传统的数据库是基于行式存储的。

在列式数据库中，文件的存储方式以列为元素存储。在同一个列中的元素的格式都是相近的。列中的元素会存入到相同的块中，于是，在需要读取列中的信息的时候，便可直接读取此列存储到的块。此方式减少了IO，因此当需要进行大量的信息读取工作时，使用列式存储方法，可以提高读取效率。并且同一列数据格式相近的特点，便于数据压缩的实现，节省大部分的存储空间。在行式的数据库中，与列式数据库相反，文件存储方式以行为单位进行写入，存入到块中。当进行数据库的读取时，当提出读取整列的数据文件时，则需要读取每一行数据才可以得到所需信息。这种情况下，数据的IO较大，当数据量非常大的时候，会导致系统负荷过重，无法完成服务，或者效率非常低下[17]。由此可知，列式数据库与行式数据库相比较，由于数据分析中常见的数据读取方式为列式读取，因此，列式数据库更适合于进行数据分析场景下。

Hbase框架的使用场景更适于具有大量的插入同时需要大量的读取的操作。Hbase的更新删除操作也是插入操作，更新操作，表示新插入一行，新的一行中带有新时间戳。而删除操作则表示在有delete标记的一行中，插入新的一行。因此Hbase所有对数据的操作都是查询和插入操作。它通常是以文本文件的形式在硬盘中进行读写操作。在这种操作原理下，其运行效率依靠于硬盘和机器之间的传输效率。具有局限性。

Hbase的查询操作往往支持的是较简单的key-value形式的查询，但是如果需要特定条件查询，Hbase只能在一般的主键上进行查询，无法实现复杂的查询工作。

Hbase可以保存相同文件在不同时间戳的不同版本，数据库中允许进行冗余情况。Hbase的这个特点使其经常应用于网站中，因为可以查询出不同时间段所浏览的网页，写的博客等[18]。

Hbase适用的场景具有许多局限性，若需要进行大量的读取，写入工作时，其可以提供十分高效的支持。

4）Hive

Hive为hadoop的数据仓库工具。Hive与Hadoop中提供了查询、处理、分析的功能。Hive可以将hadoop中的数据文件转换为hive中的一张表。Hive提供了较为完整的类sql查询方式，同时任务的运行也通过shell，JDBC等接口，实现了sql语句转化为MapReduce任务。

Hive的数据存储于HDFS中，对于数据的存储没有固定的格式。Hive中的表的组织非常自由，不需建立索引，解析数据只需提供创建表时的列和行分隔符便可。Hive中包含了四种数据模型，分别是：Partition，Table，Bucket，ExternalTable。Table是Hive中的表，每一个Table都有一个对应的目录，在目录下存储着对应的Table信息。Partition对应于表的目录，与传统数据库中的索引是类似的。Bucket是为了方便并行设计的，主要功能是计算hash，并且根据计算出的hash值切分数据。ExternalTable可创建Partition，指向HDFS中的数据。Hive提供了进行数据交换的多个接口，在Hive中大部分的查询都是通过MapReduce完成的。

5）Pig、ZooKeeper、Chuwa

Pig为大数据的程序编写操作和并行计算提供了简易的接口和操作。Zookeeper是一个可调整扩展的，高效系统。Chuwa是一个由yahoo贡献的集群监控系统。

Hadoop各个主件之间依赖共存，以多种接口相同连接通信，在进行大数据的处理分析时，进行协调分工，保证系统的稳定与高速[19]。

### 2.1.1 HDFS架构

HDFS的全称是Hadoop Distributed File System，为Hadoop系统提供了文件分布式存储的底层支持。

HDFS是主、从结构。在HDFS中往往是一主多从的方式。整个系统中包括多个节点，其中所有的节点分为了NameNode与DataNode。NameNode为集群中的主节点，主要负责对于文件的访问，系统文件的命名空间，对DataNode的调度等。DataNode主要负责数据的存储管理，负责数据在客户端的读写。DataNode在NameNode统一的调度下进行数据处理操作[20]。NameNode负责管理HDFS中所有的元数据，然而，一般的用户数据是不经过NameNode的。

数据被分为多块被存储于DataNode中，DataNode中各个数据块的映射由NameNode负责。

HDFS的体系架构如图2.2所示：



图2.2 HDFS体系架构

图中的NameNode是HDFS主服务器，DataNode是HDFS的子服务器，Client为需要请求HDFS服务的客户端应用程序。

接下来介绍HDFS读取文件流程。

HDFS读文件过程如图2.3所示：



图2.3 HDFS读文件过程

1. 初始化文件系统，客户端使用相关函数打开文件系统中的文件。
2. 文件系统读取元数据节点中的数据块信息，获得需要的数据块的数据节点地址。

3）文件系统返回用来读取数据的信息给客户端。

4）客户端读取数据。其中首先读取的数据节点是最近的保存文件第一个数据块的数据节点。

5）与第四个步骤类似，当读取的数据块信息读取完毕后，将连接保存着下一个数据块的最近的数据节点。

6）所有数据读取完毕，文件系统关闭。

在读取文件的过程中，倘若出现连接的数据节点出现问题的情况，则此数据节点将会被记录。在进行下一次的读取工作时，将不再连接此节点。

HDFS写文件过程如图2.4所示：



图2.4 HDFS写文件过程

1）文件系统初始化，客户端创建文件。

2）在保证客户端拥有创建新文件并且文件名不重复的前提之下，文件系统调用元数据节点，与命名空间中创建新文件。

3）文件系统返回客户端写数据的消息后，客户端开始写数据。

4）数据以块的方式写入，元数据节点分配数据节点。之后于各个数据节点中写入数据块。由于各个数据块的存储往往是有备份的，且默认复制3份，因此当第一个数据节点写入数据块后，之后数据块再从第一块数据节点复制到第二个数据节点，最后再从第二个数据节点发送到第三个数据节点。

5）文件系统等待数据节点发送写入完毕的确认信息。

6）写入数据结束，调用函数。

7）队列中的数据节点向元数据节点发送确认信息，确认数据已写入[21]。

在HDFS中，读写文件都是首先需要客户端提出读写文件的请求，NameNode根据需求，向DataNode分配任务，客户端完成读写任务之后，调用close函数结束读写。

在文件系统，HDFS将每个数据块默认备份三份。分别备份于三个不同的DataNode中，其中一个DataNode是由NameNode指定的，还有一台是与NameNode指定的DataNode不相同的DataNode中，最后一台是与指定的DataNode于同一Rack上的DataNode。数据备份提高了数据的安全性，减少Rack的失败对数据造成的损失。在性能上，分布式系统考虑到，当需要读取整块硬盘时，往往需要耗费大量读取时间。仅仅一块硬盘的存储和传输能力有限，若一台独立物理计算机的存储不能够满足所需要存储的数据量大小的时候，则需要将文件分区存储到多个独立的机器上。

HDFS适用于通用硬件，容错性强。并且HDFS以流处理访问模式进行文件存储。用户在读取数据时，不需要考虑到存储地点与数据类型，每一个数据在系统之中都可以找到对应的本地映像[22]。HDFS满足高吞吐量的需求，但是对于时延没有办法保证。HDFS适合超大数据量的应用。

### 2.1.2 Map Reduce模型

MapReduce是一个计算框架，其表现形式就是，通过信息的输入，经过一定的计算模型处理之后，将处理结果输出，所输出的内容便是用户需要的内容。

Map表示映射，是指将一个较为复杂的任务分为可以单独执行的多个较为简单的任务。Reduce表示规约，是指将map输出的结果进行整合处理。MapReduce思想为：分而治之、大事化小[23]。

MapReduce的工作流程如图2.5所示：



图2.5 MapReduce运作机制

此图中可以看出，MapReduce的运作机制总共包括了四个实体，分别是客户端，Master，Worker，HDFS文件系统。

其中各个实体的分工如下：

1）Client：即客户端，负责提出需求，分配任务。

2）Master：负责任务的初始化工作，与Worker相互通信，协调任务的完成。

3）Worker：与Master进行通信，执行分配的任务。

4）HDFS：文件系统负责的是数据的存取工作，HDFS中不仅存储着与任务相关的数据，并且存储着系统相关的配置文件[24]。

MapReduce任务通常是，首先根据输入的数据计算需要分片的数量，每个分片对应一个map任务。于map阶段，将数据集切分为数据块，分别进行数据处理工作，将处理结果传入Reduce阶段。MapReduce的处理阶段中的都是在文件系统中进行数据的读取与存入操作。在一般情况下，MapReduce的处理的机器与HDFS进行存储的机器是在相同主机上的。因此充分的利用了系统的带宽，并且可以在已有框架上高效的完成任务。

MapReduce提高了数据处理的运行速度，增大了吞吐量，具有高容错性。现在已经被广泛的应用于数据处理中。但是MapReduce在现阶段的使用中，仍然存在一些在局限性：MapReduce是针对海量数据设计的并行处理机制，在进行运行处理过程中，通常会消耗大量的时间进行数据处理，在此情况下，数据的一次处理过程成本较大。因此，MapReduce任务对于导致错误的操作都是零容忍的。进行MapReduce处理的机器中，由于往往只设置一台为Master。此情况下，倘若在Master机器中出现了单点故障的问题，将导致MapReduce任务执行出现严重失误。

## 2.2 Spark框架

Spark是与Hadoop类似的一个集群计算环境。Spark集群与Hadoop相比，Spark的一些特点更适合于多次迭代的MapReduce计算任务。在进行MapReduce计算时，任务中所产生的中间结果，不必通过HDFS进行存取，可以直接保存在内存中，计算过程更加高效。Spark的工作负载在某些工作之中比Hadoop更加优越。

Spark是基于Scala进行开发的，二者紧密集成。Scala操作数据集就像操作本地集合对象一样轻松。Spark在迭代计算方面的优势，可以说是对Hadoop的计算进行的补充。Spark集群环境可以与Hadoop集群相结合使用[25]。

Spark的核心是RDD，弹性分布数据集。RDD是Spark进行运算的基本单位。RDD提供了输入算子，转换算子，缓存算子和行动算子四种算子。其中输入算子负责将原生数据转换为RDD。Spark应用中，执行流程在在逻辑上会形成DAG，转换算子是生成DAG图的对象。缓存算子主要功能是对需要多次使用的和重要的数据进行备份缓存。行动算子是将运算的结果转换为原生数据，运算的结果为RDD。

接下来对Spark的四大模块和模型进行介绍。

### 2.2.1 Spark模块

Spark在底层基础上，支持的模块有：Spark SQL模块，Spark Streaming模块，MLlib模块和GraphX模块，这四大模块。如图2.6所示：



图2.6 Spark基本模块

1. Spark SQL

Spark SQL中的表数据使用的是列式存储方式，这种存储方式，当进行大型数据存取时，在吞吐量和空间占用量都具有优势。Spark SQL提供了多种交互方式，包括DataFrames API，Datasets API和SQL等，所有交互方式都有对应接口，并且运行方式都十分简单。Spark SQL各交互方式的内部运行引擎是相同的。

Spark SQL的核心是DataFrame，DataFrame可以将JSON，RDD，HDFS等格式的数据加载为表进行查询[26]。读取相应数据后，数据过滤，数据整合等对应数据的操作都是十分方便的。Spark SQL可以实现利用读写数据进行分析的自动化工作。

2）Spark Streaming

Spark Streaming为Spark提供了对流数据进行处理的框架，Spark Streaming中输入实时流数据，数据按照时间片被切分为每个小块，之后使用类似批处理的方式进行数据的处理。Spark Streaming高效且容错性强，会备份每一批输入的数据，当计算的节点出错时，可以通过调用其他节点备份的数据继续数据处理的工作[27]。Spark Streaming拥有基于内存的高速计算引擎与大量的API，这种条件，使用户可以结合交互查询，流式处理等应用。

1. MLlib

MLlib为常用的机器学习算法提供了实现，MLlib中含有统一的接口，这种方式，令机器学习算法的调用十分方便[28]。

1. GraphX

GraphX为Spark进行图处理提供了支持，其中包含了大量关于图的算法。

Spark的四大模块为Spark对于各种数据的处理提供了良好的支持。实际应用中，根据不同的任务需求，使用其中一个或多个模块进行数据的分析处理工作。

### 2.2.2 Spark运算模型

Spark应用程序的执行过程如图2.7所示：

1）用户编写Driver application程序。

2）用户程序使用SparkContext进行任务的提交。Spark Context向Cluster Manager请求资源。

3）Cluster Manager向Worker分配任务。Worker任务完成后会将结果返回SparkContext中。



图2.7 代码执行过程

上文中对Spark集群进行了相关介绍，Spark集群与Hadoop集群相比，在特定情境下，比Hadoop更具优势。Spark集群通过使用内存计算，速度为基于磁盘运算的Hadoop集群的100倍。Spark框架设计令运算中的IO减少，减少时间延迟，因此比Hadoop更适合于对时延有要求的项目中。除此之外，Spark同时提供了大量资源库与相关接口，为用户提供了对各种数据类型进行计算的方法[29]。

## 2.3 本章小结

在本章中，主要针对大数据分析处理的相关框架进行了详细介绍，主要包括Hadoop框架和Spark框架，其中对Hadoop框架中的两大核心，HDFS与MapReduce进行了介绍。Spark框架中主要介绍了各大模块和其运算模型。Hadoop与Spark框架都有其独特的优势，在具体的大数据分析和开发中，应当针对实际情况，选择合适的框架。

# 关键技术介绍

为了词语聚类的实现，本章节对数据预处理，中文分词工具与词语聚类的多种方法设计进行了分析介绍。

## 3.1 数据预处理

进行数据分析时，许多成熟的算法对运行的数据集合都有一定的质量要求，为了数据分析的结果更加具有说服力，干净准确的数据是必不可少的，因此数据预处理是数据分析十分重要的一个环节。

干净简洁的数据是进行数据分析的必要条件。目前被广泛接受的数据质量标准包括：数据的完整性、准确性、合时性、一致性、解释性、可信性。但是一般情况下，由于获取数据方法，或者数据输入错误等失误，令所得数据集都是“脏”数据，这种情况通常是不可避免的[30]。所获取的“脏”数据可能会出现下面几种情况：数据缺失，即所获取数据集中部分属性值，或者所需要的属性值有所缺失等；数据有噪声，离群值，和错误数据；存在冗余数据等。对于这些数据若不经过处理，直接进行数据分析与数据挖掘时可能会导致分析结果不准确，使决策失误。因此数据预处理的重要性不言而喻。

数据预处理包括：数据清洗，数据集成，数据变换，数据规约等。当针对不同的业务，获取了需要的数据资源后，便可以针对业务需求，处理获得的数据，为下一步工作做准备。接下来，主要对数据预处理中的数据清洗，数据变换，数据集成，数据规约的原理和方法进行介绍。

### 3.1.1 数据清洗

数据清洗主要工作是负责填补缺失数据，平滑噪声，识别离群值，处理冗余数据，完成一些数据类型的转换等[31]。

数据清洗可以分成无监督和有监督两种方法。无监督方法是指使用样本数据训练算法，使用机器学习的方法，令其获得一定的经验，之后再根据这些经验继续自动清洗其他数据。有监督方法是指在领域专家的指导之下，根据所获数据，分析数据，填补缺失值，去除冗余数据，去除明显的噪音数据等。

针对缺失数据，含噪声数据，冗余数据，错误数据，所使用的清洗方法如下所示：

1）缺失数据

由于设备故障，人为录入信息或修改数据出错等原因，可能会导致数据丢失，数据不可用的情况发生。

对于缺失数据的处理，当数据量较小时可以使用人工填充的方法，但是通常情况下，人工录入可行性较低，工程量过大。当人工填充不可行时，填补空缺值可以使用一个全局变量填充，如使用“unknown”等，也可使用某属性的平均值进行填充。更为科学的方法是，使用判定树或者贝叶斯公式等基于推断的方法或者回归的方法，推测出最有可能的值进行填充。

1. 噪音数据

噪声是指一个测量变量中的随机错误或误差，是与其他数据不一致的数据。由于命名规则不一致，技术限制，收集工具等原因，可能会导致数据中含有噪声

针对噪声数据，目前数据平滑技术是应用的最为广泛的方法。此类数据除非在经过专业的分析确定其是无用或者删除对于之后的操作无影响的之后，才删除处理。可以使用聚类，回归，分箱等方法进行处理。聚类方法是指使用聚类技术，检测出孤立点并修正，一些数学方法如粗糙集，灰色数学等方法可以与聚类技术结合起来检测孤立点。回归方法表示拟定回归函数，让使用数据适应函数来平滑数据。分箱法是根据相邻的数据得出数值。使用分箱方法需要对数据进行排序，并且以一定的规则分到一些箱子中。之后可以按箱的平均值、中值，边界值平滑数据。除此之外，使用人工与机器相结合的方法，进行噪声数据处理，可使用机器检测出可疑的孤立点，以人工确定。

3）冗余数据

数据冗余包括了属性冗余和属性对应数据冗余两种情况。可以使用因子分析或者根据经验，计算属性之间的相关性，使用数学的方法，判断部分的属性是否足够表现问题。倘若部分的属性已经足以满足信息的挖掘，则可以删除多余属性。但是经过分析，当冗余属性对后续工作有帮助时，可以对冗余属性进行备注保留。

4）错误数据

对于错误的数据。可以结合实际需求，删除，更新数据。也可结合数学方法，寻找约束函数。根据历史的数据，更正错误数据[32]。

数据清洗是数据预处理中工作量最大的一个步骤，针对错误，缺失，冗余，以及噪音数据都有相对应的十分科学的方法进行清洗工作。提高数据质量。

### 3.1.2 数据变换

数据变换的工作主要是将数据转换为统一的适合挖掘的形式。将维度较高的数据经过维度变换或者转换的方法，转化为维度较低的数据，消除数据在时间、空间、精度以及属性之间的差异。

数据变换涉及如下内容：

1）平滑处理

平滑处理的过程中主要是去噪工作。使用的方法有回归和聚类方法。

2）聚集处理

数据聚集处理过程将数据汇总。如“年-月-日”这类不同估量单位的聚集处理。

1. 数据泛化处理

数据泛化处理是指将具体的低层的概念表示成抽象的高层的概念。如：“颜色”属性可以向上泛化为“视觉效果”属性。针对数值型的属性，可以通过不同表达方式映射到更高层概念。比如“商品购买价格”属性可以映射到“平民，昂贵，奢侈品”。数据泛化处理操作可以降低数据复杂度。

4）数据规范化

数据规范化也可以称为数据归一化。数据规范化将数据缩放到较小的范围区间之中。数据规范化的方法有：小数定标规范化，最小-最大规范化和z-score规范化。

规范化令原始数据以一定的比例缩放，数据的变化很大，尤其是小数定标规范化和z-score规范化的方法。为了以后也可以按照相同的方式进行缩放数据，有必要保存规范化参数。

5）属性构造

属性构造是为了增加对高维数据的结构的理解。操作中在已有的属性之中构造出新的属性，并且存入已有属性集。

数据变换的方法的实用性比较强，虽然数据变化对原始数据的损害较大[33]。但是针对不同的研究目的和数据属性的不同特点，选择不同的变换方法，数据变换可利用较小的变量获得了数据的最大变化。

### 3.1.3 数据集成

数据集成是指将不同文件或者不同数据库的数据集合到一致数据存储中。数据集成过程需要合并处理不同数据源的异构数据，解决数据的冲突与不一致。清除语义的模糊性。

不同数据源的数据经常存在着异构性。如命名、单位、含义不一致，字段之间的单位、字长不一致，同名异义等情况也时常发生。数据集成时需要解决这些问题，需要对数据进行规范和统一，统一原始数据的所有矛盾之处。同时，在存储数据时，需要对数据类型进行分析选择，选择存储空间最小的数据类型。

数据集成有两个主要问题：

1. 模式集成和对象匹配，实体识别问题

在匹配不同数据源中的元数据时，由于命名规则不同，导致不同数据源对于同一物体的命名通常是不同。例如：A.student=B.stu。在匹配现实世界的同一实体时，由于不同的表示，同一属性的不同属性值可能不同，例如Bill Clinton=William Clinton。来自不同数据元的数据这种情况通常是不可避免的。

1. 冗余

当集成多个数据源时，可能会出现不同数据源中存在相同属性或对象使用不同字段名称。或者出现存在属性或对象可以由其他的属性或者对象推导出来的情况。这便是数据集成时发生的数据冗余情况。这种情况可以通过数据相关性分析检测出来。通过统计分析方法，分析出属性的相关性，计算公式为：

其中A，B表示检测的属性值，n表示属性的数量，和表示属性的均值，，表示的是其标准差。当计算出值为0时，表示两个属性不具有相关性，不是冗余的。而当大于0，越接近于1表示属性越接近，小于0，越接近于-1，表示两个属性相互影响越大，且成反作用。

仔细的将数据集成起来，减少数据集成发生的冗余情况，科学检测出不同表示方式表示的为同一对象。可以提高数据挖掘的质量和速度。数据集成根据不同的学科有不同的规则与学科依据，相对而言，数据集成是数据预处理中较为困难的步骤。

### 3.1.4 数据归约

由于进行挖掘所需的数据集巨大，若对此海量数据进行数据挖掘，所耗费的时间和资源往往也是十分巨大的。如果能够从所获取的海量数据集中找到规约表示，在保证数据完整性的条件下，根据任务需求和数据本身的特征，找到可以表示这些数据的特征，减少数据的存储空间[34]。

数据规约的方法有：

1. 维归约

维归约表示通过特征选择或者启发式的方法，从所有的属性中删除任务不需要的属性或者维度。特征选择减少属性数量，在属性集中找出可以表示完整属性集的最小属性集。使得所得数据类的概率分布尽可能的与原数据的概率分布相同。启发式的方法包括了：判定归纳树，逐步向前选择，逐步向后删除，向后删除和向前选择相结合的方法。

2）数据压缩

数据压缩方法包括了字符串压缩，音频/视频压缩，分形技术等。数据压缩方法可以分为有损压缩和无损压缩方法两类。有损压缩包括了主要成分分析和小波变换。字符串压缩通常为无损压缩，有着精妙的算法和广发的理论，音/视频压缩中的压缩精度可以递进选择，通常为有损压缩。

3）数值规约

数值规约减少数据量的方法是通过选择较小的，可替代的数据表达形式。分为有参方法和无参方法。有参数方法中常用多元回归，线性回归方法等。有参方法便可除了部分离群值外不用存储数据，最后结果只需存储参数。无参方法常用的是聚类，选样，直方图等方法。无参方法和有参方法的区别在于是否使用模型的方法。无参方法不使用模型方法存储数据，而有参方法使用参数模型估计数据[35]。

除了以上的三种规约方法，数据规约还可以使用数据立方体聚集，离散化和概念分层的方法。

业务中，所获取的数据中的属性有些并不是完全需要的，加入这些属性甚至可能对影响分析结果。因此使用数据规约的方法，对数据属性做出相应处理，以最大程度的缩减数据，既能保证任务结果，也可以提高运行效率，减少资源的耗费。

数据预处理中的数据集成、数据清洗、数据规约和数据变换四个步骤在实际的应用中，并不是完全分开的。四种方法相辅相成，针对实际目的，处理所获数据。在进行数据预处理后，倘若之后的挖掘结果不合理，与期待不一致，在排除了数据挖掘过程中操作不合理都情况后，若对使用的数据有所怀疑，可以进行二次数据预处理工作。已修正第一次数据预处理中的失误。但是倘若此次挖掘结果仍然与预期不符，则需要斟酌多方原因。数据预处理过程中，应当以人工和机器相结合的方法，以人工经验结合实际环境与需求判断所使用的预处理方法，结合机器完成预处理工作。尽早处理成适合的数据，高质量的数据为之后的数据挖掘与数据分析工作奠定良好的基础，为高质量的决策提供理论依据。

## 3.2 中文分词工具

在进行文本挖掘，词语聚类时，通常需要根据语料中每个词语上下文的语义关系，推断对应词语的相近程度。在使用搜索引擎时，当于搜索引擎中输入对应词语，输出的结果往往是按照与本词语相关性由高到低排列的[36]。因此中文分词是进行文本挖掘，文本检索必不可少的一步。

中文分词表示将文章，句子或者稍微长一点的词语，切分为多个词语。中文分词属于自然语言处理的范畴。进行中文分词时，由于与英文词语直接通过空格或者标点切分的方式不同，中文的词语并没有十分明显的切分，并且中文词语的语义多样，中文词语具有歧义性和不断更新的特点，中文词语这些特点导致个人的中文学习都十分困难，若要令机器学会这些知识，几乎是很难实现[37]。中文的种种特点导致中文分词的实现较为复杂困难。但是在自然语言处理方面，针对中文分词，目前已研究出相应的中文分词工具。

接下来对几款中文分词工具进行介绍：

1）ICTCLAS

ICTCLAS是中国科学院计算技术研究所开发的一个最早的中文开源分词项目之一，全称为Institute of Computing Technology, Chinese Lexical Analysis System。ICTCLAS被称为全球最受欢迎的中文分词系统。

ICTCLAS支持中文和英文的分词，分词精度达到了98.45%，开发语言使用的是C/C++。支持的操作系统有：FreeBSD，Linux，Windows。此系统的主要功能包括：词性标注，中文分词，新词识别，命名实体识别。系统在词典及其扩展性上，可以单条，也可以批量导入用户词典。不仅支持繁体中文，同时支持多种编码格式，如：UTF-8，GBK，UNCODE等。

ICTCLAS原理是使用了层叠隐马尔科夫模型，一共分为了五层。通过分层的方法，以保证分词的精度与效率。

ICTCLAS性能上分词效率约50万字/秒，如今也在不断地进行开发改进，应用范围广，分词效果出色[38]。

2）MMSEG4J

MMSEG4J是基于Java的开源中文分词组件，提供lucene和solr 接口

MMSEG4J用 Chih-Hao Tsai 的 MMSeg 算法实现的中文分词器，并实现 lucene 的 analyzer 和 solr 的TokenizerFactory 以方便在Lucene和Solr中使用。MMSeg4j支持的语言有中文，中文中包括了一些字符的处理，还支持数字，英文，希腊，俄文[39]。

MMSeg 算法有两种分词方法：Simple和Complex，都是基于正向最大匹配。Complex 加了四个规则过虑。官方说：词语的正确识别率达到了 98.41%。mmseg4j 已经实现了这两种。MMSEG4J支持自定义覆盖词典，其所用的词库为sougou词库。

3）IKAnalyzer

IKAnalyzer分词工具是轻量级的中文分词工具包。它是基于java开发的一个开源分词工具。在最初的开发中，它以开源项目Luence为应用主体的，结合词典分词和文法分析算法的中文分词组件。之后的开发中，便发展了独立于Lucene 项目且提供对Lucene的默认优化实现的公用分词组件[40]。

IKAnalyzer支持的语言包括：中文，数字，英文，韩文字符，日文字符。IKAnalyzer支持自定义停用词和自定义用户词典的扩展，它的词库中收录了27万中文词汇。

IKAnalyzer采用了多子处理器分析模式，它的原理使用了正向迭代最细粒度切分算法。性能为83万字/秒。

## 3.3 词语聚类方法

词语聚类表示根据词汇语义，语用等特征，将词语聚合成一个又一个簇，聚类结果使得各个不同的簇之间的相似度尽可能小，而同一个簇内的词语相似度尽可能高。词汇是可以单独运用的最小单位，在语音识别与信息检索方面，词语聚类的应用十分广泛而且十分重要。

词语聚类的步骤一般是分为两步，首先词语聚类的前提是计算出各个词语之间的相似度或者进行词语的特征计算。之后再根据词语特征与需要实现的聚类效果，选择聚类方法，实现聚类[41]。

词语聚类的方法分为两种，第一种为，基于规则的词语聚类方法，第二种为基于统计的词语聚类方法。

1. 基于规则的方法

基于规则的聚类方法也称为基于知识的聚类方法，在这种方法中一个十分典型的例子是依照词性进行分类。这种方法是凭借着词语的语法或者语义特征进行聚类的。但是在早期的研究中发现，这种方法虽然在有特定领域的知识的情况下，根据语法进行聚类，将语法功能相近的词语聚成一类的时候，这种方法的聚类效果比较好。但是一般情况下，这种方法会提高模型的复杂度。

2）基于统计的方法

基于统计的方法为数据驱动的方法。这种方法是根据所获得的语料库对词语进行聚类的方法。此方法不需要考虑到词语的语义，语用和语法等特征。降低了模型的复杂度，只依照语料库的统计信息对词语进行聚类实现[42]。这种方法使得机器自动对语料库中的词语进行聚类成为可能。

除了上述两种常用的方法，由于词汇是一个十分开放而且复杂的系统，每一个词语不仅仅存在着一个含义。中文的一词多义，使词语聚类需要扩展更多方面。词语聚类根据每个词语分的类别个数又可分为明确聚类和模糊聚类。明确聚类表示一个词语仅仅分到一个类别之中，然而模糊聚类则可将一个词语分到多个类别。模糊聚类方法使词汇的分析更加科学细致，但也增大了词语聚类的难度[43]。

词语聚类在自然语言处理方面应用十分广泛，而且知识较为基础。目前词语聚类的研究对于词语的特征并未完全挖掘，建立模型也并非十分完整。未来对于词语聚类的研究可以考虑到词语更多的特征，使聚类结果更加智能，更具参考性。

# 词语聚类模型设计

本文对词语聚类的总体实现流程设计主要是，数据预处理，中文分词处理，词向量的转化和词句聚类的实现。

本文针对所获取的数据和所需要实现的效果，选择的词语聚类方法，是基于语义特征的词语聚类，表示按照语义相近原则，将词语聚成不同的词簇。所用的语料为商品词语，对获得的商品语料进行处理聚类。

## 4.1 需求设计

### 4.1.1 性能需求

此次数据量为20万数据，因此采用大数据分析方法进行数据分析处理，传统的处理方式响应时间过长。处理性能是功能实现的基石，针对较大的数据量，如果无法保证处理能力，可能会面对无休止的等待[44]。对于应用的大数据技术，本流程中所需要的性能需求如下：

1）本次数据处理中，保证处理过程流畅不出现卡顿的情况，为了保证处理效率，使用分布式的处理方法，决定使用Spark集群进行数据处理，各个节点的部署扩展方式简单，并且满足动态扩展。

2）支持分析处理打的数据量，进行一次分析处理的时间应当控制在20分钟以内。

### 4.1.2 功能需求

实现词语聚类主要分为三个模块，分别是：数据采集，数据处理，词语聚类。

1）数据采集

获取数据是进行大数据处理的前提。数据采集是根据实验需求，有目的的收集特定数据的过程，采集方法既可以通过数据库也可以通过其他媒介。本次词语聚类由于主要目的是对电子商务领域的数据进行处理分析，因此获取的数据为电商数据。采集的数据来源于网络，是使用网络爬虫技术爬取的天猫双十二商品数据，数据来源于真实数据并且具有时效性。

网络爬虫可以实现自动的根据一定的规则，从特定页面入手，通过不断地跳转到其他页面，从页面中抓取信息。数据采集过程采用爬虫技术从活动主页面入手，首先爬取了商品的ID信息，之后根据ID信息获得了其他信息。

2）数据存储

获取到的数据为json文件，在处理过程中，将数据存储备份与HDFS。程序运行时，通过HDFS读取文件，便于之后的使用。

3）数据处理

数据处理是进行数据分析的重要步骤。根据研究需求，对获取的数据进行数据清洗，变换。使数据经过处理后，满足进行下一步实验的要求。一般来说，数据处理是数据分析过程中最耗费时间的流程。高质量的分析离不开高质量的数据，因此数据处理部分需要针对本次所获取的数据特点，处理成所需数据格式与数据内容。

本文根据所采集的数据设计了数据处理方案，由于主要需要的是物品词语数据，因此，从所得语料中抽取出商品名称，并且保存为文本文件输出，之后使用分词工具将文本数据进行分词处理。

4）物品聚类

物品聚类是对物品类型进行识别，并将同属一类的物品集合到一起。从而得到不同类型的物品在快递流通中的情况，并了解某段时间或者某个时间点的物品热度，也可以了解到不同地区的主要流入、流出物品的类型。

数据处理后得到了已经分好词的数据，在此基础上进行词向量转化与词语的聚类实现。并将输出数据保存于文本文档之中。

## 4.2 架构设计

在研究中，离不开对系统架构进行设计。一个完整的系统架构包括了软件架构和硬件架构。选择并构建一个合适的架构以满足研究中对性能和功能的需求。一个好的硬件架构保证了实验中性能最优化，一个好的软件架构，在硬件架构的基础上，发挥硬件的最优效果，保证实验效果的最优化。

### 4.2.1 硬件架构

针对物品词语聚类所设计的硬件架构如图4.1所示，整个结构由两大部分组成：Spark 部分、HDFS部分。



图4.1 硬件架构

本研究中选用了Spark与Hadoop的一体机集群，作为本次数据分析系统底层架构。本次都于相同的机器上部署Spark与Hadoop集群。部署的Spark集群包括了Mater和Slaves。其中仅一台Master，其他均为Slave节点。部署的Hadoop集群与Spark集群部署情况相同。选用了Spark集群的Master作为Hadoop集群的Master节点，在此情况下，倘若部署了Spark与Hadoop的机器出现故障。其中，若出现故障的机器是Slave节点，在其他机器数量满足性能需求的条件下，可以不需要处理故障，继续处理数据。但是倘若出现故障的机器为Master节点，则需要考虑是否选择另一台机器，于所选机器中配置Spark与Hadoop集群环境，并且与其他Slave节点配置无密码连接，作为集群中的Master节点。也可以选择Slave节点中的一台机器，修改配置文件，将其配置为集群的Master节点。这种硬件架构可以允许配置为Slave节点的机器出现故障，并且不影响系统的使用。然而若Master节点出错，则系统便无法继续使用。

选择Spark与Hadoop集群作为硬件架构，使各个节点之间通过网线连接，便可同时接收到Master节点的指令。同时，此硬件架构扩展性较强，当性能需求不被满足时，可以扩展多个Slave节点进行分布式处理分析。将Spark与Hadoop集群搭建到相同的机器中，保证了硬件资源的合理利用，不造成硬件资源的浪费[45]。这些配置将两个大数据分析系统与大数据存储系统相结合。在实际应用中，选择合理的使用方式充分挖掘利用两大系统的优势。

### 4.2.2 软件架构

针对词语聚类所设计的软件架构如图4.2所示，整个架构由5个部分组成：



图4.2 软件架构

系统层，图中的Linux OS便是系统层，处于最底层。Linux OS支持大数据处理技术与大数据系统架构。Linux OS安装便捷，稳定可靠，并且是开源免费的系统。由于Linux OS的这些特点，便选择了Linux OS作为底层系统的基础。配置的Spark与Hadoop集群系统的各个节点，均是在Linux OS环境下搭建的。Linux OS提供了很好的支持，使安装与配置过程十分便捷，使集群的配置与运行十分稳定[46]。其中Spark与Hadoop集群的部署都是采用的一主多从的部署形式。

数据存储模块，本研究中数据存储主要使用的是HDFS进行数据的存储。在进行数据分析的时候，客户端从HDFS中读取数据，之后将数据进行数据处理，处理结果重新存入HDFS。

数据处理模块，数据处理是研究流程中最为核心的一个部分，数据处理主要是应用Spark集群系统实现分布式数据处理。Spark集群系统与HDFS相结合，Spark集群系统通过HDFS进行数据的输入与输出。

应用层，应用层主要是面向使用者与程序编辑者的。应用层使用了常用的Windows系统进行程序的开发，并且在程序运行的过程中，配置了以Windows作为客户端，Spark作为服务器的系统环境。在Windows中，配置系统环境与程序编译器，使其与Spark与Hadoop集群可以进行通信。于Windows下配置spark local模式，进行Spark程序的编写与调试，并且直观地显示程序输出结果，方便了开发人员的开发调试工作。

## 4.3 流程设计

系统流程主要按照数据的流向进行设计，首先为数据的采集存储；获取数据后，便进行数据的处理。数据的处理过程，首先是进行数据预处理，对原始数据进行数据清洗，提取出所需的物品数据，之后对提取出的数据进行分词，将分词后的数据存储于HDFS之中。最后使用相关算法进行词语的聚类实现。系统流程设计如图4.3所示：



图4.3 总体流程设计

## 4.4 本章小结

本章节对物品词语聚类具体的实现过程进行了设计，分析设计了物品聚类的具体流程，说明了系统架构性能与功能需求。

# 物品词语聚类实现

第四章中对于物品词语的聚类过程进行了详细的设计，并且提出了系统环境的需求与总体分析流程。在本章节中，将对上一章节中设计的流程进行详细设计。词语聚类主要分为了两个大模块，分别为数据采集模块和数据处理模块。

## 5.1 数据采集模块

本次所采集的数据是天猫双十二的活动数据，数据量为23万条数据。获取的数据中包括商品ID、商品名称、商品网址、商品价格等详细信息。数据格式如表5.1所示：

表5.1 原始数据结构

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **变量名** | **数据格式** | **示例** |
| 商品网址 | url | VARCHAR2 | https://detail.tianmao.com |
| 优惠活动 | discount | VARCHAR2 | 全天猫实物商品通用 |
| 商品ID | id | VARCHAR2 | 539050274417 |
| 服务保障 | guarantee | VARCHAR2 | 闪点到家，正品保证 |
| 原价 | priceOriginal | FLOAT | 1999.00 |
| 活动开始时间 | timeBegin | VARCHAR2 | 2016-12-12 00:00:00 |
| 快递费 | priceExpress | FLOAT | 0.00 |
| 卖家地址 | addSeller | VARCHAR2 | 广东广州 |
| 会场 | conference | VARCHAR2 | 电脑办公会场 |
| 现价 | priceNow | FLOAT | 1079.00 |
| 活动结束时间 | timeEnd | DATE | 2016-12-12 23:59:59 |
| 标题 | name | VARCHAR2 | 现货Xiaomi/小米 小米平板3代 64G轻薄wifi学生平板电脑 |

## 5.2 数据处理模块

数据处理模块是系统的核心模块，它主要工作在三个方面：数据预处理、中文分词处理和词语聚类。下面是对三个方面实现细节的分析说明。

### 5.2.1 数据预处理

进行数据处理的第一个步骤就是将采集到的数据进行数据预处理，为接下来的分词工作做准备。所获得的数据由于只需要商品名称，因此采集的数据大部分在本次研究之中是无用的数据。针对这组数据，根据数据清洗，数据变换，数据规约，数据集成中的相关方法，设计了符合需求的数据预处理方法。

1）数据清洗

采集到的数据量较大，因此在采集数据的过程中，可能由于采集方法或者数据存储出现失误，导致少数数据存在缺失，或者不合规范的情况。

在本研究中，每条数据都代表了一次天猫交易。本阶段所进行的数据清洗就是识别出不规范的信息，将有缺失值的数据进行清理，如果发现有缺失或者与大部分数据格式不相同的数据。考虑到出现这种情况的数据量占比非常小，并且对总体数据运行所得出的结果没有造成影响。因此，为了节省时间，方便数据清洗操作。所选用的策略是直接删除不合规范的数据，经过此处理既可以为后续的数据预处理工作节约时间，降低后续工作的复杂度，也可以达到提高数据处理效率和节约存储的目的。

数据清洗后，所得的数据都是统一格式，并且内容相一致的数据形式。但是清洗后的数据也并非完全是所需要的数据。在进行数据清洗工作后，还需要从所得数据中抽取出本次研究中真正需要的数据。

2）数据抽取

数据抽取表示根据需求，在原始数据之中抽取出所需数据。

本次研究中，所选取的词语聚类方法只需要词语名称即可。数据清洗过程只是实现了数据的规范化工作，清洗后的数据中，商品ID、商品价格等都是不需要的数据。这些数据对接下来的数据分析工作并没有帮助，并且由于数据量较大，占用大部分的存储空间，降低了数据的处理速率。因此需要在清洗后的数据中进行数据抽取工作，抽取出商品名称字段，保存为json文件，进行下一步数据处理。

3）数据去重

在商品交易中，经常出现一种商品多次交易的情况。在本次交易数据中处理结果可能出现很多商品名称重复数据。这些重复数据或许会导致分析结果出现偏差，不够准确。因此为保证词语聚类的效果，应当删除重复数据。基于此目的，对得到的物品词语进行了数据去重设计与实现，将名称相同的数据进行匹配并且删除重复值。去重后的数据保存为文本文档。进行下一步数据处理。

在数据预处理中，由于所需要的数据对于不存在特殊符号，不存在乱码等的更高要求，可能数据预处理中的各个步骤需要反复进行，直到得出可以进入下一步操作的数据为止。因此数据预处理的过程需要在实际的操作中，对每一步得到的数据进行多次的审核与处理。

### 5.2.2 分词处理

预处理后的数据为一个文本文档数据，此文档中为原始数据中的物品名称数据，分词处理便是对所得数据进行分词，将物品名称切分为最小粒度。

经过对多种中文分词的工具调研分析，对分词的效率，分词的效果进行考虑，所选择的分词工具为IK分词工具。IK分词工具如今的开发使用已经较为广泛。并且经过实验得出的分词效果满足需求。IK分词实际应用中，可以解决全角半角问题，对大小写的问题进行统一，比如“5”和“五”，统一不同的表达方式。IK分词对于乱码的解决问题效果较好，并且支持标点符号的转换与去除。分词速度较快，支持用户自定义词典与用户自定义停用词。与此同时，IK分词是一个开源的分词工具，如今网络上的资源较为丰富，而且可以通过修改IK分词的代码，对其功能进行扩展开发，使其充分应用于本次研究。IK分词的这些特点，对源数据的限制较小，分词效果更满意[47]。

数据进行分词处理后，所得的结果是切分到最小粒度的词语集。将此词语集输出保存到文本文档。进入下一步的工作。

### 5.2.3 词语聚类

上述工作结束后，已经得到了进行词语聚类实现所需要的数据，之后的工作便是根据所得的语料，实现词语聚类的功能。所选用的词语聚类的方法是基于统计的聚类方法，此方法考虑到相似的词语处于的语言环境往往也是相似的，因此词语的相似度判断方法是词语所在的上下文的相似度。此方法往往是选取多个特征词语，根据特征词在词语上下文出现的频率，计算出每个词语的上下文同现向量。此方法使每个词语根据上下文信息得出了一个单独的向量，之后，通过词向量便可以计算两个词语之间的相似度，从而达到词语聚类的目的[48]。

本文使用基于统计的词语聚类方法，实现机器对词语集进行自动聚类的功能。

经过研究，首先将词语转换为词向量以数值表示，之后使用Kmeans聚类算法，根据词语相似度，将相似度高的物品词语聚类。在此研究中，词语转换为词向量的工具选择为Word2vec。

1）Word2vec介绍

Word2vec是2013年有google开发的，一款实现以向量的方式表示词语的开源工具。经过word2vec转化出的词向量在自然语言处理中可以应用于词性标注，聚类等。Word2vec使用的由Hinton在 1986 年提出的Distributed representation，它是词向量表示方式。Word2vec是使用深度学习中的神经网络进行训练的。它的基本思想为，通过计算转化后的词向量的距离来判断相似度[49]。Word2vec通过模型的训练，将词语训练为k维的词向量，之后根据如欧氏距离，cosin相似度等计算向量之间的距离的方法进行距离计算。两个词语向量间的距离越近，表示两个词相似度越高。

Word2vec的训练模型分为两种，cbow，即连续词袋模型和skim-gram模型。Cbow模型是根据词语的上下文信息来预测词语生成的概率，skim-gram模型则与cbow模型相反，根据词语进行其上下文词语出现概率的预测[50]。两种模型都有一个共同的特点是，对于相同的语料库，输出的概率值的和为1。

Cbow模型与skim-gram模型如图5.1和图5.2所示：



图5.1 cbow模型



图5.2 skim-gram模型

接下来对cbow与skim-gram模型的实现原理与实际应用方法进行详细的介绍。

（1）Cbow模型

Cbow模型为一个三层的神经网络模型。输入是已经获得的上下文信息，输出的是经过上下文信息所预测出的词语。

Cbow模型结合了哈夫曼树，哈夫曼树中的非叶结点每个都直接与隐藏层相连接。非叶结点中是一个二分类器，通常使用的是softmax感知机等。哈夫曼树的叶节点对应着语料库中的所有词语。Cbow模型训练过程中，当输入层输入了已知的上下文向量后，在模型中的隐藏层会累加所有输入的词向量，此时得到了一个中间向量。之后将次中间向量输入到哈夫曼树的根节点之中，根节点根据规则，将此中间向量分入其左子树或者由子树。之后于每个非叶结点中的二分类器进行分类处理，直至此向量到达某个叶子节点。到达的叶子节点便是预测的输出词语。

实际应用时，需要根据获得的语料库建立相应词汇表，并且在词汇表中的所有的单词都有一个对应的词向量。选择语料库中的一段文本进行模型的训练。选取特定的单词，已知此词语的上下文词向量，将其输入到cbow模型中，在cbow模型中，上下文向量从隐藏层的计算到哈夫曼树的分类，最后输出到一个叶子节点中。在此过程中，实验前已经确定了此词语，并且经过实验，确定了哈夫曼树各个分类器做出的预测与从根节点的输入到叶子节点的输出的正确的路径。在之后，为了实现各个词向量输出的实际路径与其正确的路径相同，需要使用梯度下降法进行词向量的调整。经过调整后，各个词语便得到了相对应的词向量[51]。

（2）Skip-gram模型

Skip-gram模型是一个三层的神经网络模型。输入是已知的单词的信息，输出为通过此单词预测出的上下文信息。

Skip-gram模型与cbow模型一样，推测上下文信息同样结合了哈夫曼树。每一个单词输入到哈夫曼树的根节点，之后经过非叶子节点的判断到达叶子节点，叶子节点存储的词语便是输入词语的上下文词语。

在实际的应用中，与cbow模型相同，需要对词汇表中的每一个单词进行多次迭代，并且通过不断地调整输入的词语向量，使得在训练中的路径与正确的路径相符合。从而得出较为精确的结果[52]。

在本研究中，由于需要计算各个词语对于部分词语的相似度，并且选取的是部分的词汇，根据选取的词汇对词语集进行词向量模型建立。因此使用的模型是skip-gram模型。当实现了词语对应词向量的转化之后，便继续结合聚类算法，通过计算每个词语之间的相似度，实现词语的聚类。

2）Kmeans算法介绍

Kmeans算法的迭代过程如下：

（1）从处理后的数据集中随机的抽取k个数据作为聚类中心；

（2）对每个数据计算与各个聚类中心的距离，距离最小的聚类中心为同一类，计算方法可以选用欧式距离：

（3）聚类结束后根据聚类结果，重新计算聚类中心；计算聚类中心对应公式为：

其中的取值通常为1或者2。

（4）迭代第（2），（3）步，直到重新计算的聚类中心不再变化或者小于某一阈值为止，聚类结束[53]。

## 5.3 本章小结

本章中详细介绍了所需要的数据与技术。本章中对数据的采集进行了实现，展示了所需的数据，对数据预处理做了详细设计，数据预处理主要包括数据清洗与数据去重部分。对于分词选择了IK分词工具，对IK分词的原理进行了较详细的介绍。词语聚类部分，详细介绍了实现此功能所需的词向量转化与聚类算法，本文中词向量转化选择的是Word2vec工具，聚类算法选择的为经典的Kmeans聚类算法。

# 物品词语聚类结果分析

上面分析研究了进行词语聚类所需要实现的模块，之后针对项目的特征和需求对物品词语聚类进行了设计与实现。

## 系统部署

系统在应用层上所采用的开发系统为win10系统，使用IntelliJ IDEA作为编辑器，与应用层配置环境信息，连接Spark与Hadoop集成系统，使用hadoop中的HDFS进行文件的存取工作，使用Spark进行数据的处理工作。在此环境下进行程序的开发运行。

所使用的系统与软件相关信息如表6.1所示。

表6.1 软件相关信息

|  |  |  |  |
| --- | --- | --- | --- |
| **项目** | **版本信息** | **项目** | **版本信息** |
| 操作系统 | Cent OS 6.3 | JDK | 1.8.0 |
| Hadoop | 2.7.1 | Python | 2.7.9 |
| Spark | 1.5.1 |  |  |

本实验中Spark与Hadoop集群系统选取了四台独立物理机器作为节点。四台机器中，三台作为从节点，一台作为主节点。Spark集群的配置方式选取的是Spark Standalone模式。

以四台主机所配置的Spark与Hadoop集成环境表6.2所示：

表6.2 硬件相关信息

|  |  |  |
| --- | --- | --- |
| **主机名** | **IP** | **角色** |
| master | 192.168.31.192 | NameNode、JobTracker |
| slave1 | 192.168.31.236 | DataNode、TaskTracker |
| slave2 | 192.168.31.188 | DataNode、TaskTracker |
| slave3 | 192.168.31.207 | DataNode、TaskTracker |

## 6.2 测试与分析

实验中，使用了一台高性能的一体机作为服务器，一体机上包括了12个节点，其中每一个节点都可以是一部单独的高性能的服务器，此机器中的12个节点在各个方面都是相同的。在此服务器上可以实现根据不同数据量与性能需求添加或减少节点数，实现大数据分析。所有的底层架构都是以此机器作为硬件基础，于机器上配置环境，进行数据的处理分析。

在本次实验中，已经充分的提供了所需的硬件设施，为之后的数据处理工作奠定良好的基础。

### 6.2.1 测试数据集

实验所用的数据是利用网络爬虫从天猫爬下的实时数据，经过人为预处理后进行了词语聚类分析。实验的数据被保存在HDFS上进行输入与输出。

由于所得数据可能存在不完整的，可能缺少某些属性值或离群值等。这是不可避免的，而且目前的数据难以直接进行接下来的流程。为顺利的完成此工程，并且为了保证词语聚类的实现效果，数据采集后对原始数据进行数据预处理。

首先根据数据特点，对数据进行了清洗，将有缺失值的数据直接删除。之后进行数据抽取，由于所获数据中许多内容是不需要的，因此需要从中抽取所需要的物品名称字段，于数据中为标题字段。处理后所形成的部分结果如图6.1所示：

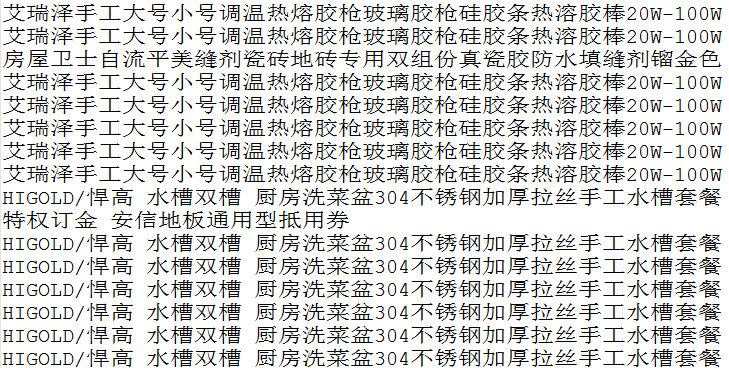


图6.1 数据抽取结果

很明显可以看出所处理后的数据具有重复数据，因此需要继续进行数据清洗，进行数据去重操作。操作后的部分数据如图6.2所示：

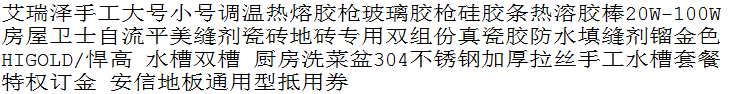


图6.2 数据去重结果

重复字段已经去除，但是此数据中，如：“HIGOLD/悍高 水槽双槽 厨房洗菜盆304不锈钢加厚拉丝手工水槽套餐”，这段数据中，“HIGOLD/悍高”，“304”均为无效字段，针对此数据，需要将其清洗为：“厨房洗菜盆不锈钢加厚拉丝手工水槽”，方为可用数据。编写脚本对此数据进行分词并且去除不需要的部分，处理后的部分结果如图6.3所示：

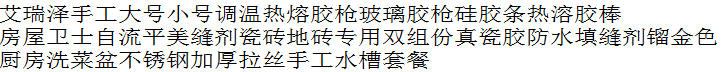


图6.3 数据抽取结果

再次抽取了所得字段中的有效字段，可能会在数据中再次出现重复字段，因此需要对再次抽取后的字段进行数据去重操作。

经过上述操作后，数据预处理结束，所得到的数据将继续进行中文分词操作。

### 6.2.2 分词测试

如上小结所示，预处理后的数据为一个较长的词语，而分词过程需要将所获得的每一个词语分解为一个一个较小的词语。挑选的IK分词工具可以实现此目的。分词结果如图6.4所示：

C:\Users\ADMINI~1\AppData\Local\Temp\1492655865(1).png

图6.4 分词结果

首先选取了处理后的一小段语料进行测试，发现分词效果不是很满意，切分词语粒度过小，如“厨房洗菜盆”这类固定搭配的词语进行分词时，IK分词将其分词为：“厨房”，“洗菜”，“盆”，针对这种情况，需要保证这些固定搭配的词语为一个可识别的完整的词语。由于IK分词中具有扩展词典的功能，因此在实现过程中，于扩展词典中加入了固定的词语。进行分词时，辨认出所挑选的词语不需要进行进一步分词，以此保证分词质量。

针对所测试的数据，于扩展词典中加入的数据均为物品类别名称，如：“女士衬衫”，“男士洗面奶”等。当使用IK分词工具时，可以保证加入的扩展词语不被切分为更小粒度的词语。此时分词时便会保证固定词语搭配不被切分开。

加入扩展词语后，分词结果更加符合接下来进行词语聚类的要求。

### 6.2.3 词语聚类测试

使用分词后的数据进行word2vec的编程实现与词语聚类的编程实现。

Word2vec的编程实现结果，如图6.5所示：

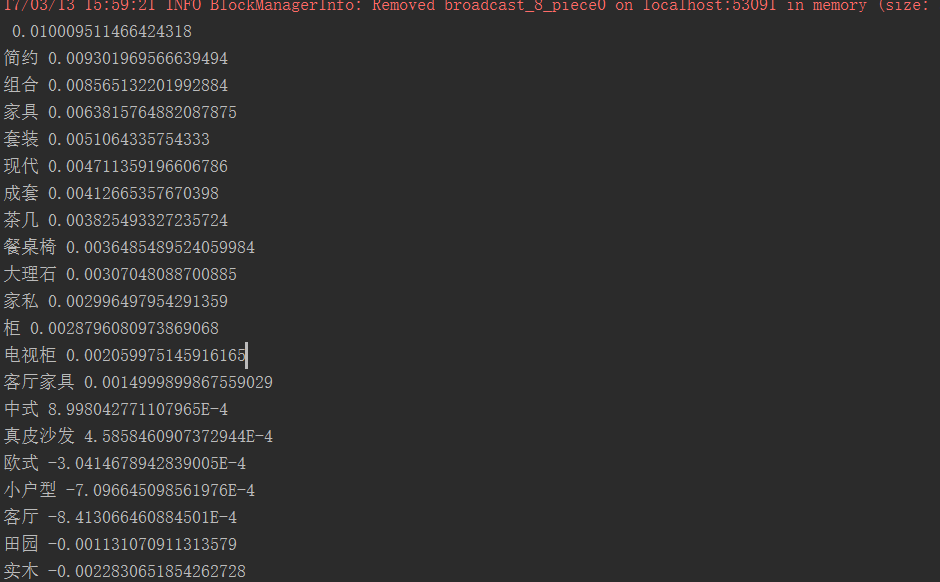


图6.5 word2vec编程实现结果

测试的词语为：“沙发”，输出结果左侧表示相似的词语，右侧表示输出的词语与输入的词语对应的相似度。输出结果按照相似度大小由大到小进行排序。word2vec计算的是余弦值，距离范围为0-1之间，值越大代表这两个词关联度越高，所以越排在上面的词与输入的词越紧密。

词向量的转换之后再进行词语聚类，词语聚类结果如图6.6所示：

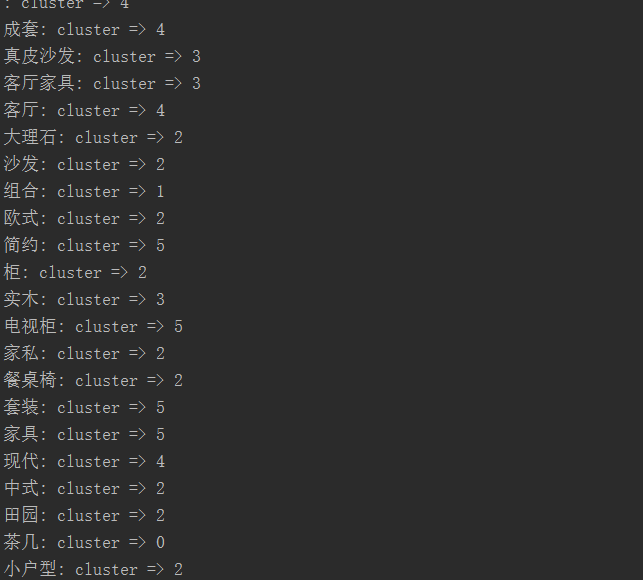


图6.6 词语聚类结果

其中对应数字相同表示聚为同一类，并且所有类别经过处理后相同的类处于文件相近的地方，便于查看。

### 6.2.4 结果分析

为了本次词语聚类的方法进行检测，决定使用不同的语料库，进行对比试验。从所获得的数据中，选取部分数据作为测试集。不同的测试集中，测试集之间，词语的数据量存在差异。各个训练集分别选取了1000，2000和3000个词语作为词语的训练集。为了使实验的对比性更强，首先在所有的数据集中选取了3000个词语作为第一个训练集，之后在3000个词语的训练集中选取了2000个词语作为第二个训练集，最后在2000个词语的训练集中选取了1000个词语作为训练集。选择的训练集都包含了相同的数据，并且在选择数据集时，选择的方式是于原始语料库中，选取的连续的词语集。以此保证各个词语的上下文情况是相似的。

将不同数据量的训练集，使用相同的词语聚类模型进行训练，根据输出，计算分析本词语聚类模型对于词语聚类实现的不同效果。由于模型中使用的是无监督学习的自动聚类方法。因此各个类别并没有相对应的类别名称。在此情况下，考虑到各个训练集中的数据量较小，因此决定使用人工判断的方法，计算聚类结果的精确度。

在此模型的角力如：于训练结果中，发现“高脚桌”，“餐桌椅”，“衣柜”，“笔记本电脑”聚类结果是聚在相同类别之中，而“茶几”，“沙发”却与“女士衬衫”，“高腰裤”，“连衣裙”等聚在相同的类别。在此情况下，通过人工进行查找判断出聚类中出现差错的词语，以此计算聚类精确度。

在考虑聚类的效果的同时，也对运行效率进行了分析检测。计算其运行效率，检验此词语聚类的方法，是否可以满足较大的数据量。本次性能检测与精确度检测都是同时进行的。对于各个不同数据量大小的训练集，每一个都进行了三次实验，之后通过计算三次实验的平均值，进而得出聚类效果的精确度和运行时间。

由此种方法得出的结果，如表6.3所示：

表6.3 词语聚类效果检测

|  |  |  |
| --- | --- | --- |
| **训练集词语数量** | **精确度** | **运行时间（秒）** |
| 1000 | 0.62007 | 2.007 |
| 2000 | 0.67567 | 4.444 |
| 3000 | 0.72662 | 7.956 |

这次实验设计不同的数据量，使用相同的硬件和软件环境进行训练集的训练。并且在此过程，通过各个数据集中包含了相同的数据，由此尽最大可能的排除了由于数据中内容不一致导致精确度计算不准确的因素。通过设计多次实验，选取平均值，排除了实验可能会出现的的偶然性情况。由这些设计保证了实验的可靠性。

通过表 得出，训练集中的词语数据越多，词语聚类的精确度越高，同时由所计算出的精确度可知，训练集挑选的数据量较小，导致训练之后的结果，精确度不够高。在运行时间的对比中得出，运行时间是成倍增长的。通过运行时间与词语数量计算出的运行效率差别不大，但是可以发现效率仍是逐渐降低的。由此可以得出，随着数据量的增大，训练效率会随之降低。

通过上述不同大小语料库的实验中，可以看出，本文中设计的词语聚类模型，是可以应用于词语聚类的。并且运行效果与运行效率上，都已满足了需求。但是，次词语聚类模型中，通过调整参数，或者调整计算方法，在保证效率的前提下，词语聚类的精确度还可以继续提高。因此，本次研究中，对于词语聚类的效果还可以进行多方改进，使其功能更加强大。

## 6.3 本章小结

本章节介绍了词语聚类的具体实现流程与细节。数据的预处理部分于第四章已经有详细的介绍，本章介绍编程实现了Word2vec与Kmeans算法结合，所实现的词语聚类。

# 总结与展望

## 7.1 论文总结

如今大数据的发展越来越迅速，各企业对大数据的应用研究也越发深入，在大数据的背景下，自然语言处理的发展也已经十分成熟，本文中主要介绍了大数据的基本理论，与自然语言处理方向的词语聚类的分析设计与实现。本文针对此方向具体工作如下：

1）Spark与Hadoop集群环境配置。在本文中的实现环境为以Windows微客户端，Spark为服务器，HDFS进行存储的环境。

2）数据的爬取与数据的处理。数据来源是使用网络爬虫爬取了交易信息数据，数据的处理包括了数据的预处理与分词部分，数据预处理部分主要进行了数据提取与数据去重。预处理中首先提取所需数据，本文所提取的是数据中的商品名称部分，之后进行数据去重，去除重复的商品名称。进行预处理后的数据继续使用IK分词工具进行分词，为之后词语聚类做准备。

3）词语聚类实现。词语聚类的实现主要使用了Word2vec分词工具与Kmeans聚类算法，首先将词语转化为了词向量，之后使用Kmeans聚类算法，实现了将相似度高的词语进行聚类，并且保存聚类结果。

4）结果分析。通过使用不同数据量的训练集进行词语聚类的训练与结果输出，分析数据量大小对于聚类效果的影响与对聚类性能的影响。从此次对比分析结果得出。在相同的环境下，词语聚类精确度随着数据量的增大而增大，运行效率随着数据量的增大而降低。

## 7.2 展望与后续工作

本文对词语聚类的设计主要针对与物品词语，预处理部分的处理是针对所爬取的数据特点进行预处理，得到所需的数据。物品的词语聚类可以应用于物流信息处理，分析各个省市，各个大客户的主要发件物品种类，便于进行异常点检测与商业决策。大数据技术的成熟使得即使处理TB级的数据都可以十分高效，因此在实际应用中，超大型数据量的处理可以通过大数据技术提高性能与准确率。本次实现中于Spark与Hadoop集群的使用中也亲自感受到了此环境的优势，集群环境对于处理数据的效率远高于单机条件下运行程序的效率，当于单机下出现由于数据量过大，数据处理无法全部处理结束，输出结果混乱的情况下，选择集群系统进行本次聚类实现显然是实验成功的原因之一。

在感受了大数据技术的实用效果下，接下来的工作中，应该对大数据技术进行更加深入的研究，数据是大数据应用实现的先决条件，因此可以考虑使用不同类型的多种数据进行对比实验，通过实验结果，观察哪种类型数据更适合本次实验。本次实验中明显发现了数据量的多少对于运行结果的影响，发现数据量越大，准确率越高，因此后续工作中应该考虑继续增大数据量以保证聚类效果。

除了数据的影响，后续工作中应该考虑对可视化部分进行设计与实现，以更直观的方式展示所处理的结果。

聚类的模型构建也应当深入考虑，本文实现聚类是以Word2vec工具与Kmeans聚类算法相结合模型。可以考虑对Kmeans算法进行深入研究分析，加以改进，使其更适用于词向量转化后的聚类。

本文中对聚类效果的优劣判断，仅仅依靠着人工判断，使得测试过程过于消耗人力，对于这一部分可以加以改进研究，设计更加系统科学的聚类效果判断模型。在性能上的判断，由于进行训练时，需要通过windows客户端连接Spark服务器，因此在运行时间中包含了连接服务器产生的时延。针对这一点，本文没有设计相应的方法进行精确的计算，或许令测试在不影响结论的情况下，存在着些许误差。因此，如果希望测试结果更加具有说服力，应该充分考虑各种误差，并且进行相应的处理。

# 参 考 文 献

[1] 奚雪峰, 周国栋. 面向自然语言处理的深度学习研究[J]. 自动化学报, 2016(10):1445-1465.

[2] 袁里驰. 几种基于统计的词聚类方法比较[J]. 中南大学学报(自然科学版), 2016(09):3079-3084.

[3] 许宁宁. “互联网+”时代电子商务的发展[J]. 科技视界, 2016(05):129.

[4] 蒋骁. 基于知识管理的企业电子商务应用研究[D]. 武汉理工大学, 2005.

[5] 王强, 李俊杰, 陈小军. 大数据分析平台建设与应用综述[J]. 集成技术, 2016(02):2-18.

[6] Junfei Qiu, Qihui Wu, Guoru Ding, Yuhua Xu, Shuo Feng. A Survey of Machine Learning for Big Data processing. EURASIP Journal on Advances in Signal Processing. 2016, Vol.2016 (1), pp.1-16

[7] Bonnie L. Westra, Martha Sylvia, Elizabeth F. Weinfurter, Lisiane Pruinelli, Jung In Park, Dianna Dodd, Gail M. Keenan, Patricia Senk, Rachel L. Richesson, Vicki Baukner, Christopher Cruz, Grace Gao, Luann Whittenburg, Connie W. Delaney. Big Data Science: A Literature Review of Nursing Research Exemplars[J]. Nursing Outlook, 2016.

[8] 刘金辉王军. 大数据的国内外研究现状及发展动态分析[J]. 电子技术与软件工程, 2015(23):200.

[9] 孙越恒, 曹桂宏, 侯越先. 对称和非对称词语聚类模型的比较研究[J]. 计算机工程, 2009(10):14-17.

[10] 刘华. 基于语料库的领域词语聚类C#实现[J]. 计算机工程与应用, 2005(36):167-169.

[11] 郭怀恩, 朱礼军, 徐硕. 词聚类技术研究综述[J]. 数字图书馆论坛, 2010(5):15-19.

[12] 林作新. 对互联网+与电商的反思[J]. 家具与室内装饰, 2016(02):14-15.

[13] 孙科. 基于Spark的机器学习应用框架研究与实现[D]. 上海交通大学, 2015.

[14] Orçun Yildiz, hadi Ibrahim, abriel Antoniu. Enabling fast failure recovery in shared Hadoop clusters: Towards failure-aware scheduling[J]. Future Generation Computer Systems, 2016.

[15] 郝树魁. Hadoop HDFS和MapReduce架构浅析[J]. 邮电设计技术, 2012(07):37-42.

[16] 顾荣, 严金双, 杨晓亮, 等. Hadoop MapReduce短作业执行性能优化[J]. 计算机研究与发展, 2014(06):1270-1280.

[17] 孙惠生. 浅谈列式数据库[J]. 企业技术开发, 2010(21):43-47.

[18] Sachin Bende, Rajashree Shedge. Dealing with Small Files Problem in Hadoop Distributed File System[J]. Procedia Computer Science. 2016.

[19] 刘刚. Hadoop应用开发技术详解. 机械工业出版社. 2014.

[20] 陈虎. 基于HDFS的云存储平台的优化与实现[D]. 华南理工大学, 2012.

[21] 连玉明. 中国大数据. 当代中国出版社. 2014, 07.

[22] 师洪波, 吴振新. 基于HDFS的分布式长期保存系统实现研究[J]. 图书馆学研究, 2012(01):29-35.

[23] 顾荣, 严金双, 杨晓亮, 等. Hadoop MapReduce短作业执行性能优化[J]. 计算机研究与发展, 2014(06):1270-1280.

[24] 李伟卫, 赵航, 张阳, 等. 基于MapReduce的海量数据挖掘技术研究[J]. 计算机工程与应用, 2013(20):112-117.

[25] 王韬. 基于Spark的聚类集成系统研究与设计[D]. 西南交通大学, 2015.

[26] 刘春雷. 基于代价模型的Spark SQL查询优化研究[D]. 电子科技大学, 2016.

[27] 王春凯, 孟小峰. 分布式数据流关系查询技术研究[J]. 计算机学报, 2016(01):80-96.

[28] 胡于响. 基于Spark的推荐系统的设计与实现[D]. 浙江大学, 2015.

[29] 张长水，杨强. 机器学习及其应用-2013. 清华大学出版社. 2013.

[30] 王秀峰刘明吉. 数据挖掘中的数据预处理[J]. 计算机科学, 2000,27(4):54-57.

[31] 和春慧. 数据仓库中数据预处理的研究与算法实现[D]. 吉林大学, 2004.

[32] 刘云霞. 数据预处理. 厦门大学出版社. 2011.

[33] 周晓分, 黄国彬, 白雅楠. 科学计量可视化软件的对比与数据预处理研究[J]. 图书情报工作, 2013(23):64-72.

[34] 李季, 孙凯, 白文. 复杂设备维修保障数据预处理技术研究[J]. 计算机与数字工程, 2016(04):711-714.

[35] 关大伟. 数据挖掘中的数据预处理[D]. 吉林大学, 2006.

[36] 梁喜涛, 顾磊. 中文分词与词性标注研究[J]. 计算机技术与发展, 2015(02):175-180.

[37] 周祺. 基于统计与词典相结合的中文分词的研究与实现[D]. 哈尔滨工业大学, 2015.

[38] 张晴. 概念整合理论视角下的许渊冲唐诗英译——基于ICTCLAS2013的案例分析[J]. 读与写(教育教学刊), 2015(03):24-25.

[39] 黄翼彪. 实现Lucene接口的中文分词器的比较研究[J]. 科技信息, 2012(12):246-247.

[40] 柴洁. 基于IKAnalyzer和Lucene的地理编码中文搜索引擎的研究与实现[J]. 城市勘测, 2014(06):45-50.

[41] 李泽中. 最大熵结合词语聚类的中文词性标注研究[D]. 大连理工大学, 2010.

[42] 康铁钢, 戴汝为. 一种基于大规模标注语料库的词语聚类方法[J]. 系统仿真学报, 2003(10):1439-1442.

[43] 徐硕郭怀恩朱礼军. 词聚类技术研究综述[J]. 数字图书馆论坛, 2010(5):15-19.

[44]李学龙, 龚海刚. 大数据系统综述[J]. 中国科学:信息科学, 2015(01):1-44.

[45] 陈虹君. 基于Hadoop平台的Spark框架研究[J]. 电脑知识与技术, 2014(35):8407-8408.

[46] 叶惠仙, 贾如春. 大数据架构关系分析及应用[J]. 计算机时代, 2016(12):42-45.

[47] IK Analyzer 2012发布中文分词库[J]. 硅谷, 2012(06):73.

[48] 李锦. 仿射传播算法在中文词语聚类中的应用研究[D]. 昆明理工大学, 2011.

[49] 周练. Word2vec的工作原理及应用探究[J]. 科技情报开发与经济, 2015(02):145-148.

[50] 熊富林, 邓怡豪, 唐晓晟. Word2vec的核心架构及其应用[J]. 南京师范大学学报(工程技术版), 2015(01):43-48.

[51] 朱雪梅. 基于Word2Vec主题提取的微博推荐[D]. 北京理工大学, 2014.

[52] 李跃鹏, 金翠, 及俊川. 基于word2vec的关键词提取算法[J]. 科研信息化技术与应用, 2015(04):54-59.

[53] 蔡洪山. 基于Spark的Kmeans并行算法研究[J]. 电脑知识与技术, 2016(04):74-75.

攻读硕士学位期间取得的学术成果

发表论文情况：

[1] Yufen Cheng, Wenqian Shang, Ligu Zhu, Di Zhang. Design and Implementation of ATM Alarm Data Analysis System. Computer and Information Science (ICIS), 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS). (EI检索)

[2] Yufen Cheng, Wenqian Shang, Ligu Zhu, Di Zhang, Dongyu Feng. Items Analysis of Postal Supervision. Computer and Information Science (ICIS), 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS). (EI检索)