基于大数据平台的中文文本分析系统的设计与实现

摘要：进入21世纪，互联网技术的发展突飞猛进，互联网已经与各行各业息息相关，形成了互联网+的大格局，同时，连接在互联网上的各种设备在源源不断产生着数据，造成数据的大爆炸式产生，为大数据时代的到来奠定了基础，这其中包括大量的文本信息，这些文本信息以日志、评论、文章等形式呈现在互联网上，由于互联网与人们的生活越来越紧密，并且网络舆情对社会热点的影响也越来越大，如何分析网络观点、预测网络情绪并正确引导网络舆情成为当今社会乃至全世界亟需解决的问题。其中对文本分析技术的研究是解决这一问题的关键点，由于中文的特殊性，对中文文本的分析更加困难，中文文本分析技术主要包含文本采集、文本清洗、文本标注、文本训练、文本分类这5大类技术，这5大类技术中包含了许多关键的算法和系统，通过对这些算法的优化和对系统的整合可以满足对文本分析任务的需求，其中最具代表性的任务就是对文本倾向性的分析，文本倾向性分析是指通过采集、标注、训练、分类等手段分析文本的立场、观点、情感、情绪等主观信息，从而对文本倾向性做出褒贬划分。这一技术涉及了文本分析技术的方方面面，并可以应用到评论系统、舆情系统、问答系统、推荐系统等多个领域。

同时，随着技术的不断成熟，开源社区的繁荣，研究人员对海量数据的处理不再困难，各种大数据平台的出现为处理海量数据提供了方便，尤其是2009年推出的大数据计算框架Apache Spark，并在2014年成为Apache基金会的顶级项目，其不断的完善为大数据处理带来了新的曙光。

本论文基于Spark大数据平台，并以互联网上的评论信息作为研究对象，构建一套针对中文文本倾向分析的大数据分析系统，通过利用文本分析相关技术，对评论信息进行采集、过滤、标注、训练、分类等操作，最终提供一套分析评论文本正负倾向性的解决方案，并通过利用LDA算法和doc2vec算法相融合的方式提高文本倾向分析的准确率和召回率。本文主要包括以下几个方面：

首先，本论文分析了课题的研究背景和国内外研究现状，并针对中文文本分析场景提出了可行性研究，阐述了关键技术和主要思路。其次，针对中文文本分析领域详细系统的介绍了相关算法和技术，同时，还介绍了大数据平台的特点和架构，并分析了基于大数据平台开发系统的优势，以及如何在大数据平台上构建文本分析系统。然后，在次基础之上设计了中文文本倾向分析系统的总体架构，并提出了基于情感标注和标签融合的文本标注算法和基于doc2vec和LDA的文本表示算法，最后根据系统设计完成各个模块的代码编写和系统整合，并提出总结和分析。

关键词：Spark 文本倾向分析 doc2vec LDA神经网络 机器学习

1.绪论

1.1研究背景及意义

进入21世纪以来，互联网已经成为人们生活工作中不可缺少的工具，人类进入了信息化时代，在过去的10年里计算机为人类提供了巨大的生产力，提高了生产效率，随着计算机科学的发展，尤其是互联网的发展，每天产生的数据量以指数级的速度增长，尤其是近几年社交网络、电子商务平台、自媒体的崛起，使得人们对数据的重视程度越来越高，数据已经成为新型的生产资料被人们发掘、利用，并且在实际的生成环境中带来效益，人类已经不可避免的从信息化时代步入了大数据时代[4]，但是大数据时代的到来既给我们带来机遇也给我们带来了挑战，如何对海量数据进行存储、过滤、处理和分析一直是工业界和学术界有待解决的问题，尤其在这些大数据中80%的数据是非结构化的文本数据，其中包括以社交媒体为代表的微博、博客、日志，以虚拟交易平台为代表的商品描述、用户评论，以新闻媒体为代表的时事新闻、讨论等海量文本信息，通过对这些信息的处理和分析可以实现情感分析、舆情分析、商品推荐、金融预测、用户行为描述、广告精准投放等具有实用价值的应用。本文针对互联网上大量需要处理的文本数据，设计并实现一种基于大数据平台的中文文本分析系统，旨在为需要处理海量文本，尤其是中文文本的用户提供一种可行的解决方案。

大数据对于全球还是比较新的课题，各个国家都有自己的大数据研究项目，尤其是以Google为代表的新型互联网公司，对大数据技术的探索和发展产生了推动的作用，Google作为搜索引擎巨头公司，在2003年开始陆续发表了三篇有关大数据的论文，分别论述了GFS[1]、MapReduce[2]、Bigtable[3]三大技术，这三篇论文揭开了大数据技术的神秘面纱，成为工业界对大数据理论的实现标杆。

同时，受到Google的启发，Apache基金会于2005年秋天引入了Hadoop项目，Hadoop作为开源项目和其稳定的架构设计迅速成为了大数据平台的佼佼者，可以说Hadoop已经成为了大数据的代名词，其最大的核心就是实现了一个高容错性的分布式文件系统HDFS[5]和MapReduce计算模型，其中前者为海量数据存储带来解决方案，后者为海量数据计算提供了支持。但是随着Hadoop在工业界的广泛使用，其为企业带来便利的同时也暴露出自身的一些局限性，例如抽象层次低、上手难度大；只提供map和reduce两种操作，表达力不足；复杂的计算需要多个mapreduce过程，job之间的依赖关系由开发者自己管理；时延高，只适合计算密集型任务；对于迭代式数据处理性能差等。

正在大家努力寻求可行的解决方案的时候，Spark[6]作为新一代的大数据框架正悄悄地走进人们的视线，Spark是UC Berkeley AMP lab开源的类Hadoop MapReduce的通用并行框架，其不但补充了Hadoop的不足，而且还集成了很多优秀的组件，例如其增加了数据库和数据框架库、流处理库、机器学习库和图形库。通过引入这些库，用户可以执行更加复杂的任务，并且spark将中间结果保存到内存中而不是磁盘上，这使得spark的效率也非常高，可以想象spark未来发展的前景是非常乐观的。但是，由于Spark技术出现的比较早，人们对spark的应用仍然处在探索阶段，如何利用spark解决现有的大数据问题仍然值得我们去研究和探索。

文本分析相对来说发展的比较成熟，现有的文本分析算法也非常多，效果也经历了工业界的考验，文本分析是自然语言处理发展的必经之路，首先对于自然语言来说，文字是表达语言的重要形式之一，计算机如何正确的理解文本内容，就相当于其可以正确的理解自然语言，在文本分析的初期，计算机理解文本的方式是通过词典映射，间接的“理解”其意思，其实是通过最大字符串匹配获得结果，这种方式其实是没有解决计算机理解自然语言的根本问题，随着人们不断的研究，基于概率统计的算法被应用到文本分析领域[7][8]，并且产生了很好的效果，文本分析进入了新的阶段，近年来机器学习、深度发掘[9][10]等新的领域发展迅速，人们又找到了文本分析新的突破口，现阶段文本分析的主要思路是结合词典和概率统计算法对语料进行初级处理，之后利用机器学习分类算法对其进行训练，最终得到比较理想的结果。

但是基于机器学习的文本分类算法对训练文本的依赖性很高，而且需要对文本的特征进行提取，不同的特征提取方式会导致结果的精确度不同，常用的的特征提取方式是采用TF-IDF作为文本的特征值进行训练，效果虽然可以满足要求，但是TF-IDF算法没有考虑文本之间上下文语境，准确度仍然有待提高，本文以文本倾向分析作为切入点采用LDA作为文本特征提取的优化算法，并结合神经网络中的doc2vec算法对文本进行训练，完成文本分析任务。采用神经网络的算法考虑了文本上下文之间的关系，可以提高文本倾向分析的精度。

同时，处理海量的文本信息，需要进行大量的计算任务，如何将这些任务并行化是文本需要解决的另一个问题，本文通过利用Spark大数据平台，对文本进行分析和处理，其中包括对文本的切分，存储，训练和分类等文本分析涉及的过程。利用大数据平台的计算能力，采用LDA作为文本特征提取算法，并结合最新的doc2vec算法对文件进行倾向分析，设计并实现针对中文的文本倾向分析系统，并大大的提高了准确率和效率。

1.2国内外研究现状

文本分析是自然语言处理过程中最重要的环节，其中文本分析任务最首要的难题是如何表示文本，让计算机更好的理解文本。经过研究者的不断努力，文本表示算法已经有了突飞猛击的发展，从最初的基于词典的表示，到基于上下文的表示，最后到基于词向量的表示，文本表示算法已经满足现阶段的需求。其次需要解决的问题时如何获取文本特征，由于文本表示的特征向量往往具有高维性和稀疏性，这样的表示方法增加了处理文本的难度和成本，如何从高维特征中提取出最具代表性的特征就是文本分析需要解决的第二个难题。最后是对特征表示后的文本进行分类，完成任务需求，分类任务相对于比较容易，因为基于机器学习的分类算法已经非常成熟，但是如何使用特定的分类算法需要根据具体的需求来分析，这样才能达到理想的效果，并且进行算法模型的训练时间往往比较长，如何提高训练效率也是需要考虑的问题。

针对以上提到的三个问题，本文以文本倾向分析为切入点来阐述文本分析领域研究现状。主要通过两个方面进行综述：一、文本倾向分析研究综述，二、基于大数据平台的文本分类。

1.2.1 文本倾向分析研究综述

文本倾向分析是指通过对文本的处理获得该文本的正负情感倾向性，文本倾向分析是文本分析的重要应用领域，通过对文本倾向分析的研究可以应用到整个文本分析过程中涉及的关键技术和重要算法，文本倾向分析概念的形成经历了一个漫长的探索过程，最初是以“意见挖掘(opinion mining)”的概念在2003年被Dave发表在 www大会上的论文提出，其认为理想的意见挖掘工具是可以处理一个给定主题的搜索结果集，生成一个产品属性（质量，功能等）的列表，并汇总他们每个人的意见（褒义，中立，贬义）。与“意见挖掘”同时期出现的概念还有“情感分析(sentiment analysis)”，情感分析是由Das、Chen以及Tong在2001年发表的论文中提出的概念，用来形容在市场情绪分析领域，自动分析预估性文本和预测性评论的方法。在2002年的ACL大会上和2003年的EMNLP大会上，情感分析这一概念再次被多为学者作为论文的主要论点进行发表。之后几年意见挖掘、情感分析、倾向分析成为了文本分析领域和自然语言处理领域的热点研究课题，并成为文本分析领域的主流研究方向。

在国内，文本倾向分析也获得了学术界的关注，各种以文本倾向性分析为主要主题的计算机会议也纷纷在国内举办，其中比较有代表性的会议有2008年，由中文信息学会信息检索专委会主办的中文倾向性分析评测（Chinese Opinion Analysis Evaluation, COAE2008）会议，该会议提出了关于中文倾向性分析的主要任务，阐述了中文倾向性分析的难点和要点。由于中文与英文的不同，没有固定的分割符区分单词，而且中文存在歧义导致中文分词与英文分词的难度不在一个级别，而文本倾向分析又依赖于文本切分和处理的精确性，所以中文的倾向性分析相较于英文来说更加具有挑战性。

可以发现，2001年以后的近几年是文本分析领域发展的重要时期，这期间在各大顶级会议期刊上发表了近百篇有关情感分析和意见挖掘的论文，同时，文本倾向分析的研究也逐渐被应用到生产环境中为人们的生活带来便利，产生这一现象的主要原因包括机器学习领域在自然语言处理中的应用，以及互联网的发展产生了大量的可用来研究的语料数据，同时，人们对软件服务质量的需求提升也促进了该领域的发展。

但是基于机器学习的文本分析算法需要训练标注好的文本获得训练模型，这种方式对文本的特征提取方式依赖很大，最近提出的基于神经网络算法的方式进行文本分析是无监督的训练方式，而且训练的文本模型的准确度更高，本文采用了doc2vec神经网络模型作为文本倾向分析任务的主要模型，并结合LDA主题模型对文本特征进行提取，产生了不错的效果。

1.2.2 基于大数据平台的文本分类综述

上一节提到了文本倾向分析领域的发展，其中涉及到如何高效的对文本特征进行分类的问题，随着互联网技术越来越成熟，互联网与人们的生活息息相关，但是随之而来的是以指数级速度增长的信息数据，普通的计算节点已经无法承受如此规模的计算任务，2003年Google公司发表了MapReduce计算模型，该论文阐述了如何使用廉价的计算节点构造计算集群来处理大数据计算任务，2005年Apache基金会开源了Hadoop分布式框架，该框架提供了一套进行MapReduce计算的分布式系统，这使得各个研究机构和公司可以应用大数据计算来处理相关的课题和业务，同时大数据也成为研究热点进入人们的视线。

2009年Spark项目在加州大学伯克利分校AMPLab成立，并在2014年成为Apache的顶级项目，其出色的运行效率以及丰富的操作迅速成为大数据计算中的佼佼者，并被应用到各个领域中，在文本分类领域，大数据平台也被应用到其中作为提高训练速度的工具，同时，企业和用户对文本处理任务的需求越来越多，也促进了以文本处理为核心的大数据平台的推出，比较具有代表性的是2016年Google推出的Cloud Natural Language API，以及2016年微软推出的Natural Language Processing Tools，这两个工具包都是针对NLP领域的并行计算工具，其中容纳了许多自然语言处理相关的算法，包括文本表示、特征提取、文本分类等相关算法。

但是，这两个工具包并没有提供针对中文文本分析接口，本文采用Spark大数据平台实现了针对中文文本的倾向性分析系统，利用spark的分布式计算能力，完成中文文本的切分、过滤、特征提取以及模型训练等过程。

1.3本文的研究内容

(1).文本数据预处理研究

* 针对特定文本数据的标签过滤提取

为了获取真实的互联网数据进行模型训练，常常采用互联网爬虫的方式摄取文本数据，但是通用的爬虫对文本的处理只是简单的过滤掉html标签，保留其正文部分，这种摄取方式打乱了文本的关联性，而且增加了语料的噪声，训练的模型往往准确率低，满足不了实际的应用需求。针对以上问题，本系统设计了一种带标签过滤的定制化爬虫，这种爬虫对于特定的文本内容，只抽取与该主题相关的文本信息，并对特定标签进行过滤，保留与主题相关的特征项，这种摄取方式可以保证主题不会受到冗余信息的干扰，而且通过对特征标签的分类，可以细粒度的判定主题的特征。

* 文本数据的存储

互联网上80%的数据是文本数据，而且是非结构化的文本数据，现有的关系型数据库以及无法满足这些海量数据的存储，而且简单的将文本数据存储到本地磁盘也是不现实的，因为本地文件系统是针对现有的操作系统设计开发的，大多不支持分布式存储，而且本地文件系统的容量有限，可扩展性和可维护性差，为了解决以上问题，需要将海量文本数据存储到分布式文件系统上，HDFS是hadoop的分布式文件系统，满足对海量数据的分布式存储，而且容错性和稳定性良好，非常适合存储非结构化的文本数据，同时，HDFS可以兼容其它大数据平台系统，为之后的处理分析提供了便利条件。

* 文本语料的预处理

Spark是新一代的大数据平台，其天生优势是可扩展性强，并且兼容HDFS。相对于HDFS而言，Spark的RDD技术是更加抽象的支持容错和并行的数据结构，可以让用户显示地将数据存储到磁盘和内存中，并能控制数据的分区。同时，RDD还提供了一组丰富的操作来操作这些数据，例如map、flatMap、filter等转换操作，除此之外，RDD还提供了诸如join、groupBy、reduceByKey等更方便的操作，以支持常见的数据运算。

(2).针对文本分析的中文语料分词算法的研究与优化

中文分词是中文文本处理的第一阶段，由于中文的特殊性，词与词之间是没有分割符间隔的，这就使得中文分词过程相对于其他文字分词异常的繁琐，目前主流的分词算法是基于词库获得切分结果，由于中文分词受限于语料和词库，所以切分效果差强人意，而且分词算法是计算密集型任务，目前大多分词器都是基于单机环境设计实现的，分词效率受限于单个节点的处理能力，如果可以将分词并行化，不仅可以构建足够庞大的词库和语料库，使得分词更加准确，而且可以提高单个结点分词效率达到快速分词的效果。

中文分词并行化需要将分词中使用的算法进行并行化处理，例如最大匹配算法、字典树算法、HMM算法等，这些算法都是分词过程中解决不同问题的有利工具，如何将分词并行化，就是解决如何将这些算法并行化的问题。首先，本课题需要对中文分词中用到的各种算法进行深入研究，确定各算法的优缺点，其次，需要根据实际情况将这些算法进行并行化处理。最后，将并行化后的算法移植到大数据平台下，并通过实验验证其效果。

(3).针对文本倾向分析相关算法的研究与改进

文本倾向分析涉及到自然语言处理的诸多算法，如何让计算机理解自然语言一直是学术界探讨和研究的问题，由于自然语言的表达方式非常丰富，机器无法识别文本的语义，而仅仅是将文本看作数据处理，然而语义识别可以应用到诸多领域，对计算机科学的发展具有深远意义，目前语义识别算法大多都是潜层语义识别，利用机器学习的相关算法分析语义相关度，效果在某些特定领域还是比较理想。例如通过语言模型进行文本特征抽取获得文本的特征，再通过相应的算法进行训练最后可以得出很多语义相关性的应用。本系统针对文本的语言模型提出采用融合的文本表示方法来解决文本倾向分析问题，通过利用doc2vec模型获得文本特征向量并进行训练，来完成文本倾向性分析任务。

Spark大数据平台自身包含一个机器学习库MLIB，其中包含许多文本分析相关的机器学习算法，主要由以下部分组成： 基本的统计学算法、分类和回归算法、聚类算法、降维算法。本课题采用LDA算法对文本进行特征提取，并结合doc2vec算法对文本倾向分析进行优化，提高准确率。

(4).基于大数据平台完成中文分析系统的设计与实现

中文文本分析系统是搭载在spark大数据平台上，利用大数据平台的分布式计算能力和丰富的组件构建一个专门提供中文文本分析解决方案的系统，为用户提供命名实体识别、文本关键字提取、相似度分析、舆情和情感分析等文本处理里常用的功能，并且满足用户对大数据存储和访问的需求。

本课题需要熟练的掌握Spark大数据平台的使用和原理，并且可以在其基础上进行二次开发，充分理解Spark的spark streams、spark SQL、spark MLIB、spark Graphx库，尤其是Spark RDD和spark MLIB，并且掌握spark的任务下发和管理模型，spark API编程等技术要点，同时，需要对中文文本处理相关算法非常熟悉，可以根据spark的特点以spark的编程接口实现这些算法，这些都是本课题实现大数据中文文本分析系统的基础。

1.4本文的组织结构

根据以上阐述的内容，本文的组织结构安排如下:

第一章 绪论部分。本章主要对文本分析理论的研究和大数据平台背景进行阐述，并分析了国内外在相关领域的研究成果和本课题的创新之处，最后结合研究背景归纳了本文的主要研究内容。

第二章 中文文本分析算法及关键技术。本章主要是对中文文本分析算法和关键技术的介绍，详细的介绍了和本课题相关的文本分析算法的细节，通过本章节的介绍可以为后面的章节打下理论基础。

第三章 应用大数据平台处理文本数据。本章主要介绍Spark大数据平台的特点及内部原理，通过本章的介绍，阐述了为什么要使用大数据平台处理文本数据，以及如何使用大数据平台处理文本数据来提高计算效率。

第四章 中文文本倾向分析系统的设计。本章介绍了中文文本倾向分析系统的总体设计思路，以及总体框架设计。通过对文本倾向分析系统进行需求分析，总体框架设计，倾向分析算法设计三个小结来阐述整个系统的设计过程。

第五章 中文文本倾向分析系统的实现。本章节给出了整个系统的实现过程，通过对各个模块的分析以及核心代码展示来阐述系统的实现过程。其中包括文本预处理模块的实现，文本存储模块的实现，文本分析模块的实现。

第六章 系统展示与结果分析。本章主要是对系统的结果进行分析，并给出系统最终的展示效果。

第七章 总结与展望。本章是对整个课题的总结，并给出对研究课题的进一步展望。

2.中文文本分析算法及关键技术

2.1 文本分析相关知识点介绍

文本分析中涉及了许多自然语言处理的专业术语，这些术语是研究文本的基础，本小结通过三方面来讲解这些关键知识点，其中包括语言模型及文本表示方法、文本特征及特征抽取方法、文本分类及分类模型介绍。这三个方面是进行文本倾向分析过程中必须经历的三个关键过程，通过对文本的一系列处理，最终可以得到训练后的模型，之后可以通过系统提供的接口，根据具体的需求对训练模型进行应用，完成实际的任务需求。

* + 1. 语言模型及文本表示方法

(1). 语言模型

语言模型是用来对自然语言进行建模的方式，传统的语言模型是统计语言模型（Statistical Language Model）,它是一个表示语言片段的概率分布函数，其数学表达形式如下：

其中W==表示由T个词按顺序构成的语言片段，则p(W)代表这些词组合在一起的概率。根据贝叶斯公式，可以将p(W)链式的分解为：

统一将每个词的上下文记作Context，则可得到(1)式的表达形式。根据对Context的划分策略不同，可以形成不同的语言模型，常见的有n-gram模型、n-pos模型、决策树模型、最大熵模型、最大熵马尔科夫模型、神经网络语言模型等方法，对于文本的不同建模方式会有不同的效果，同时各语言模型也各有特点，以下是对各语言模型的介绍：

* N-gram模型

n-gram模型认为某个词出现的概率与其前面n个词有关，最极端的情况是n=1的时候，即某个词出现的概率只与该词本身有关。这种语言模型称为上下文无关模型，即:

当n>=2时是称为上下文相关模型，在一般应用中取n=2或n=3，即Bigram或Trigram。n-gram模型的优点是考虑了前n-1个词的因数，而这n-1个词在自然语义上有很强的意义，同时采用Bigram或Trigram的方式可以大大简化计算规模，提高效率。但是n-gram语言模型自身也有一定的局限性，例如：n-gram模型依据语料的关系，语料不足训练的结果一般不理想，而且这种模型忽略了词之间的相似关系，只考虑了词与上下文的关系而没有考虑词与词之间的关系，其次n-gram模型在有些元组没有出现过的情形下会出现统计概率为0的情况，这会导致整个语言序列的概率为0，出现这种情况往往需要进行修正才能获得准确的结果。

* N-pos模型

n-pos模型是基于n-gram模型的基础上衍生的一种语言模型，n-pos模型是基于这样的假设：单单考虑前n个词不足以表示当前词的特征，而自然语言中的词语搭配往往是根据词的语法功能决定的，所以n-pos将当前词的前n个词按照语法功能进行分类，由这些词来决定当前词的概率，该分类叫做Part-Of-Speech，即n-pos算法的由来，n-pos模型的条件概率公式如下：

c是词性映射函数，表示将单词映射为其所属的词性类别，如果一段语言序列有T个词，K种词性分类，则可以将n-gram的条件概率求解过程从可能变成种，大大的提高了计算效率，而且这种效率的提高也不会影响准确率的下降。

* 决策树模型

决策树模型是为了解决n-gram模型中对相似的词类搭配重复求解导致的效率损失问题，构造一颗决策树表示当前词的上下文，由对根节点提问的不同回答进入子节点，直至叶子节点，从而得到当前词上下文的分布信息。为构造决策树，需要预先定义一个关于上下文信息的问题集和一个评论问题的优劣函数。

决策树模型的优点是该模型是一种通用的语言模型，n-gram模型、n-pos模型都可以用决策树模型表示出来，而且决策树中的分布信息不是固定好的而是根据训练语料库得到的，但是，由于决策树模型的抽象性导致这种模型的构造难度偏大而且时间复杂度和空间复杂度都很高。

* 最大熵模型

最大熵模型的基本思想是：对一个随机事件的概率分布进行预测时，若概率模型需要满足一些约束，则在满足约束的情况下对未知的情况不做任何假设，这种情况下，概率分布最均匀，得到的概率分布的熵也最大。最大熵语言模型的概率分布公式如下所示：

其中是参数，Z(context)为归一化因子。

* 神经网络语言模型

神经网络语言模型(Neural Network Language Model)是近几年比较流行的语言模型，即通过神经网络算法来拟合语言模型的方式求概率分布的最大似然估计，NNLM采用的是Distributed Representation来表示词向量，神经网络语言模型与n-gram、n-pos、决策树模型、最大熵模型最大的区别是前者是基于统计的语言模型，后者是基于深度学习的语言模型，这些模型所关注的结果都一样，但是建模思路不尽相同.

常用的NNLM模型有C&W的SENNA、M&H的HLBL、Mikolov的RNNLM，其中比较经典的是Bengio等人在2001年发表在NIPS上的文章《A Neural Probabilistic Language Model》，该文章用一个三层的神经网络模型来构建语言模型，这三层网络分别是输入层，即以词向量首尾相连得到一个很长的向量作为输入向量，第二层是隐藏层，使用d+Hx计算所得，d是一个偏执项，然后使用tanh作为激活函数，最后一层是输出层，共有|V|个节点，其中|V|表示词表的大小，每个节点表示下一个词为i未归一化log概率。最后使用 softmax 激活函数将输出值 y 归一化成概率。整个模型的计算公式为:

式子中的 U是隐藏层到输出层的参数，整个模型多数的计算集中在 U和隐藏层的矩阵乘法中。式子中还有一个矩阵 W，这个矩阵包含了从输入层到输出层的直连边。直连边就是从输入层直接到输出层的一个线性变换，如果不需要直连边的话，将 W置为 0 就可以了。

(2).文本表示方法

文本表示方法是指将文本向量化，转换成可以计算的数学矩阵，一般是将文档作为矩阵中的行(如上图)，单词作为矩阵中的列，而元素就是需要数字化的表示方式，在机器学习领域常用的表示方法是Bag-Of-Word(BOW)模型和Bag-Of-N-gram模型，而在深度学习中常用的模型是Word Representation 或Word Embedding, 后者通常叫做词向量模型。以下是对文本表示方法的介绍：

* 词袋模型

词袋模型是进行文本分类中经常用到的文本表示方式，因为这种方式方便快捷，易于理解，但是同时词袋模型忽略了词语之间的顺序，所以词袋模型一般应用在那些对词序要求不高的文本分析领域，词袋模型通常需要一个词典保存文档集合中所有出现过的单词，并对单词进行编号，然后文档可以表示成一个一维向量，其中向量维度为词典的大小，向量元素一般单词的出现的频率(也可以用TF-IDF表示)，以下通过例子来讲解词袋模型的原理。

假设有两个文档，文档1为“我 爱 北京 天安门， 我 是 中国人”，文档2为“她 喜欢 看 电影， 她 是 美国人。”则根据文档1和文档2可以构造一个词典：{1:“我”，2:“爱”，3:“北京”，4：“天安门”，5：“是”，6：“中国人”，7：“她”，8：“喜欢”，9：“看”，10：“电影”，11：“美国人”}，根据词典大小可以将文档表示成为一个11维的向量，其中每个元素是单词在词典中出现的频率。

文档1：[2,1,1,1,1,1,0,0,0,0,0]

文档2：[0,0,0,0,2,0,1,1,1,1,1]

为了提高词袋模型的实用价值可以将文档中的元素用单词的TF-IDF值表示，这样可以提高单词和文档的特征关系。

* Ngram词袋模型

以上提到的词袋模型是最简单的文本表示方式，其没有考虑单词与单词之间的搭配关系，这种表示方式丢失了很多文本特征，在文本分类过程中使用这样的模型得到的准确率往往不高，Ngram词袋模型是对这种模型的优化，即使用n-gram语言模型得到单词的条件概率，用该条件概率表示单词和文档，这种表示方式保留了单词之间的搭配关系，但是无法获得单词和文档之间的关系，即无法获得语义相近的单词，丢失了文档与文档的特征，模型不够平滑。为了使得模型更加平滑，还需要采用平滑处理来优化模型。

* 词向量模型

词向量是指将自然语言中的词用数学化的向量表示，这样做的好处是可以对词进行数学公式计算，把计算机无法理解的文字数字化，常见的对词向量的划分有one-hot representation，就是用一个只有一项为1其他项全为0的长向量表示一个词，向量的维度N为词典的大小，向量中为1的项对应该词在词典中的索引，例如对于“中国”这个词，假设其在字典中的索引为2，这”中国”可以表示成如下的词向量：

这种词向量表示方法虽然简单，但是其只标识了单个词的信息，向量十分稀疏，而且向量的维度很高，这样会导致维数灾难，不适合作为计算机运算的输入，这种表示方式还有一个问题是向量中没有词与词之间的关系，即词的上下文关系，所以在训练过程中会丢失许多特征信息，其数学意义不大，所以很少被使用。

另一种对词向量的表示是Distributed Representation，其基本的思想是：通过训练将自然语言中的词映射成为一个长度固定的向量，该向量维度一般比较短，这样所有的词向量可以构成一个向量空间，同时由于向量长度较短，也有利于对其进行相应的数学运算从而研究它们之间的关系。这种词向量表示方式比较符合计算机输入参数的要求，且自带平滑，保留了词的上下文特征和词与词之间的特征，是比较理想的文本表示模型。

2.1.2 文本特征及特征抽取

文本特征是表示文本的基本单位，文本的特征抽取是文本挖掘领域中重要的研究课题，对于文本来说其特征一般都具备一定的特性。首先，文本特征项需要能够标识文本的主要内容。其次，特征项需要代表文本的特征属性，即与其他文本的特征有明显区别。最后，特征项的维度不宜过多，容易导致维度灾难。在中文领域，中文文本由字作为基本单元组成，但是字所包含的特征不足够表示文本的特征，相对于单个字来说，词和短语所包含的信息更加丰富，所以一般采用词或短语作为文本的基本特征单元。如果用词或短语作为文本的特征项，随着文档的规模越来越大，特征向量包含的特征值就会非常巨大，这样会导致维度灾难，无法进行计算。这时就需要借助特征抽取技术来完成文档的特征抽取(或文档的稀疏表示)，特征抽取需要完成的任务是在不丢失文本核心信息的情况下，用更少的特征表示文本，即将原来稠密矩阵用稀疏矩阵进行表示。

特征抽取的方式一般有3种：（1）通过映射或变换的方法减少特征的维度，（2）通过排序获取最具代表性特征值，（3）用机器学习的方法训练文本获取特征模型。

以下是基于统计的特征提取方法：

1. TF-IDF:

TF-IDF(term frequency–inverse document frequency)算法是用来衡量单词权重的统计方式，其中TF指为单词频率，IDF为逆文档频率，即该词出现在文档中频率的倒数，该参数用于计算该词区分文档的能力。TF-IDF之所以能作为衡量单词权重的特征是因为其基于这样的思想：如果单词在一篇文档中出现的频率很高而在其他文档里出现的概率很低，那么说明这个单词对这篇文档很重要，也就是该单词能作为这篇文档的特征。TF-IDF通常用来提取文档的关键词特征，即通过排序获取TF-IDF值高的单词作为文档的关键词来代表文档的特征。

1. 互信息（Mutual Information）

互信息(Mutual Information)是信息论里一种有用的信息度量，它可以看成是一个随机变量中包含的关于另一个随机变量的信息量，或者说是一个随机变量由于已知另一个随机变量而减少的不肯定性。互信息的定义如下：

其中X,Y为随机变量，其联合分布为p(x,y),边际分布为p(x),p(y),互信息I(X,Y)是联合分布p(x,y)与乘积分布p(x)p(y)的相对熵。

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

1. 信息增益方法（Information Gain）

信息增益是信息论中的一个重要概念, 它表示了某一个特征项的存在与否对类别预测的影响, 定义为考虑某一特征项在文本中出现前后的信息熵之差。某个特征项的信息增益值越大,贡献越大, 对分类也越重要。信息增益方法的不足之处在于它考虑了特征未发生的情况。特别是在类分布和特征值分布高度不平衡的情况下, 绝大多数类都是负类, 绝大多数特征都不出现。此时的函数值由不出现的特征决定, 因此, 信息增益的效果就会大大降低。信息增益表现出的分类性能偏低。因为信息增益考虑了文本特征未发生的情况，虽然特征不出现的情况肿可能对文本类别具有贡献，但这种贡献往往小于考虑这种情况时对特征分值带来的干扰。

1. 遗传算法（Genetic Algorithm, GA）

文本实际上可以看作是由众多的特征词条构成的多维空间,而特征向量的选择就是多维空间中的寻优过程,因此在文本特征提取研究中可以使用高效寻优算法。遗传算法(Genetic Algorithm, GA)是一种通用型的优化搜索方法,它利用结构化的随机信息交换技术组合群体中各个结构中最好的生存因素,复制出最佳代码串,并使之一代一代地进化,最终获得满意的优化结果。在将文本特征提取问题转化为文本空间的寻优过程中,首先对Web文本空间进行遗传编码,以文本向量构成染色体,通过选择、交叉、变异等遗传操作,不断搜索问题域空间,使其不断得到进化,逐步得到文本的最优特征向量。

1. 主成分分析法（Principal Component Analysis，PCA）

主成分分析法不是通过特征选取的方式降维的，而是通过搜索最能代表原数据的正交向量，创立一个替换的、较小的变量集来组合属性的精华，原数据可以投影到这个较小的集合。PCA由于其处理方式的不同又分为数据方法和矩阵方法。矩阵方法中，所有的数据通过计算方差一协方差结构在矩阵中表示出来，矩阵的实现目标是确定协方差矩阵的特征向量，它们和原始数据的主要成分相对应。在主成分方法中，由于矩阵方法的复杂度在n很大的情况 以二次方增长，因此人们又开发使用了主要使用Hebbian学习规则的PCA神经网络方法。

主成分分析法是特征选取常用的方法之一，它能够揭示更多有关变量\_丰要方向的信息。但它的问题在于矩阵方法中要使用奇异值分解对角化矩阵求解方差一协方差。

1. n-gram算法

n-gram算法的基本思想是将文本内容按字节流进行大小为N的滑动窗口操作,形成长度为n的字节片段序列。每个字节片段称为gram,对全部gram的出现频度进行统计,并按照事先设定的阈值进行过滤,形成关键gram列表,即为该文本的特征向量空间,每一种gram则为特征向量维度。由于n-gram算法可以避免汉语分词的障碍,所以在中文文本处理中具有较高的实用性。

特征提取是为了获取文本的特征，需要根据特定场景需求选择特定的特征提取方式，对特征的提取不局限于单一的方法，往往需要通过多种方法对基础语料进行处理才能获得比较理想的特征项，本课题采用特征抽取的第三个方法，即通过机器学习方法训练文本获取特征模型，本文采用LDA模型作为文本特征提取算法，LDA算法是以N-gram算法为基础的文本表示算法，其可以将文本进行稀疏表示获得文档与主题模型，该模型可以作为文档特征进行文档分类任务。

* 1. 基于主题模型和文本向量的文本倾向分析算法

本节主要介绍本课题使用的文本倾向分析算法，通过上一个章节的介绍，对文本分析领域的基本概念有一定的了解，文本分析需要对文本建模，并提取特征，之后需要对文本进行训练获取模型，然后通过文本分类算法对文件进行分类，获取文本的正负倾向性。本文采用LDA模型获取文本主题作为文本特征的一部分，然后结合神经网络模型训练获取词向量，并训练得到word2vec模型，最后通过随机梯度下降算法对文本进行分类，以下是对这三个算法的原理介绍。

2.2.1 基于LDA的文本特征提取算法

2.2.2 基于word2vec的文本表示算法

上一节中提到了NNLM模型，word2vec是基于NNLM模型训练语言模型的算法，Word2vec算法是通过神经网络语言模型拟合自然语言概率模型，从而获得计算语言概率模型的高效方法，在训练过程中，word2vec会获得语料库中每个词的词向量，该词向量是Distributed Representation类型的词向量，其维度由输入参数决定，一般在100-500维。根据词向量我们可以得到一个词向量空间，该词向量空间包含了词本身的特征以及词与词之间的特征，可

本节主要阐述word2vec的算法原理，word2vec中主要运用了CBOW模型(Continuous Bag-of-Words Model)作为其语言模型，其中CBOW模型是在已知当前词的上下文,,,的前提下预测当前词的概率。由上一节可知自然语言的语言模型为（1）式所示，而基于神经网络的语言模型通常取其对数似然函数作为目标函数：

(2)

word2vec构造了一个三层的神经网络：输入层、投影层和输出层，这三层的含义分别是：

(1).输入层：包含Context()中2c个词的词向量，即：

v(),…,v(),v(),v(),v(),…,v()。

(2).投影层：将输入层的向量做求和累加，即:

(3).输出层：输出层对应一课Huffman树，Huffman树是以语料中各词的词频为权值构造而成的，其叶子结点为语料中的词。

构造了以上神经网络模型后，主要的问题是解决如何通过以上神经网络模型来拟合条件概率，word2vec将其视作一个二分类问题，利用逻辑回归的方法求每个结点的概率，即从Huffman的根结点到叶子结点的过程，将左结点分为负类，右结点分为正类，利用逻辑回归，易知，一个结点被分为正类的概率是：

显然，其被分为负类的概率就是：

其中，为sigmoid函数,其作为神经网络的激活函数，为词向量，为参数向量，其代表Huffman树中的非叶子结点向量。

根据以上定义就可以得到计算的算法：对于词典D中的任意词w,Huffman树中必定存在一条从根结点到词w的路径。路径上存在个分支，将每个分支看做一次二分类，每一次分类就产生一个概率，将这些概率乘起来，就是所需的。

于是，可以得到条件概率的公式为：

其中:

表达式中，表示路径中第j个结点对应的Huffman编码，表示路径中第j-1个非叶子结点对应的向量。

将上式带入对数似然函数(2)，可得：

其中，记L(w,j)为：

则上式即为CBOW模型的目标函数，使得该函数最大化，当超过某个阈值时，就可以用来判断语言模型是否是自然语言，Word2vec通过采用随机梯度上升法来计算该目标函数。

基于神经网络语言模型的训练，其是无监督的训练算法，不需要进行人工识别，只需要将文本切分成单个词语，将切分后的语料进行训练就可以得到语言概率模型，大大节约了训练成本，而且神经网络语言模型通过滑动窗口的模式保留了词语上下文的关系，这样对语义分析也有很重要的意义。

同时，对于word2vec神经网络语言模型中涉及的参数，都是由样本训练得到的，这有别于通常的机器学习算法，需要在训练样本前给定参数，所以词向量和概率模型都是在训练中得出，词向量作为训练过程中的副产品对解决自然语言领域中的问题有极大的参考作用，同时，其无监督的训练方式和高效的训练速度，也是word2vec算法受到人们青睐的重要原因。

2.2.3 基于SGD的文本分类算法

2.3 文本分析的关键技术

文本分析是一个综合学科，其覆盖了许多知识点，除了前几节提到的有关算法方面的内容，还包含文本采集技术、文本切分技术、文本标注技术等，这些技术是进行文本分析的基础，掌握这些关键技术，在处理复杂的文本分析任务时会游刃有余，不仅可以提高效率而且还可以灵活的应对不同的需求，本节通过三个方面来对文本采集技术，文本切分技术和文本标注技术进行阐述。

2.3.1 文本采集关键技术

文本采集通常是采用爬取技术爬取网络上的数据进行分析，或者是收集自有的日志数据、数据库数据进行分析，网络爬虫的优点是可以根据需求收集各种类型的文本数据训练模型，缺点是网络上的文本数据大多是没有进行标注的文本，而且存在大量的网络用语，需要进行清洗才能满足训练要求。

网络爬虫一般分为通用爬虫和定制爬虫两种类型，通用爬虫一般是为搜索引擎等系统设计的爬虫，其目的是快速的爬取互联网上的内容，而定制爬虫一般是针对某一特定领域或网站进行爬取，目的是获取特定的资源，在效率方面定制爬虫不如通用爬虫那样高效，但是在质量上，定制爬虫比通用爬虫更加出色。

作为文本采集任务，一般采用的是定制爬虫来采集文本信息，在效率容忍的范围内可以获取到符合需求的训练语料，定制爬虫一般都具有对特定的网页内容进行过滤的能力，这样可以精确的获取的想要的文本内容，达到初次过滤文本的目的，例如对于网站的评论信息，定制爬虫可以将评论主题，评论时间等有关评论的文本爬取下来，并对个字段分别保存，而通用爬虫一般只是单单的爬取整个网页，或者是过滤掉HTML标签，这样获取的语料带有许多无关信息，对模型训练或文本进一步分析都是额外的噪声，所以文本采集任务更适合使用定制爬虫。

2.3.2 文本切分关键技术

文本切分关键技术是针对中文文本而言的，因为中文文本没有分隔符来区分文本之间的单词，所以需要进行分词处理，将整个文本切割成由单词组成的文本序列，中文分词技术是自然语言处理里非常重要的技术，同时，也是进行文本分析的前提条件，常见的分词算法是基于词典和规则的最大匹配算法，这类算法一般需要有强大的词库用来进行分词，但是，对于词典中不存在的词，这类算法就会失去作用，而且对于存在歧义的文本序列，这类算法同样不能达到满意的效果，于是，出现了基于概率统计的分词算法，其中最经典的是使用HMM算法进行分词，HMM算法可以消除对歧义词语切分不准的问题，而且还可以切分出未登录的词，所以现代的中文分词程序都是结合两种方法进行分词，而且分词效果也十分出色，一般能达到98%的准确率。

2.3.3 文本标注关键技术

文本标注一般是在文本分类训练过程中对语料进行标注，形成熟语料，从而可以训练所需的模型，文本标注任务根据需求的不同标注方法也不尽相同，一般的标注方式有三种：人工标注，基于特征词典的文本标注，基于标签的文本标注，以下是对三种标注方式的解释：

(1).人工标注：顾名思义，人工标注就是采用人工的方式对文本进行标注，这种方式的标注成本很高，但是标注效果好，一般是对标注要求很高的需求才会使用人工标注的方式。

(2).基于特征词典的文本标注：基于特征词典的标注一般是采用一个特征词典，根据分词匹配结果计算文本的最终标注结果，特征词典一般包含特征的级性，采用线性叠加的方式计算文本片段，计算结果就是文本的标签值，采用特征词典的文本标注适用于标签比较少的请求，例如情感倾向标注就适合使用该标注方式。

(3).基于标签的文本标注：标签标注是通过文本在源文本中的标签特性进行标注，一般依赖原文本的标签值，例如在爬取评论信息时，根据评论的星级以及点赞次数，可以标注该评论文本的倾向和情感强度，这种标注方式依赖源文本的标签，如果原文本没有明显的标签就需要结合人工识别的方式指定特定的标签作为标注特征，再结合特征词典的方式会达到很好的效果。

3.应用大数据平台处理文本数据

文本处理任务是一个计算密集型任务，而且通常需要训练的语料库非常庞大，动辄几十G，上百G的语料需要训练，而且随着互联网的更新迭代速度的加快，数据的产生以指数级的规模不断的扩大，单机环境已经无法满足文本数据的处理需求，所以十分需要借助分布式计算能力来提高计算效率，随着大数据技术的发展，大数据计算平台越来越成熟，同时也越来越受到业界的青睐，使用大数据平台来处理海量文本数据已经成为许多企业首选的解决方案，本节主要通过介绍Spark大数据平台来阐述使用大数据平台的优势，并给出如何使用Spark处理文本数据的步骤。

3.1 Spark大数据处理平台

Spark是一个围绕速度、易用性和复杂分析构建的大数据处理框架。最初在2009年由加州大学伯克利分校的AMP Lab开发，并于2010年成为Apache的开源项目之一。Spark作为现今最流行的内存分布式计算框架，其优点远远不仅仅是作为大数据的计算中心这么简单，它已经形成一套完整的大数据处理生态系统，其中包括Spark Streaming:用来处理实时数据，Spark SQL:用类SQL语句对Spark数据进行查询，Spark MLIB:可扩展的机器学习库,包括常用的机器学习算法,Spark Graphx:支持图计算和并行图计算的Spark库。

Spark大数据计算框架是建立在MapReduce基础之上的分布式计算框架，但是相较于Hadoop简单的Map和Reduce操作，Spark提供了一个分布式内存抽象数据集RDD,针对RDD还提供了更加丰富的算子操作，这些操作被划分为转换（Transformations）和动作（Action）。转换操作是对RDD进行的用来返回一个新的RDD的算子，包括map、reduce、union、filter、reduceByKey、partitionBy、join、count、sample等，动作是在数据集上经过一次计算后返回的结果。所有的Spark应用都是由驱动程序组成，这些驱动程序用来并行执行用户提交的计算任务，计算任务又由各个单向的转换操作组成，在Spark中，所有RDD的转换都是惰性求值的，旧的RDD数据经过转换后生成新的RDD集合，每个RDD又可以有多个分区，这样每次RDD转换相当于生成一个结点到另一个结点的有向无环图(DAG),由多个DAG构成一个Spark任务，同时，Spark对于DAG任务进行优化处理，确定阶段、分区、流水线、任务和缓存，数据按照DAG的方向流动，并当遇到Action类型的算子后进行实际运算，这样数据的流转都在内存中进行，大大提高了计算效率。下图展示了Spark的计算模型。



图1 Spark计算模型示意图

3.2 Spark大数据平台的优势

* MapReduce

MapReduce是一种抽象的编程模型，通过将复杂的数据处理分解为Mapper和Reducer两个过程，然后将这些分解后的计算job分发到几十到几百台服务器上分布式的运行，同时对外隐藏了并发、分布、故障恢复、集群通信等细节问题，从而可以处理大数据集的计算问题。

MapReduce过程是将复杂问题抽象化，隐藏抽象处理的过程，这些抽象的细节被称为shuffle过程，shuffle过程是MapReduce过程中非常重要的一环，当问题被分解为以Mapper和Reducer构成的有向无环图(DAG)时，各个job的最终执行过程是受shuffle过程影响的，如果分解合理会大大提高执行的效率，反之，即使经过了MapReduce过程可能也不会达到预期的效果。下图中map和reduce中间的过程就是shuffle过程。



所以，MapReduce也存在一定的局限性，首先MapReduce的抽象层次比较低，只提供Mapper和Reducer两个操作，对程序员不够友好，很难上手写MapReduce程序。其次，一个复杂的任务需要分解为多个job进行计算，job的管理需要开发者自己维护，如果设计的不够合理会导致效率下降，而且处理的实际逻辑隐藏在shuffle过程中，不受开发者的管理，出现问题需要修改MapReduce过程，维护成本很高。最后，Reducer需要等待所有Mapper执行结束后才能执行，且中间结果需要保存到HDFS上，没有完全利用集群的内存资源。

针对MapReduce的局限性，研究者在MapReudce的基础上进行了许多改进，出现了如Pig，Cascading，JAQL，OOzie，Tez等技术，Spark是其中比较新的一种技术，而且基于其出色的表现能力，逐渐成为独立的大数据计算框架。

* 从MapReduce到Spark

Spark是基于MapReduce的大数据内存计算框架，其主要特点是有一个抽象的分布式内存对象RDD(Resilient Distributed Dataset)，RDD作为Spark编程模型的核心，具有很多丰富的操作，这些操作可以分为两类：Transformations和Action。Transformations包括map, flatMap, filter, union, sample, join, groupByKey, cogroup, ReduceByKey, cros, sortByKey, mapValues等，Action包括collect，count，save，lookupKey等。

Spark作为新型的计算模型相对于Hadoop来说具有很多优点，首先，Spark是基于内存的抽象计算模型，在性能上比Hadoop的MapReudece有很大的提升，2014年10月，Spark完成了Daytona Gray排序测试，该测试结果显示Spark用相对于Hadoop1/10的计算资源完成该项测试的耗时比Hadoop的MapReduce过程的耗时快3倍。以下是该次测试的实际结果对比：



其次，Spark中的RDD都是惰性求值的，一个RDD会转换成新的RDD，RDD之间不是强依赖关系，对于非依赖的RDD计算，并不需要等待其运行结束就可以执行其他运算。所以Spark对迭代式计算有更好的支持。最后，Spark具有丰富的扩展，其不单单是一个内存计算模型，还包含交互式（Spark SQL），流式（Spark Streamng），机器学习（MLLIB）,图形计算（GraphX）等模块，所以Spark已经成为一个大数据计算平台，许多计算任务都可以在Spark上进行运行。下图是Spark模块的构成：



3.3 如何使用Spark大数据平台处理文本数据

以上两节分别介绍了Spark大数据平台和Spark的优势，本节针对如何使用Spark大数据平台处理文本数据这一问题进行阐述，根据第二章介绍的有关文本分析的知识可知，文本分析主要的计算任务是需要对文本语料进行预处理和模型训练。文本预处理包括文本切分、文本过滤、文本标注等，文本模型训练包括文本特征训练、文本特征抽取、文本分类模型训练等。通常情况下，这些文本任务都是通过不同的模块进行分别处理，而基于Spark平台的支持，这些任务可以结合在一起完成。

首先，对于初始语料的处理，Spark提供了针对Java、Scala、Python的接口，可以选择其中的任何一种语言对语料进行处理，而且Spark也提供了对语言内部数据结构和RDD相互转换的接口，所以，可以通过现有的语言编写好对语料的处理逻辑，然后根据Spark提供的接口，将处理逻辑转换成Spark内部的数据结构，即RDD内存模型，然后进行分布式的处理。下图是以初始语料集处理为例，展示如何将文本处理任务运行到spark平台上：



初始语料可以保存在HDFS上，从HDFS中将语料文本导入Spark系统，构造初始语料RDD，通过map操作将语料RDD切分成词序列保存到新的RDD里，这里记作RDD1，Spark的map操作可以通过函数将RDD映射到新的RDD上，这里的映射函数就是通过其他语言实现的分词模块，形成的RDD1实际上是一个列表，其中每个元素是一个token对象，该对象包含三个属性：word,offset,nature。其中word指单词本身，offset为该词在语料中的位置，nature指该词的词性。根据nature我们可以通过filter算子获得具有特定特征的词语的集合RDD2，然后，我们通过停用词词典将切分后的语料过滤，剔除无用的停用词，例如“的，是，了”等，形成待训练的新语料RDD3，用来后续的文本分析任务。

以上，就是对初始语料的处理，之后我们可以通过利用Spark提供的MLLIB库对语料进行训练获取语言模型，MLLIB库提供了丰富的机器学习算法，其中就包括第二章提到的用于进行文本分析的算法，对于没有实现的算法，我们同样可以使用外部的机器学习库来训练语料，然后通过将其转换成RDD在进行其他任务。

通过以上的描述，我们知道了使用Spark大数据平台处理文本是非常方便的，同时，基于Spark出色的表现力，还可以提高文本处理的效率，具有开发方便，性能卓越的优点。

4.中文文本倾向分析系统的设计

4.1 需求分析及应用领域设计

4.2 文本倾向分析系统总体框架设计

4.3 文本倾向分析算法的设计

4.3.1基于情感标注和标签融合的文本标注算法的设计

4.3.2基于word2vec和LDA的文本表示算法设计

5.中文文本倾向分析系统的实现

5.1 文本预处理模块的实现

5.2 文本存储模块的实现

5.3 文本分析模块的实现

6.系统展示与结果分析

7.总结与展望