基于词向量和 K 近邻算法的文本分类探究

摘要

随着互联网技术的发展，网络上的新闻报道数量日益增多，如何按照需求对其进行处理并分类是文本分类领域重点关注的问题。在对此类问题进行深入研究的基础上，本文介绍一种基于词向量和 K 近邻思想的文本分类算法（Keywords Embedding and K-Nearest Neighbor Based Document Classification 简称:KKB-DC），该方法能够有效利用语料中词的语义信息，本文主要工作总结如下：

⑴提出基于分布式词向量和 K 近邻算法思想的，以关键词与类别之间的距离为权重，关键词投票决定分类的文本分类算法。实验从语料、向量维度、关键词个数等方面，探究最佳参数配置，实验结果表明，算法在最佳参数配置下，二分类实验分类准确率可达 91.89%，三分类实验准确率为 83.68%。

⑵对二、三分类实验结果进行分析，针对算法在关键词数量较多时分类准确率下降较快这一现象，对提出算法进行分析、改进，使得分类效果更佳稳定。实验证明在算法的权值优化中引入关键词的 tf-idf 值，可以有效解决关键词个数选取较多时分类准确率下降较快这一问题。

⑶与现有分类方法进行实验对比，实验证明本文所提方法在优于现有的 TFIDF+SVM、LDA+SVM 算法，同时与清华大学推出的中文文本分类工具包 THUCTC（THU Chinese Text Classification）实现的文本分类算法进行对比，证明本文所提 KKB-DC 算法的有效性。

关键字：文本分类；KKB-DC；语言模型；Word2vec

Distributed Representation and K-NN Based Document Classification Exploration

Abstract

With the development of Internet technology, large number of news reports are created, how to classify these documents to classify to proper classification is a key point in document classification research area. This paper introduce a classification method based on word distributed representation and K nearest neighbor algorithm named Keywords Embedding and K nearest neighbor Based Document Classification (KKB-DC). This method utilizes word semantic information as much as possible in the training corpus, the main work of this paper are:

(1) Proposing a document classification method based on word distributed representation and K nearest neighbor algorithm. In this method distance between keyword and the category is used as weight, and the final result depended on key words voting. The experiment explores the optimal parameter configuration from the aspects of corpus, vector dimension and the number of keywords. It shows that the classification accuracy of two category classification is greater than 91.89%, and the accuracy of three category classification experiments is greater than 83.68%, with the optimal parameter configuration,.

(2) As the rapid decreasing performance when the key words number increasing in two and three classification, we modified the method by using tf-idf value to optimize the algorithm. Experiments shows the problem has been solved with tf-idf value in weight optimization.

(3)Comparing with the existing classification method, and experiments show that the method proposed in this paper is better than TFIDF+SVM and LDA+SVM. We also compare with the classification model developed in Tsinghua University the Chinese text classification toolkit THUCTC (THU Chinese Text Classification), and the result shows that the effectiveness of the proposed algorithm is better.

keywords: text classification; KKB-DC; language model;Word2vec

目录

第 1 章 绪论

1.1 研究背景及意义

随着计算机技术的发展，特别是近几年互联网技术的不断普及与完善，移动互联网、电子商务和新媒体的快速发展，数字信息不断被使用和创造。根据中国互联网网络中心（简称 CNNIC）2017 年一月发布的第 39 次《中国互联网络发展状况统计报告》[1]，截止至 2016 年 12 月，我国网民规模已经达到 7.31 亿人，全年共计新增网民 4299 万人。网络普及率为 53.2%。在基础应用类应用的发展中，网民中即时通信用户规模达到 6.66 亿，占整体网民的 91.1%；搜索引擎用户规模达到 6.02 亿，使用率为 82.4%；网络新闻用户规模为 6.14 亿，占整体网民的 84.0%；社交应用用户，微信、朋友圈、微博用户使用率分别为 85.8%，67.8%，37.1%。

由这些数据可以看出，网络应用主要集中在：（1）沟通交流：主要通过短信息、微博、电子邮件为代表的社交平台；（2）信息获取：主要涉及搜索引擎与网络新闻；（3）网络娱乐：同样涉及搜索引擎与社交平台。用户与网络应用交互的媒介有图像、视频、文本、语音等。虽然视频、图像的使用以及数量在急速增长，但是文本依然是网民与网络应用的主要交互手段。

以上这些应用都会产生大量文本信息，社交平台如微博、微信、qq 空间等每天都有过亿条消息发布，搜索引擎每天接受的搜索请求也在数十亿次以上。而网络新闻、问答网站、电商评价等每天也在以惊人的速度增加。如何有效地组织处理这些大规模数据信息，更好的服务广大网民、公司以及政府，是非常值得探究的课题。

短消息、商品评论等短文本要有更好的过滤、分类功能。如识别垃圾广告、根据用户的评价汇总获取用户对某一商品的评价等级；搜索引擎根据用户输入的文本信息判断用户的真实搜索意图，这样才能返回更好的搜索结果。网络新闻是现在人们获取新闻资讯的重要途径，对网民舆情的分析对相关部门了解民生、制定制度越来越重要，如政府机构在重大会议举行前展开民众调查，了解民众对某些事件的整体评价以了解民众对这些事件的反应。

所有这些需求，都涉及到一项技术——文本分类。分类是通过相关技术如机器学习等，将信息混乱度降低的一种手段，应用的领域比较广泛，如垃圾邮件检测、人脸识别、声音识别等；文本分类是根据预先指定的主题类别，按照一定的规则或算法将待分类文档集合中的文本自动确定一个或几个类别的方法[2]。目前国内外有众多学者关注此领域的发展。

1.2 国内外现状

文本分类是研究者重点关注的一个领域，也是自然语言处理（Natural language processing，简称 NLP）中一个关键应用领域。文本分类，即对已有文本集合中的每一篇文档进行类别划分，使得每篇文档都尽可能正确的分到相应的类别中去。

1.2.1 国外研究现状

文本分类最早的研究可以追溯到上世纪 50 年代，Luhn[3]首次提出将词频统计应用到文本分类当中，其后也有许多研究者在此方向进行探索并取得一定的进展。1960 年，Maron 等[4]人首先提出了基于关键词进行分类的概念，之后更多学者基于此思想进行研究，文本分类技术得到了更多的研究与应用。1975 年 Salton[5]提出的空间向量模型进一步推动了文本分类技术的发展。80 年代，基于知识工程方法占主导地位，该方法主要是以专家的知识作为规则人工组建分类器以实现文本分类[6]。

二十世纪 90 年代，随着信息技术的不断发展，电子文本的数目日益增加，基于知识工程的自动分类算法已经不能满足人们的需求，因此文本分类逐渐发展到一个新的阶段，基于机器学习的文本分类方法[7]出现并成为文本分类领域内的研究热点。基于机器学习的文本分类技术通过对训练样本的学习，训练得到一个类别区别于其它类别的特征，在归纳这些特征的基础上构建文本分类器。当有待分类文本需要进行分类的时候，将已经构建的文本分类器对该文本进行分类，得到分类结果。相比于基于词频的技术，机器学习应用了机器学习方法之后，文本分类的效果和性能得到了明显的提高，由于这类分类器的训练是基于某些算法而非特定领域的专业知识，因此基于机器学习的文本分类方法具有更好的通用性。常见的有朴素贝叶斯算法、K-近邻算法、决策树、支持向量机等，这些算法均在文本分类领域取得了较好的分类效果。基于机器学习的文本分类研究主要分为三个方面：文本表示、空间维度约减、文本分类器研究。其中，文本分类中最基础的一步就是文本表示的特征选择，如果这一步做的很差，即使再好的分类工具都无法达到更好的提升。因此文本特征的准确提取，成为当今文本分类中的热门研究内容。

二十一世纪以来，特征学习（feature learning）得到了国内外学者的广泛关注。词的分布式（word embedding）表示最早由 Hinton[8]于 1986 年提出，其基本思想就是通过深度学习训练将每个词映射成为 N 维空间中的实数向量，通过词与词之间的距离，如欧氏距离、向量夹角等来判断他们之间的语义相似度。2003 年 Bengio 等人首次将词的分布式表示应用到统计语言模型[9]中，2008 年 Collobert 等[10]人首次介绍了词向量的计算方法，2010 年 Joseph Tutian[11]等人对不同的词向量表示方法进行比较研究。2011 和 2012 年 Xavier Glorot[12]和 Bengio 等[13]人在跨领域学习分类任务上也做了相关研究。2013 年谷歌 Mikolov[14][15]等对连续词袋模型和 Skip-gram 进行扩展，开源词向量学习工具 Word2vec，该工具提供连续词袋模型与 Skip-gram 模型的实现[16][17][18]。

1.2.2 国内研究现状

相比于国外对文本分类的研究，国内相应的研究起步较晚，二十世纪 80 年代侯汉清教授对文本分类作了相关的介绍性工作。直到二十世纪 90 年代中后期才逐步升始对

中文文本分类进行研究[16]。2001 年，周水耕[19]提出了将隐含语义索引应用于中文文本处理中。2005 年，朱靖波[20]等人提出一种新的以领域知识为基础的文本分类方法。2009 年黄秀丽[21]等人提出 SIG（Square Root IG）方法，强调低频特征在分类过程中的作用。张玉芳[22]等人于 2013 年提出了综合特征的文档频率、平均词频等四项因素在内的新的特征选择方法 CR。

我国围绕 Word2vec 的研究和优化工作一直在进行，在产业界，也有公司和研究团体正在对 Word2vec 进行研究和应用，例如奇虎 360 公司[19]。目前，国内对 Word2vec 的研究并不多，主要集中在高等院校和研究机构的理论研究中，如 2014 年周练[9]、熊福林[23]等人对其原理进行了探究和介绍，并对 Word2vec 在中文中的应用进行介绍。2015 年江大鹏[37]等人对词向量在短文本分类中的应用进行探究，在 Word2vec 的基础上提出了加权连续词袋模型,同时也研究了主题模型 LDA，并提出基于主题分布的词向量生成算法。2016 年王明亚[24]等人将 Word2vec 训练得到的词向量应用到传统的特征选择中，研究了词向量之间存在的相似性关联，对特征词进行了适当的扩充，以弥补 " 特征词不完备 " 这一不足，同年 S Lai[39]等人对如何生成高质量的词向量进行了探究，认为模型、语料、参数三方面会影响词向量的训练。

1.3 本文主要工作和创新

本文对目前文本分类涉及到的中文分词、文本表示模型、词向量模型、语言模型、特征选择算法、常用分类算法等技术做了介绍，认为现有分类算法可以进行改进的地方：1、算法仅对训练语料中词的出现与否、词的频率等信息做统计，忽略了单词所处的上下文环境，忽略了文字之间的结构信息，不能充分利用训练文本中的语义信息；2、仅仅使用特征选择等方法将关键词提取出来，但并未将关键词对待分类文本的重要程度引入算法当中。本文对谷歌发布的词向量训练工具 Word2vec 做了深入的研究与实验。针对现有方法对文本分类中文本特征表示的语义信息利用不充分这一问题，提出一个新的解决方法，本文主要工作总结如下：

（1）本文通过研究文本分类相关技术，结合对语言模型的研究，认为现有语言模型训练得到的分布式词向量能够有效表达语料中词的语义信息，本文基于分布式词向量和 K 近邻算法思想，提出新的文本分类方法——KKB-DC，该方法能够充分利用由语料训练所得词向量模型中词的语义信息；

（2）针对算法在关键词个数选取较多时分类准确率下降较快这一问题，提出对 KKB-DC 算法的改进，引入待分类文本关键词的 tf-idf 值进行权重优化。实验证明在算法的权值优化中引入关键词的 tf-idf 值，可以有效改善算法性能；

1.4 本文组织结构

本文的组织结构如下：

第 1 章绪论介绍文本分类技术的背景，现阶段文本数据量的迅速增加对文本分类技术提出了更高的要求。同时研究了国内外对于文本分类的研究及相关成果，指出其中可能存在改进的地方；

第 2 章详细介绍本文所涉及到的相关技术，如文本分类的概念、中文分词、文本表示模型、特征选择方法、常用的分类算法，并对词向量模型和语言模型的原理做了介绍；

第 3 章介绍了谷歌开源的词向量生成工具——word2vec，并对其原理进行介绍，介绍词与词之间语义相似度的方法，举例说明词与词之间语义相似度的衡量。

第 4 章描述了本实验的数据准备工作，并针对现有分类器对词的语义信息利用不足这一点进行算法设计与优化，提出一个新的基于词向量和 K 近邻算法思想的文本分类算法,简称:KKB-DC；

第 5 章设计实验探究可能会对本文提出算法 KKB-DC 的可能影响因素，分析本文所提分类算法的实验结果；

第 6 章针对本文所提算法的不足进行分析和改进，在分类算法中引入待分类文本的 tf-idf 进行权重优化，通过实验证明权值优化是可行的。此外，本章做了与现有分类模型的对比实验，实验证明本文算法的有效性；

第 7 章对本文做整体总结，阐述本文所做的贡献与未来可能需要改进的地方。

第 2 章 相关技术综述

本章将介绍文本分类中用到的相关技术，包括文本预处理、文本表达、中文分词和特征选择方法等。通过这些内容，可以更好的帮助理解文本分类常用的技术与方法，也为后续章节的描述打下基础。

2.1 文本分类的概念

文本分类(Text Classification)是指将文本按照内容的不同判别到一个或多个预先确定的类别之中的过程, 文本文类是一种有指导的映射过程，也称做监督学习，整个过程中，需要计算机通过已经标注好的数据，学习文本特征和类别之间的关系模型，然后以此模型来预测新的文本所属的类别。给定一组事先定义好的文本类别集合 $C=\{c_1,c_2,c_3 \ldots c_n\}$和一组待分类文本集合 $D = \{d_1,d_2,d_3 \ldots d_m\}$，其中 n、m 分别表示文本类别数量和待分类文本数量。

文本分类的过程就是找到一个文本分类模型 f，f 本质上是从文本集合到文本类别集合的一个映射 f：D → C，如图 2.1 所示：

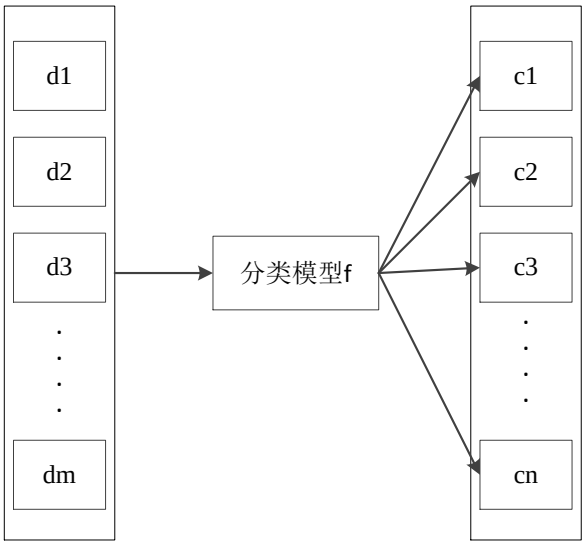

图 2.1 文本集合到文本类别的映射

Fig. 2.1 Mapping from text collection to text category

2.2 中文分词

随着中文信息处理技术的发展和硬件水平的提升，中文信息处理效率得到明显提高。中文信息处理[25]分为汉字信息处理与汉语信息处理两部分，具体内容包括对字、词、句、篇章的输入、存储、传输、输出、识别、转换、压缩、检索、分析、理解和生成等方面的处理技术。其中，中文分词是其中最基础并且也是最重要的技术之一。

中文分词，又叫中文切词（Chinese Word Segmentation）[26]，指的是将一个汉字序列切分成一个一个单独的词。例如：使用中文分词系统对一个中文语句进行分词：“特朗普成功当选美国总统。”，应得到“特朗普 | 成功 | 当选 | 美国总统。”。

中文分词的研究开始于二十世纪 80 年代，发展至今大致分为三个阶段：第一阶段从 80 年代到 90 年代，这一阶段以基于词典的算法为主；第二阶段从 90 年代到 2003 年，这一阶段以基于统计的分词方法为主；第三阶段从 2003 年至今，中文切词从基于词的方法向基于字方法过渡，最近几年主要研究主要集中在对字在词中位置的标注上。

中文分词一直是自然语言处理的热点与难点。文献[27][28]提出了一个基于字分类器和词分类器的简单而有效的半监督学习算法；Wang L Z H 等[29]人针对中文分词监督学习模型在训练过程中需要大量注释数据而提出了一个可扩展的半监督方法；Sun X 等[30]针对高维特征训练速度较慢这一缺点，提出了基于特征频率信息的网络梯度下降方法，加快了训练速度和提高训练精度；Qian X 等[31]使用一种新的解码算法判别中文分词的效果；文献[60][32]从深度学习、主动学习出发，分别提出深度神经网络模型和词边界注释模型应用到中文分词问题。

目前中文分词技术已取得了很多成果，出现了一大批实用的、可靠的中文分词系统。其代表有：基于 lucene 为应用主体开发的 IKAnalyzer 中文分词系统；庖丁中文分词系统；纯 C 语言开发的简易中文分词系统 SCWS；中国科学院计算技术研究所张华平博士推出的汉语词法分析系统 ICTCLAS，现在为 NLPIR 汉语分词系统；哈尔滨工业大学信息检索研究室研制的 ICTCLAS；另外国内的北大语言研究所、清华大学、北京师范大学等机构也推出了相应的分词系统。

从上述分析可以看出，中文分词依然存在很多难点，一方面是中文不是向英文那样简单的以空格作为划分，另一方面是中文语义的复杂性。本文实验中采用的中文分词工具是目前在中文分词领域表现较好，由张华平博士团队研发的 NLPIR[33]中文分词工具包。

2.3 文本表示

人类阅读文章之后，可以凭借自身经验对文章内容产生整体上的理解。与人类不同的是，计算机无法像人类一样直接识别文章，从本质上来说，计算机只能识别 0 和 1，所以必须将文本转换成计算机能够识别的格式。文本是大量字符的集合，一般为非结构化或者半结构化信息，不能直接被任何分类器识别，必须将其转换成一个统一的能够被计算机和分类器识别的形式，才能进行进一步的处理。将文本从非结构化向结构化转换的过程就是文本表示。

文本表示首先是从文本中提取能表示该文本的特征。文本的特征应该不仅能够对文本进行充分表示，而且要反映出文本在特征空间中的分布，具有较为明显的统计规律，还要尽量减小文本映射到特征空间的计算复杂度。

常用的文本特征有字、词或短语等。在实际应用中，选择何种粒度的特征来表示文本需要结合处理速度、存储空间等方面的具体要求来决定。常见的文本表示模型有：布尔模型、概率模型和向量空间模型[34]。

(1) 布尔模型

布尔模型（Boolean Model）是简单而又常用的严格匹配检索模型[35]。基本思想是以关键词是否出现来表示文档内容。该模型定义一个二值变量集合来表示文档，这些变量对应于文档中的词或者短语，也称特征项。如果某一特征项在文档中出现时，该特征项所对应的的变量值就为 1，否则为 0。布尔模型的表达形式如下：

$$\mathrm{di} = (w_{i1}, w_{i2}, \dots, w_{ik}, \dots, w_{in})$$

其中，n 为特征项个数，w_{ik}为 0 或 1，表示第 k 个特征项是否在d_i个文档中出现。

布尔模型优点是：速度快，很容易表达结构化信息。它的缺点是：仅仅保留词是否出现的信息，不能反映某一特征项对于文本的重要性，缺乏灵活性。

(2) 向量空间模型

向量空间模型（Vector Space Model，简称 VSM）[5][36]是 Salton 等于 1975 年提出。其基本思想是：一份文档所属类别仅与某些特定的词或者词组在该文档中出现的频率有关，与词语出现的位置或顺序无关。向量空间模型是一种非常经典的文本表达方式，因其具有良好的计算行和可操作性，在信息领域得到了广泛的应用[24]。文档通过模型表示之后，文档之间的相似性便可以通过计算向量之间的夹角或者欧氏距离等大小进行衡量，计算式通常采用向量夹角的余弦值[37]。

在向量空间模型中文本被表示为 n 维空间中的向量：

$$\mathrm{D} = (W_1, W_2, \dots, W_i, \dots, W_n)$$

其中，W_i表示第 i 个特征的权重。

向量空间模型将每篇文档表示成实数类型的向量，如果每个分量对应一个词语，分量的值通常采用 TF-IDF（Term Frequency, Inverse Document Frequency）权重。TF-IDF 将在 2.6.3 节中介绍。向量空间模型缺点是虽然给计算带来了方便，但是忽略了特征项之间的顺序，损失了文本中的语义信息。

(3) LDA(Latent Dirichlet Allocation)主题模型

LDA(Latent Dirichlet Allocation)主题模型。是由 Blei 等[38]人于 2003 年提出的。它的原型是概率隐形语义索引(probabilistic Latent Semantic Indexing, pLSI)，在原有的基础上经过扩展得到的三层贝叶斯概率模型。LDA 模型的结构分为三层，分别为文档、主题、词项。它的主要思想是将文档看做多个隐性主题集合上的概率分布，同时将每个主题看做相关词项集合上的概率分布。LDA 模型可以用来识别大规模文档集或者语料库中潜在的主题信息[39]。LDA 模型是基于词袋模型的，认为文档和单词都是可交换的，忽略单词在文档中的顺序和文档在语料库中的顺序，从而将文本信息转化成计算机可识别的数字信息。

2.4 词向量模型

2.4.1 One-hot Representation

在自然语言处理中，最常见的一步是创建一个词库表并把每个词顺序编号。这实际就是词表示方法中的 One-hot Representation，这种方法把每个词顺序编号，每个词就是一个很长的向量，向量的维度为所选语料对应词库表的大小。其中只有对应位置上的元素值为 1，其余为零。在实际应用中，一般采用稀疏编码存储，采用此的编号。

这种表示方法一个最大的缺点是无法捕捉词与词之间的相似度，就算是近义词也无法从词向量中看出任何关系，如：“西红柿”表示为：$[0, 0, 0, 0, 1, \ldots\ldots 0, 0, 0]^T$，“番茄”表示为：$[0, 1, 0, 0, \ldots\ldots 0, 0, 0]^T$，即使“西红柿”与“番茄”表示的是同一事物，但是从词向量无法看出两者的联系。此外这种表示方法还容易发生维数灾难，尤其是在深度学习相关的一些应用中。

2.4.2 Distributed Representation

Distributed Representation（分布式表示）最早是 Hinton 于 1986 年提出的[8]，可以克服 One-hot Representation 的向量维数大缺点，词的分布式表示使用维度更低的稠密向量，K 的值可以人为指定，以几十或者几百维为常见，一般远小于词库表的大小。其基本思想是通过训练将某种语言中的每一个词映射成一个固定长度的词向量，将所有这些向量放在一起形成一个词向量空间。

分布式词向量不使用某一位的值代表指定词，而是使用整个词向量的数值的分布情况来表示该词在语料中所包含的信息。观察单个词对应的词向量是没有意义的，通过对不同词对应的词向量进行比较，才能得出词向量保留语料上下文信息的结论。下面从 word2vec 训练得的词向量模型中抽取部分词对应的词向量，举例如下：

“苹果”对应的词向量表示为：[0.30108 0.06369 …… -0.32933]；

“香蕉”对应的词向量表示为：[0.38464 -0.13620 …… -0.20238]；

“火车”对应的词向量表示为：[-0.01621 -0.49133 …… -0.11707]。

从上述几个示例中，可以看到“苹果”和“香蕉”分别对应的词向量，它们对应位置的数值相差不大，但是“苹果”和“火车”两个词对应的词向量，它们对应位置的数值相差就比较大。由此可以看出，类似“水果”属性的信息包含在“苹果”和“香蕉”所对应的词向量中，相对来讲，“火车”对应的词向量所表示“水果”这一属性信息就比较少。

通过上述示例的分析，每一个词向量为该空间中的一个点，在这个空间上引入“距离”的概念，可以通过“距离”来判断它们之间的（词法、语义上的）相似性。

不同的词之间的“距离”或者相似性可以使用词所对应的词向量之间的夹角来衡量，也可以使用词向量之间的余弦值。

考虑英语和西班牙语两种语言[40]，通过训练分别得到它们对应的词向量空间 E 和 S，如图 2.2 所示。从英语中取出五个词 one，two，three，four，five，设其在 E 中对应的词向量分别为 v_1, v_2, v_3, v_4, v_5，为方便作图，文献[40]利用主成分分析[41]（Principal Component

Analysis，PCA)降维(主成分分析是一种分析、简化数据集的技术,常用于减少数据集的维数,同时保持数据集中的对方差贡献最大的特征),降维后得到相应的二维向量u_1,u_2，u_3，u_4，u_5，在二维平面上将这五个点描出来，如图 2.2 中左图所示；类似地，在西班牙语中取出（与 one，two，three，four，five 对应的）uno，dos，tres，cuatro，cinco，设其在 S 中对应的词向量分别为 c_1，c_2，c_3，c_4，c_5，用 PCA 降维后的二维向量分别为 t_1，t_2，t_3，t_4，t_5，将它们在二维平面上描出来（可能还需作适当的旋转），如图 2.2 中右图所示：

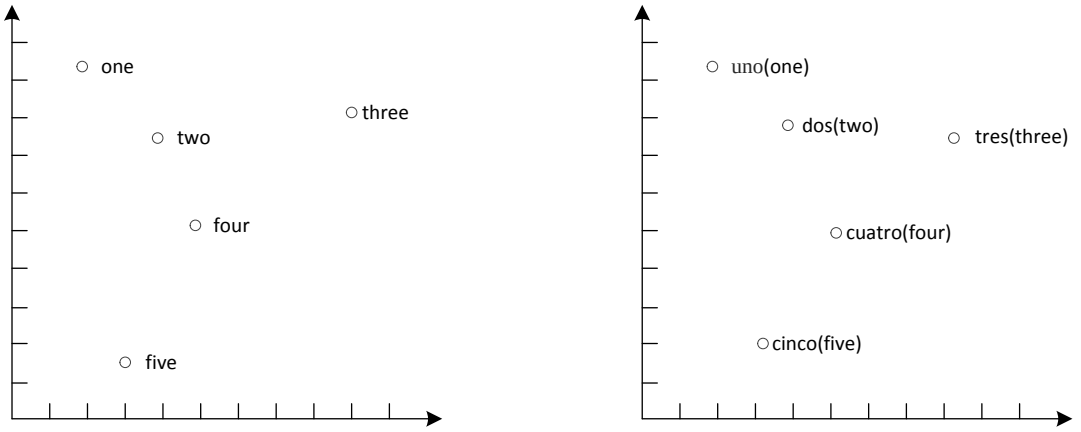

图 2.2 英语和西班牙语空间向量

Fig. 2.2 Space vector of English and Spanish

观察左、右两幅图，容易发现：五个词在两个向量空间中的相对位置差不多，这说明两种不同语言对应向量空间的结构之间具有相似性，从而进一步说明了在词向量空间中利用距离刻画词之间相似性的合理性。

受到分布式向量表示可以有效保留语料中语义信息的启发，结合文献[36]，本文利用词向量中所蕴含的训练语料上下文信息，提出一个基于词向量的文本分类算法，该算法引入训练语料中的上下文信息辅助分类决策。

本文中将使用 Word2vec 工具进行词向量模型的训练，所得词向量即本节描述的 Distributed Representation 表示，具体训练方法和过程将在 3.2 节详细描述。

2.5 语言模型介绍

2.5.1 统计语言模型

统计语言模型是用来刻画一个句子出现概率的模型[42][43]。给定一个由 n 个词顺序组成的一个句子，$S=(w_1,w_2,\dots,w_n)$，则该句子出现的概率$p(S)$即为统计语言模型。通过贝叶斯公式，可将$p(S)$分解为$p(S)=p(w_1)p(w_2|w_1)p(w_3|{w_1}^2)\dots p(w_n|{w_1}^{n-1})$。由此可见，要计算一个句子出现的概率，只需要在给定上下文语境的情况下，计算出下一个词为某一个词的概率即可，即$p(w_i|\text{context}(w_i))$，其中 context 为上下文。当一个句子中所有词的$p(w_i|\text{context}(w_i))$计算出来之后，连乘即可计算出$p(S)$。所以，统计语言模型的关键在于找到计算条件概率$p\left(w_i\middle|\text{context}(w_i)\right)$的模型。

其中 context 即为上下文，根据对 context 不同的划分方法，可以分为两类：

（1）上下文无关模型（context = NULL）

该模型仅仅考虑当前词本身的概率，不考虑该词所对应的上下文环境。这是一种最简单，易于实现，但没有多大实际应用价值的统计语言模型。

$$p(w_t|\text{Context}) = p(w_t) = Nw_t/N$$

这个模型不考虑任何上下文信息，仅仅依赖于训练文本中的词频统计。它是 n-gram 模型中当 n = 1 的特殊情形，所以有时也称作 Unigra Model（一元文法统计模型）。实际应用中，常被应用到一些商用语音识别系统中。

（2）n-gram 模型（context = $w_{t-n+1},w_{t-n+2},\cdots,w_{t-1}$）

n = 1 时，就是上面所说的上下文无关模型，这里 n-gram 一般认为是 n > = 2 是的上下文相关模型。当 n = 2 时，也称为 Bigram 语言模型，直观地想，在自然语言中“白色汽车”的概率比“白色飞翔”的概率要大很多，也就是 p(汽车|白色) > p(飞翔|白色)。n > 2 也类似，只是往前看 n-1 个词而不是一个词。一般 n-gram 模型优化的目标是最大 log 似然，即：

$$\prod T_t = 1p(w_t|w_{t-n+1},w_{t-n+2},\ldots,w_{t-1})\log pm(w_t|w_{t-n+1},w_{t-n+2},\ldots,w_{t-1})$$

n-gram 模型的优点包含了前 n-1 个词所能提供的全部信息，这些信息对当前词出现具有很强的约束力。同时因为只看 n-1 个词而不是所有词也使得模型的效率较高。

在计算复杂度方面，表 2.1 给出了 n-gram 模型中模型参数数量随 n 的逐渐增大而变化的情况，其中假设词典大小 n = 200000（汉语的词汇量大致是这个量级）

表 2.1 模型参数与 n 的关系

Table 2.1 Relationship between n and Model parameters

n	模型参数数量
1(unigram)	2*105
2(bigram)	4*1010
3(trigram)	8*1015
4(4-gram)	16*1020

在效果方面，理论上是 n 越大越好，今天互联网海量数据以及机器性能的提升使得计算更高阶的语言模型（n>10）成为可能，但是当 n 达到一定程度时，模型效果提升会越来越小。

但是 n-gram 存在一个问题，若训练语料里面有些 n 元组没有出现过，其对应的条件概率就是 0，这会导致计算一整句话的概率为 0。解决这个问题有两种常用方法：

一种是平滑法，最简单的是将出现 k 次的某个 n 元组看做出现了 k+1 次，这样出现 0 次的 n 元组就变成了出现 1 次。

另一种是回退法，即如果得不到 n 元的概率，有点像决策树中的后剪枝方法，那就用 n-1 元的概率乘上一个权重来模拟。

除了上面说的问题之外，n-gram 还存在其他问题：

1. n-gram 语言模型无法建模更远的关系，语料的不足使得无法训练更高阶的语言模型。

2.这种模型无法建模出词之间的相似度，有时候两个具有某种相似性的词，如果一个词经常出现在某段词之后，那么也许另一个词出现在这段词后面的概率也比较大, 比如：

The cat is walking in the bedroom

A dog was running in a room

如果第一句话里的元组在语料中出现的很多，训练的很充分，第二句话中的元组在语料中出现的少，训练的不充分，那么使用语言模型计算第一句话的概率就比较高，而第二句话的概率就低。

如果有一种方法，能知道 The 和 A 相似，cat 和 dog 相似等等，并且会给相似的词类似的语言模型概率，那么第二句话也可以得到高概率。

2.5.2 神经网络语言模型

随着深度学习的发展，神经网络相关研究越来越深入，神经网络语言模型（Neural Network Language Model，NNLM）越来越受到学术界和工业界的关注。NNLM 最早是由 Bengio 系统化地提出并进行了深入研究，其代表作为 2001 年发表的《A neural probabilistic language model》，并且于 2003 年发表同名论文[44]，对原论文进行了补充和丰富。Bengio 提出一个使用三层神经网络来构建语言模型，如图 2.3 所示：

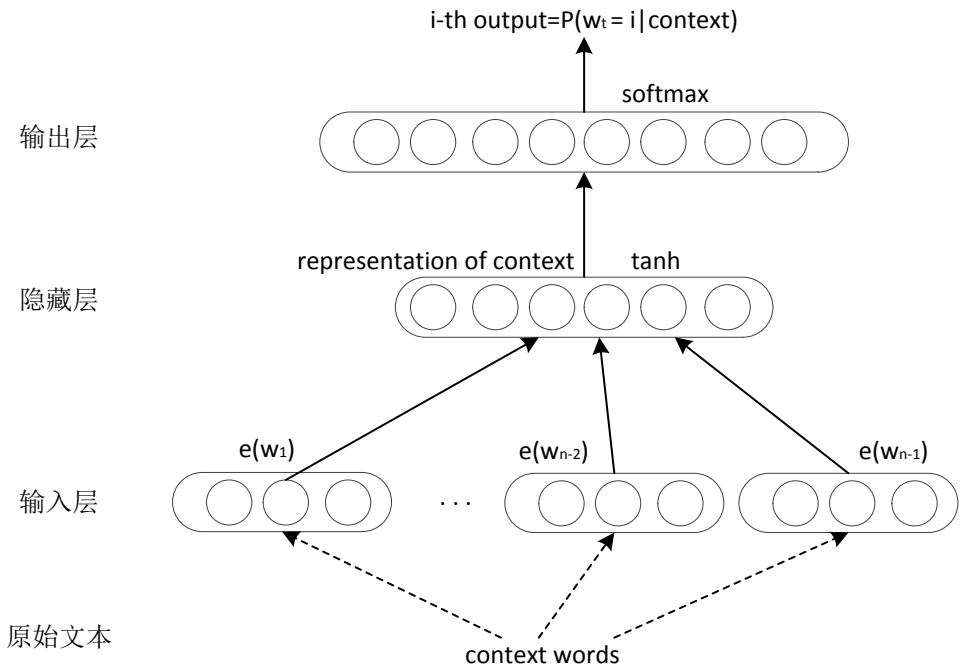

图 2.3 神经网络语言模型

Fig. 2.3 Neural network language model

该 NNLM 模型也属于 n-gram 模型，用已知的前 n-1 个词来预测下一个词为w_i。模型中，每个输入的词被映射成一个向量，NNLM 采用 Distributed Representation 来表示

词所对应的向量，保存在矩阵 e 中，其中 e(w), 表示单词 w 所对应的向量。e 是一个|V|*m 的矩阵（|V|表示词表大小，即语料中所有词的个数；m 为选取的词向量维数）。

如图 2.3，网络的第一层为输入层，是词向量映射，将所有输入单词的词向量拼接为一个矩阵作为网络的输入。第二层为隐藏层，使用 tanh 函数为激活函数，第三层为输出层，一共有|V|个叶节点，每个节点 y_i 表示下一个单词为 w_i 时未归一化 log 概率，最后使用 softmax 激活函数将输出值归一化成概率。y_i 计算公式为：

$$y = b + W x + U \tanh(d + H x) 。 \quad (2\text{-}1)$$

神经网络语言模型的输出与一般的统计语言模型类似，都是求下一个单词出现的概率。模型训练目标就是使合理序列的概率值最大，满足所有单词出现的概率值均为正数，并且所有的概率值和为 1，其数学描述为：

目标函数： $f(w_t,w_{t-1},...,w_{t-n+2},w_{t-n+1})=p(w_t|w_1^{t-1})$

约束条件： $f(w_t,w_{t-1},...,w_{t-n+2},w_{t-n+1})>0$

$$\sum_{i=1}^{|V|} f(i,w_{t-1},...,w_{t-n+2},w_{t-n+1})=1$$

通过随机梯度下降算法进行模型优化，与一般的神经网络模型不同，该模型的输入所使用的词向量也要优化。最终得到的词向量就可以作为最终的词向量使用，由此可见词向量是训练神经网络语言模型而得到的副产物。

词向量模型是具体的词与向量的集合，离线存储表现为一个向量文件。一般情况下向量的维数取值为几十到几百不等，需要根据实际情况来判断使用哪一数量级维度的向量模型更适合具体问题的解决，通常需要大量的实验进行查找。

2.6 特征选择方法介绍

在文本分类任务中，表示文本特征的数目并不是越多越好，如果利用一个特征对分类结果没有较大影响，则称这个特征是没有分类能力的[45]，经验上去掉此特征对分类结果影响不大。因此，在文本分类的任务中需要对文本进行降维，目前主要的方法是进行特征选择，即对当前大量的文本特征按照某种标准进行选择，仅挑选出能够代表文本内容、具有较好类别区分能力的特征。下对目前常用的文本特征选择算法[46]进行简要介绍：

2.6.1 互信息

互信息（Mutual Information）通过计算特征和类别共同出现的概率，来度量特征和类别的相关性[47][48]。特征 t 和类别 c_i 互信息值计算公式如公式（2-2）所示：

$$\mathrm{MI}(t,c) = log\frac{p(t,c_i)}{p(t)*p(c_i)} = log\frac{p(t|c_i)}{p(t)} \quad (2\text{-}2)$$

公式中，$p(t,c_i)$表示训练集中同时满足包含特征 t 又属于类别 c_i 的文本出现的概率。$p(t)$表示训练集中包含特征 t 的文本出现的概率，$p(c_i)$表示训练集中文本属于类别 c_i 的概率。当 t 与 c_i 完全独立时，$MI(t,c_i)$为 0；当特征 t 在类别 c_i 的文本中出现概率高，而在

其他类别中出现概率低，即特征 t 和类别 c_i 相关性大，将获得较高的互信息值 MI(t,c_i)，反之将获得较低的互信息值 MI(t,c_i)。使用互信息进行特征选择操作时，常采用最大互信息或平均互信息。

2.6.2 信息增益

信息增益（Information Gain，简称 IG），一种基于信息熵的方法[45]。为便于说明，先给出熵与条件熵的定义。

在信息论与概率统计中，熵是表示随机变量不确定性的度量，设 X 是一个有限取值的变量，其概率分布为公式（2-3）：

$$\mathrm{P}(\mathrm{X}=x_i)=p_i \quad i=1,2,3,\dots,n \tag{2-3}$$

则随机变量 X 的熵定义为公式（2-4）：

$$\mathrm{H}(\mathrm{X})=-\sum_{i=1}^{n} p_i \log p_i \quad i=1,2,3,\dots,n \tag{2-4}$$

熵越大，随机变量的不确定性就越大。

设有随机变量(X,Y)，联合概率分布为公式（2-5）：

$$\mathrm{P}(\mathrm{X}=x_i,\mathrm{Y}=y_j)=p_{ij} \quad i=1,2,3,\dots,n;\ j=1,2,3,\dots,n \tag{2-5}$$

条件熵 H(Y|X)表示在已知随机变量 X 的条件下随机变量 Y 的不确定性。随机变量 X 给定的条件下随机变量 Y 的条件熵(Conditional Entropy) H(Y|X)，定义为 X 给定条件下 Y 的条件概率分布的熵对 X 的数学期望[45]如公式（2-6）所示：

$$\mathrm{H}(\mathrm{Y}|\mathrm{X})=\sum_{i=1}^{\mathrm{n}} p_i \, H(Y|X=X_I) \tag{2-6}$$

特征 t 对数据集 D 的信息增益 g(D,t)，定义为集合 D 的熵 H(D)与特征 t 给定条件下的条件熵 H(D|t)之差，即公式（2-7）：

$$\mathrm{g}(\mathrm{D}|\mathrm{t})=H(D)-H(D|t) \tag{2-7}$$

根据信息增益进行特征选择的方法是：计算每个特征的信息增益，比较它们的大小，选择信息增益最大的特征。

2.6.3 文档频率与逆文档频率

文档频率与逆文档频率（Term Frequency–Inverse Document Frequency 简称：TF-IDF）是一种用于资讯检索与文本挖掘的常用加权技术[49]。TF-IDF 是一种统计方法，用以评估一个词对于一个文件集或一个语料库中的其中一份文件的重要程度。字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。

词频（Term Frequency，简称 TF）指的是某个给定的词在文档中出现的频率。对于一个特定的词t_i来说，它在第 j 个文档d_j中的 tf 值计算方法为公式(2-8)：

$$tf_{i,j}=\frac{n_{i,j}}{\sum_k n_{k,j}} \tag{2-8}$$

其中$n_{i,j}$是该词在文件d_j中的出现次数，而分母则是在文件d_j中所有字词的出现次数之和。为防止 tf 值偏向长文，通常使用的是对文档总词数归一化后的结果，取值范围为（0,1）。

语料库中包含某个词 w 的文档的数量称为该词的文档频率（Document Frequency，简称 DF）[50]。如果统计词语在整个语料库的文档频率，通过过滤文档频率低于一定阈值的词语可以在一定程度上去除无效词语，但是对于一些常用词，比如“我们”、“但是”等使用频率非常高的但实际包含信息量不大的词，仍会被保留下来。通常是由另一个方法来衡量——逆文档频率（Inverse Document Frequency，简称 IDF）。IDF 的计算方法如公式(2-9)：

$$\mathrm{idf(w)} = log\frac{|D|}{|\{j:w\in d_j\}|+1} \tag{2-9}$$

其中$|D|$代表文档总数，$\left|\{j:w\in d_j\}\right|$表示包含 w 的文档总数，分母加 1 是为了防止分母为 0 的情况。IDF 用来衡量词语在整个语料中的重要性，IDF 与 TF 结合起来的 TF-IDF 则可以用来衡量某个词语对某篇文档的重要性，具体计算方法如公式(2-10)：

$$tfidf_{i,j} = tf_{i,j} * idf_i \tag{2-10}$$

近年来不少人将 TF-IDF 用于特征的权值计算，并取得不错的效果。因此，本文考虑使用 TF-IDF 技术进行关键词提取和权值优化。

2.7 常用分类方法介绍

2.7.1 朴素贝叶斯分类算法

朴素贝叶斯（Naive Bayes classifier，简称 NBC）[51]，在贝叶斯学习方法中最具实用性的一种，是基于贝叶斯定理与特征条件独立假设的分类方法。假定一个属性值对给定类的影响独立于其他属性值，通过计算待分类项出现时各类别出现的概率，其属于概率最大的类。因其有着坚实的理论基础和稳定的分类效率而被广泛使用。

基本方法：设输入空间为 n 维向量的集合，输出空间为类标记集合$\{c_1,c_2,c_3,\ldots,c_k\}$。X 是定义在输入空间上的随机变量，Y 是定义在输出空间上的随机变量，P(X,Y)是 X 和 Y 的联合概率分布。训练集为：

$$\mathrm{T} = \{(x_1),(x_2),\ldots,(x_N,y_N)\}$$

通过训练数据集学习联合概率分布 P(X,Y)。具体地，学习以下先验概率分布和条件概率分布。

先验概率分布：

$$\mathrm{P(Y} = y_i),\mathrm{i} = 1,2,\ldots,\mathrm{k}$$

条件概率分布：

$$\mathrm{P(X} = \mathrm{x|Y} = y_i) = \mathrm{P}\left(X^{(1)} = x^{(1)},\ldots,X^{(n)} = x^{(n)}\middle|\mathrm{Y} = y_i\right),\mathrm{i} = 1,2,\ldots,\mathrm{k}$$

朴素贝叶斯法对条件概率分布作了条件独立性假设：每个特征词之间都是相互独立互不影响的，因此有：

$$\mathrm{P}(\mathrm{X}=\mathrm{x}|\mathrm{Y}=y_i)=\mathrm{P}\left(X^{(1)}=x^{(1)},\ldots,X^{(n)}=x^{(n)}\middle|\mathrm{Y}=y_i\right)$$

$$=\prod_{j=1}^{n}P(X^{(j)}=x^{(j)}|Y=\ y_i)\,,\mathrm{i}=1,2,\ldots,\mathrm{k}$$

根据贝叶斯定理可得，文本类别y_i的后验概率$P(y_i|x)$如公式所示：

$$P(y_i|x)=\frac{P(x|y_i)P(y_i)}{P(x)}$$

由于公式中分母表示输入 x 在整个训练集中出现的概率值，这个值对于各个类别都是常数，因此可以舍去，与公式结合可得：

$$P(y_i|x)=P(y_i)\prod_{j=1}^{n}P(x_j|y_i)$$

得到每个类别对于文档 x 的概率，然后找出使$P(y_i|x)$最高的类别y_i，将文档 x 分类给该类别。

朴素贝叶斯分类算法实验开销相对较小，分类效果也比较好。但是其缺点也是很明显的，算法对各个特征项作相互独立假设，这是不合理的，尤其会使文档语义信息的保留受损。

2.7.2 支持向量机

支持向量机[52]（Support Vector Machine，简称 SVM），过寻找最优分隔超平面将不同类别中的样本分开。最初是为了解决二分类问题被提出，现在被广泛应用于解决多维线性分类问题。其基本思想是寻找最优的分类超平面，以使得分类间隔最大化。举例如二维平面中两个类别的划分，如下示意图 2.4 所示：

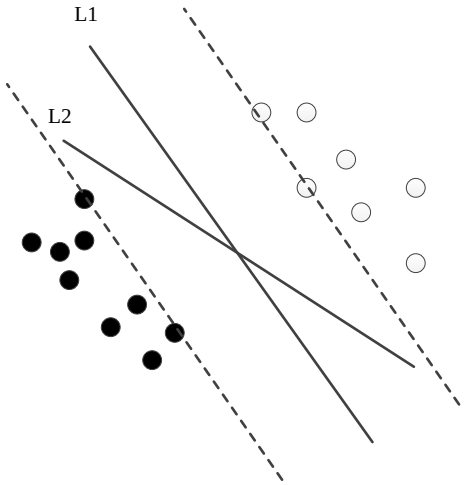

图 2.4 最优超平面示意图

Fig. 2.4 Optimal hyperplane diagram

图 2.4 中直线 L1 比直线 L2 分类效果要好，因为 L1 对两类数据进行最大程度的分离。在 SVM 中，线性不可分[45]可以通过引入核函数来解决，核函数本质上是一个映射模型，可以把向量所在的空间 X 映射到另一个空间 Y 中，Y 可以是有限维空间，也可

以是无限维空间。常用的核函数有线性核、多项式核、高斯核与 S 型核，这使得 SVM 可以对非线性问题进行求解。

直线 L1 推广到一般的 n 维线性可分的分类中，为 n-1 为超平面，如公式（2-11）：

$$w^T x+b=0 \qquad (2\text{-}11)$$

其中，x 为样本的特征。相应的分类决策函数为：

$$f(x)=sign(w^T x+b)$$

寻找最优分类超平面的过程也就是确定超平面公式中的参数 w 与 b 的过程。一个点距离分类超平面的远近可以表示分类预测的确信程度，在超平面确定的情况下，$|w^T x+b|$能够相对的表示点 x 与超平面的距离远近，$w^T x+b$的符号与类标记 y 的符号是否相等可以判断分类是否正确，可以使用 $y(w^T x+b)$表示分类的正确性及确信度，称之为函数间隔。

超平面关于样本点（x_i, y_i）的函数间隔为：

$$d_i = y_i（w^T x_i + b）$$

但是仅仅使用函数距离是不够的，从公式（2-11）可以看出，成比例地改变 w 和 b，分类超平面没有改变，但是函数间隔却按相应的比例变化。因此引入几何间隔的概念：给定数据点 x_i，其类别标记 y_i(正类为 1，负类为-1)和超平面 $f(x):w^T x+b=0$，则点到超平面的几何间隔 d_i 为：

$$d_i = y_i \frac{w^T x+b}{||w||}$$

定义超平面关于训练数据集的几何间隔为超平面关于数据集中所有样本点的几何间隔的最小值，如公式：

$$d = \min_{i=1,\dots,k} d_i$$

这样，SVM 的最优目标就变成：

$$\max_{w,b} d$$

$$\text{s.t. } y_i \frac{w^T x+b}{||w||} \geq d_i, i = 1,2,\dots,k$$

其中，k 为数据集中样本的个数。

如今 SVM 算法已经被广泛应用于人脸识别、手写体识别、文本分类等众多领域，并且经实践验证取得了不错的分类效果，本文文本分类对比实验采用的是支持向量机算法。

2.7.3 K 近邻分类算法

K 近邻法（k-nearest neighbor，K-NN）分类算法，是一种发展成熟且易于理解的机器学习算法。由 Cover 和 Hart 于 1968 年提出[46]。这是一种基于实例的算法，其基本思想是：在一个样本空间中，如果计算出一个样本与 K 个样本最相似，并且 K 个样本中

的大多数属于某一类别，K 近邻算法则将该文本判别为这一类别中。在这个算法中，得到的最相似的文本都已正确分类。

K 近邻算法没有训练阶段，它的分类要借助整个训练集，而且训练文档越多，分类计算所需的时间也将线性增加。其中 K 值的选择、距离的度量以及分类规则是影响分类效果的三个重要因素：

（1）K 值的选择：K 值的选择会对算法产生较大的影响。如果 K 值过小，则容易发生过拟合；如果 K 值过大，会增加计算成本，而且学习的误差也会增大。在实际应用中，K 值一般不宜过大，可采用交叉验证选取最有效的 K 值。

（2）距离度量：距离度量指的是待分类样本与其他样本的相似度的一种方法，需要注意的是度量之前需要将样本各个属性值进行规范化处理，防止具有较大值的属性对具有较小值的属性影响过大。计算文本之间相似度的方法有很多，最常采用的是两个向量余弦值的方法，对于给定两个文档的向量，它们之间的余弦值公式如公式（2-12）：

$$\mathrm{sim}(d_i, d_j) = \frac{\sum_{k=1}^{n}(w_{ik} * w_{jk})}{\sqrt{(\sum_{k=1}^{n} {w_{ik}}^2 * \sum_{k=1}^{n} {w_{jk}}^2)}} \qquad (2\text{-}12)$$

上式中，w_{ik}表示第 i 个文档的第 k 个属性，$\mathrm{sim}(d_i, d_j)$的值越大，则表示两文本的相似度越高。

（3）分类规则：分类规则往往使用多数表决法，即将距离待分类样本最近的 K 个样本中的多数样本所在的类作为待分类样本的类别。

K 近邻算法的主要缺点是，需要计算待分类文本与训练集中每一个样本的相似度，计算时间的开销比较大；K 值需要明确指定，K 值选择不当同样会影响分类精度。

本文算法需要对待分类文本的若干关键词进行类别判断，借鉴 K 近邻算法思想可以很好的对关键词分类。具体做法是：在关键词的类别判断中，对某一关键词，要找出其所属的类，只需使用某种方法度量出关键词与各个类之间的距离关系，取距离最小的这一分类为关键词的类别，即 K 近邻算法中 K 等于 1 的情况。在后面的 3.3 节中本文提到，使用的度量方法是两个词在词向量模型中所对应词向量之间的余弦值，余弦值越大则两个词距离越近。

2.8 本章小结

本章主要对文本分类中的相关技术与理论做了介绍。首先描述文本分类的概念，给出文本分类的定义；接着介绍中文分词相关技术背景与综述；然后对文本表示模型和词的向量表示的两种常用模型做出介绍，同时，简要介绍常用的语言模型；最后对常用的特征选择方法做了基本的说明、对常用的分类算法做了介绍。

第 3 章 词的语义相似度

3.1 概述

通过对文本分类相关技术和语言模型的介绍，结合本文第 1 章中指出的现有文本分类中语料语义信息利用不充分这一现象，我们可以使用神经网络语言模型来充分训练语料，使其训练所得的词向量中蕴含训练语料的上下文信息，然后将词向量应用到文本分类当中。因此，本章对语言模型的训练和词与词之间语义相似度计算进行介绍。

3.2 词向量训练工具

3.2.1 Word2vec 介绍

Word2vec[53]是 2013 年 Tomas Mikolov[16][17][18]在 Google 带领研究团队所提出的神经网络语言模型的一个实现，可以用来快速训练分布式词向量。Word2vec 得到了广泛关注，它一般被看做是一个深度学习（deep learning）模型，是深度学习在自然语言处理领域中突破性应用，但因该模型实际只有三层，严格来说不能算作是深层模型。

Word2vec 的输入是已经经过分词的文本语料，对于英文不要专门的分词处理，而对于中文语料则需要进行分词操作，本实验使用张华平博士研发的 NLPIR 中文分词系统对语料进行分词处理。Word2vec 的训练输出是一张词汇表，其中每个词都有对应一个指定维度的向量，即本文 2.4.2 节描述的 Distributed Representation 分布式词向量，维度一般为几十到几百维[39]。

Word2vec 的目的和功用是在向量空间内将词的向量按相似性进行分组。它能够识别出词在语料上下文环境中的相似性，继而生成向量，以分布式的数值形式来表示词的上下文等特征。而这一过程无需人工干预。

3.2.2 词向量训练模型

Word2vec 内含两种训练模型，分别是连续词袋模型（Continuous Bag-of-Words Model，简称 CBOW）和 Skip-gram[54]模型，如图 3.1 所示：

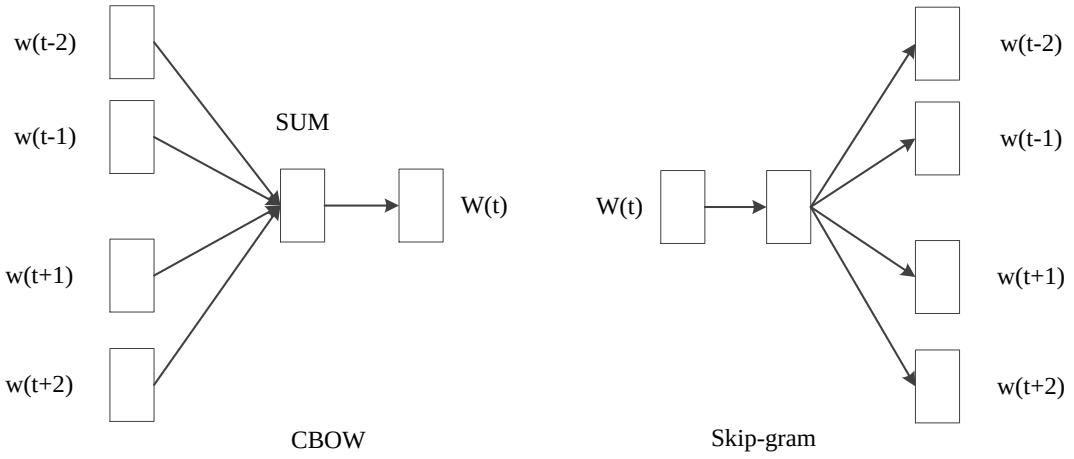

图 3.1 CBOW 和 Skip-gram 模型

Fig. 3.1 Model of CBOW and Skip-gram

Word2vec 中提供两种优化方法来提高训练效率，分别是 Hierachy Softmax 和 Negative Sampling。将训练模型和优化方法进行组合可得到 4 种训练词向量的框架[9]，如表 3.1 所示：由于本文只应用了 Hierarchical Softmax 这种方法，因此第二种方法我们将不做解释。

表 3.1 Word2vec 词向量训练框架

Table 3.1 Framework of Distributed representation training in Word2vec

模型	CBOW	Skip-gram
Hierachy Softmax	CBOW+HS	Skip_gram+HS
Negative Sampling	CBOW+NS	Skip_gram+NS

（1） CBOW 模型

CBOW 模型是一种与前向神经网络类似的模型，不同之处在于 CBOW 去掉了最耗时的非线性隐层并且所有单词的共享隐层，CBOW 模式通过上下文来预测当前词, 模型只包含输入层、投影层和输出层。下面我们将对结构图中的三层做简要说明：

输入层：与传统的神经网络语言模型一样，输入层是上下文单词的词向量

投影层：与传统的神经网络语言模型不同的是，投影层不再是对输入层进行拼接，而是对输入层中输入的 n 个向量做简单的求和累加；

输出层：模型的输出层对应为一棵二叉树，二叉树的叶子结点是语料中出现的所有单词。该二叉树是一棵通过将每个单词在语料库中出现的次数当作权值来构建的 Huffman 树，树的叶子结点个数为|V|，非叶子结点的个数为|V|-1。

对于处在叶子结点的每个单词而言，要想经过二叉树达到单词结点，必须经过一系列的二分过程。则条件概率的数学描述为

$$P(w|context(w)) = \prod_{i=2}^{l} p(d_j^w|x_w, \theta_{i-1}^w)$$

其中，1 表示到达该叶子结点所要经历的结点个数, d_j^w表示到达单词 w 中的第 j 个结点对应的 Huffman 编码, θ_{i-1}^w表示到达单词 w 所需要经历的第 i 个中间结点的向量。Huffman 树中每个结点对应的二分概率我们利用 sigmoid 函数进行判别，其数学描述为：

$$\mathrm{p}\left(d_j^w|x_w, \theta_{i-1}^w\right) = \begin{cases} \sigma(x_w^T\theta_{i-1}^w), & d_j^w = 0 \\ 1-\sigma(x_w^T\theta_{i-1}^w), & d_j^w = 1 \end{cases}$$

以上式为目标函数，与传统的神经网络语言模型相同，利用随机梯度下降法进行参数优化，最终得到训练好的词向量。

（2） Skip-gram 模型

如图 3.1 所示，Skip-gram 模型的结构与 CBOW 模型刚好相反，使用当前词来预测其上下文。Skip-gram 模型同样也包括三层：输入层，投影层和输出层。不同的是 Skip-gram 是已知当前单词 w，预测上下文 context(w)中的单词。与 CBOW 模型相比较，输入层变成一个单词的词向量，而输出层将是该单词所包括的上下文单词的 Huffman 树。

因此，其目标函数表示为：

$$P(w|context(w)) = \prod_{u \in context(u)} p(u|w)$$

其中：

$$\mathrm{p(u|w)} = \prod_{i=2}^{l} p(d_j^u | x_w, \theta_{i-1}^u)$$

其计算过程与 CBOW 模型类似，将不再赘述。

3.2.3 词向量模型训练

本节介绍如何训练并使用 Word2vec 训练词向量模型。

DeepLearning4J（DL4J）[55]是一套基于 Java 语言的神经网络工具包，可以构建、定型和部署神经网络。扩展使用 Deeplearning4j 中的代码，配置参数，即可训练生成词向量模型。配置训练模型的核心代码和参数解释表 3.2、表 3.3：

表 3.2 配置模型代码

Table 3.2 Model configuration code

```
(1)  log.info("Building model....");
(2)  Word2Vec vec = new Word2Vec.Builder()
(3)      .minWordFrequency(5)
(4)      .iterations(1)
(5)      .layerSize(200)
(6)      .seed(42)
(7)      .windowSize(5)
(8)      .iterate(iter)
(9)      .tokenizerFactory(t)
(10)     .build();
(11) log.info("Fitting Word2Vec model....");
(12) vec.fit();
```

表 3.3 参数解释

Table 3.3 Parameter Explanation

batchSize	每次处理的词的数量
minWordFrequency	一个词在语料中必须出现的最少次数
layerSize	指定词向量中的特征数量，与特征空间的维度数量相等。以 500 个特征值表示的词会成为一个 500 维空间中的点
iterations	网络在处理一批数据时允许更新系数的次数
learningRate	每一次更新系数并调整词在特征空间中的位置时的步幅
minLearningRate	学习速率的下限。学习速率会随着定型词数的减少而衰减。如果学习速率下降过多，网络学习将会缺乏效率。
vec.fit()	让已配置好的网络开始定型

使用为更好的评估 Word2vec 模型，使用的词向量模型由搜狗实验室提供的搜狐新闻数据（SogouCS）和全网新闻数据（SogouCA）[57]训练所得。由于所得模型较大，在加载的时候需要手动设置 JVM 的堆栈大小。

训练所得词向量模型为一张由词和其对应的浮点型向量组成的表，表的大小为满足出现最小频率词数的词的数量。本实验中一共得到 134535 个词的词向量模型，其中每个词所对应的向量均为 200 维的浮点型数组。

3.3 词的相似度评估

本节将通过实验证明词向量模型能够很好的保留训练集中文本的上下文信息，即通过词分布式向量表示保留训练语料的语义信息，训练出的模型使具有相同上下文环境的词具有相似的词向量表示。具体表现为，如果两个词的上下文环境越相似，那么它们所对应的词向量之间的余弦值越接近于 1，反之，则所得余弦值越接近于 0。

3.3.1 相近词的查找

使用 3.2.2 节训练所得的词向量模型，如找到与词“法律”最相近的 10 个词语，并输出它们与“法律”之间的余弦值，可用表 3.4 中的代码：

表 3.4 示例代码

Table 3.4 Sample code

```
(1) Word2Vec word2Vec = WordVectorSerializer.readWord2VecModel("/GoogleNews-vectors-
    negative300.bin.gz");
(2) Collection<String> lst = word2Vec.wordsNearest("法律", 10);
(3) for (String string : lst) {
(4) System.out.println(string+"\t"+word2Vec.similarity(string, "法律"));
(5) }
```

表 3.5 是加载该词向量模型所得与“法律”在向量空间中最近的词语列表及相应的余弦值：

表 3.5 与“法律”相近的单词排列

Table 3.5 List of nearest words with“law”

编号	单词	与单词“法律”之间的余弦值
1	法规	0.7666702270507812
2	规章	0.7154861092567444
3	条文	0.6712163686752319
4	合法	0.6702567934989929
5	相对人	0.66758131980896
6	违反	0.6529091596603394
7	规范性	0.6447259187698364
8	强制力	0.6442285776138306
9	规定	0.6384625434875488
10	诉讼法	0.6377309560775757
……	……	……

3.3.2 相似度评估

表 3.5 表明，在已经训练好的词向量模型所对应的向量空间中，如果一个单词与已知单词“法律”的余弦值越大，则表明该单词与单词“法律”在词向量空间中对应的向量之间的夹角越小，即它们在语料中有越相似的上下文环境，具有更近的语义关系。

因此，可以通过两个词所对应词向量的余弦值来衡量两个词之间的语义相似度，余弦值越大，说明两个词之间的语义越相近；反之，余弦值越小，说明两个词之间的语义越疏远，以表 3.5 中数据，考虑“法律”与“法规”、“合法”两个词之间的语义关系，如下：

词向量模型中，“法律”、“法规”、“合法”三个词所对应的向量分别为：

法规：[-0.7693 0.5577 -0.5800 0.1013 0.0889 -0.1865 …… -1.4988 0.1353 -0.0158]；

法律：[-0.4237 0.5204 0.1058 0.3376 -0.3306 -0.1479 …… -1.0358 0.8035 -0.0155]；

合法：[0.1155 0.2895 0.0575 -0.1137 -0.0114 -0.2583 …… -0.2405 0.3106 0.1534]；

由表 3.5 可知，使用训练好的词向量模型，计算“法律”与“法规”所对应的词向量的余弦值，其结果为 0.7666702270507812；计算“法律”与“合法”所对应的词向量的余弦值，其结果为 0.6702567934989929。对比两个余弦值，可以得出结论：相对于“法律”与“合法”， “法律”与“法规”所对应的词向量的余弦值更大，因此“法律”与“法规”具有更相近的语义。

3.3.3 经典算式

Word2vec 经典的词语算式例子[55]是“king - queen = man - woman”（国王−王后 = 男人−女人），可由此推出“king - queen + woman = man”（国王−王后+女人=男人）。

在本节中，使用另外一个示例进行验证，调用如下方法：

Collection<String> kingList = vec.wordsNearest(Arrays.asList("母亲", "父亲"), Arrays.asList("女儿"), 10);

结果会显示距离“母亲”−“女儿”+“父亲”这一向量最近的 10 个词，其中应当包括 “儿子”。wordsNearest 的第一个参数必须包括“母亲”和“父亲”这两个“正的”词，它们都带有加号；第二个参数则包括“女儿”这个“负的”词，带有减号，此处的正负不代表任何潜在的感情色彩；第三个参数是词表的长度，即希望得到多少个最接近的词。最终结果如表 3.6 所示：

表 3.6 算式结果

Table 3.6 Calculation results

编号	单词	与单词“父亲”之间的余弦值
1	父亲	1.0
2	儿子	0.8520398736000061
3	母亲	0.8476953506469727
4	父母	0.7992897629737854
5	相依为命	0.755556583404541
6	孝顺	0.7432591319084167
7	改嫁	0.7304872870445251
8	年迈	0.692865788936615
9	含辛茹苦	0.6793500781059265
10	老泪纵横	0.6607040762901306
……	……	……

3.4 本章小结

本章对神经网络语言模型实现——Word2vec 进行了介绍，并且对其中包含的 CBOW 模型和 Skip-gram 模型进行了介绍，接着以搜狗数据集为语料，使用 DL4J 工具包训练得到词向量模型，并在此模型上进行简单的相似度评估实验。

第 4 章 基于词向量和 K-NN 的文本分类算法

4.1 算法设计思想

上一章节讨论了语义相似度相关的问题，本章针对现有方法对语料中所蕴含上下文信息使用不足这一点，进行分类算法的研究。

本章介绍一种新的用于文本分类的方法——基于词向量和 K-NN 的文本分类方法 KKB-DC，其基本思想是：以搜集到的语料训练所得词向量模型为基础，使用 TF-IDF 算法提取方法从待分类文本中提取若干关键词，然后从词向量模型中取出关键词和各个类别对应的词向量，计算关键词对应的词向量与各个类对应词向量的余弦值，利用 K-NN 算法思想判断该关键词所属类别，然后将该关键词和与之所得余弦值大小分配到其所属的类，本文称之为关键词对该分类“贡献”。所有关键词分类之后，根据计算每个类所得平均贡献，选择平均贡献最大的类别作为待分类文本的类别。

本章将详细介绍实验准备过程及算法。图 4.1 为本文实验框架，训练集为中文新闻语料中的训练集，测试集为中文新闻语料的测试集，其中 KKB-DC 算法为框架核心，将在 4.5 节做详细介绍：

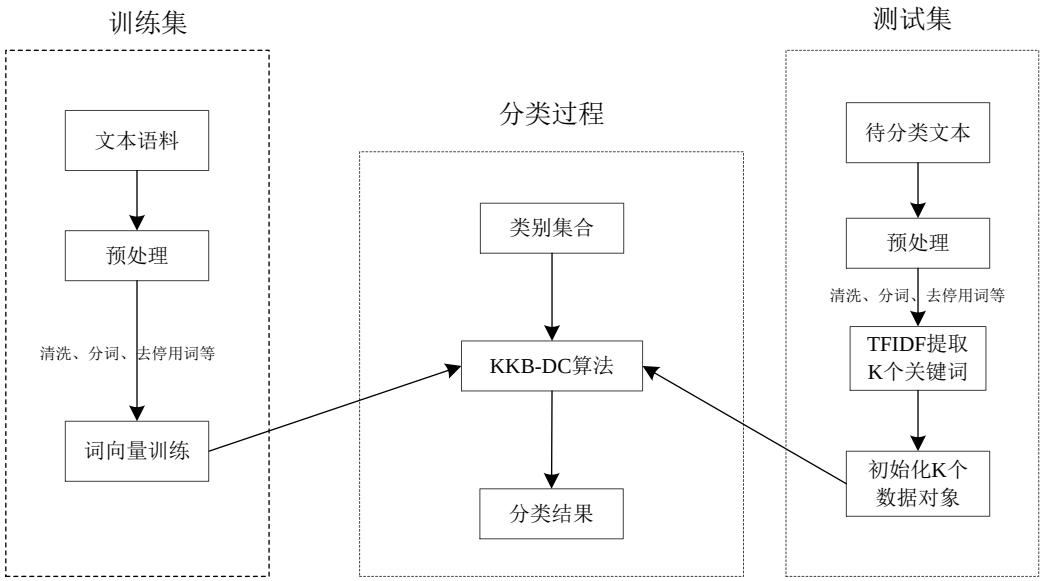

图 4.1 实验框架

Fig. 4.1 Experiment framework

在训练阶段，除了得到词向量模型以外，还会保存训练语料中所有词对应的 idf 值，以词-idf 值的 map 形式保存到本地。在使用 KKB-DC 算法对待分类文本进行分类时，可以直接使用词所对应的 idf 值。

4.2 语料收集

本实验使用的是数据堂提供的面向文本分类研究的中文新闻语料库[56]和搜狗实验室[57]提供的搜狐新闻数据与全网新闻数据。下面对两类语料进行介绍：

4.2.1 中文新闻语料库

该数据从凤凰、新浪、网易、腾讯等版面搜集，搜集时间在 2009 年 12 月—2010 年 3 月。中文新闻语料分为 8 类：Reading、Entertainment、History、Education、Society & Law、Culture、It、Military，数据规模如表 4.1 所示：

表 4.1 FinallyCorpus 数据集

Table 4.1 FinallyCorpus DataSet

类型	表单名称	文章 ID 范围	类别数目	是否为平衡语料
训练集	NewsTrainingCorpus	1- 13026	8	否
测试集	ReteursTestingCorpus	1-3254	8	否

数据库名称：FinallyCorpus.mdf，FinallyCorpus.ldf

数据格式：文本（MSSQL MDF 格式数据库）

从数据堂下载的数据文件解压后得到 mdf 格式文件，首先附加到 SQL Server 数据库中，然后从数据库中取出所需要的中文文档，包括训练集和测试集，并将其作为原始数据进行预处理。

4.2.2 搜狗语料

本实验中使用的搜狗语料包括搜狐新闻数据(SogouCS)和全网新闻数据(SogouCA)，数据来自若干新闻站点 2012 年 6 月至 7 月期间国内、国际、体育、社会、娱乐等 18 个频道的新闻数据。

得到的搜狗语料内容是 XML 格式的结构化数据，需从相应的标签中提取出正文内容，样例数据如表 4.2 所示：

表 4.2 搜狗数据样例

Table 4.2 sougou corpus sample

<doc>		
<url>	http://news.sohu.com/20120613/n345535702.shtml	</url>
<docno>	e4103f4f49da2142-69713306c0bb3300	</docno>
<contenttitle>	欧洲杯大战在即　荷兰葡萄牙面临淘汰将背水一战	</contenttitle>
<content>	中广网北京 6 月 1 3 日消息（记者王宇）据中国之声《新闻晚高峰》报道，明天凌晨两场欧洲杯的精彩比赛上演，死亡之组 B 组当中两支传统的强队……	</content>
</doc>		

以上步骤完成后，注意调整文本的编码格式，本实验中文本统一使用 utf-8 编码。

4.3 文本预处理

训练词向量模型，首先需要对获取的数据进行预处理操作，本实验中预处理操作流程如图 4.2：

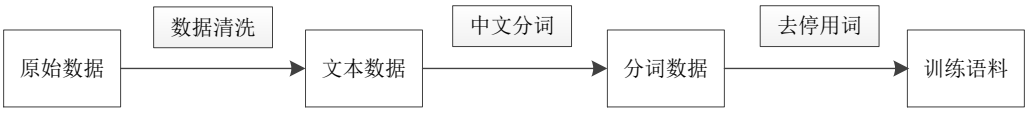

图 4.2 文本预处理流程图

Fig. 4.2 flow chart of preprocess

（1）数据清洗：此阶段针对数据库中中文新闻语料训练集和测试集

此步骤需要花费大量的时间，分析得到的文本数据，找出其中与分类无关的信息，如：“凤凰社实习记者 张三报道”、“更多新闻请点击链接……”等类似无用信息，建立规则把这些无用信息过滤掉。常见的无用信息举例如表 4.3：

表 4.3 无用信息总结

Table 4.3 useless information

类别	举例
新闻头部	**年**月**日 15:44 凤凰网文化专栏【大中小】【】0 位网友发表评论
作者信息	作者： 张三 编辑：李四
附加数据	(摘自加拿大《北美时报》作者:**法师）
声明信息	所有评论仅代表网友意见，凤凰网保持中立。
广告	更多文化内容请点击
占位字符	[1]【】【】【】【】【】【大中小】【发表评论(0)】
固定格式	[责任编辑: 张三]
地址、电话信息等	[配送地址]**路**号**楼 1107，坐地铁到**站下
……	……

从数据中总结出这些无用数据，建立相应的过滤规则，初步完成对数据集的清理工作。而对于搜狗数据的预处理则非常简单，只需把 content 标签中的内容取出即可。

（2）中文分词

过滤文本中无用信息之后，采用张华平博士研发的 NLPIR 分词工具包对文本语料进行分词，不保留词性标记信息。

（3）消除停用词

本实验中使用哈工大停用词表，并在此基础上，根据语料特点，对停用词表进行了丰富[58][59]。

表 4.4 描述各个类别文本数量。

表 4.4 中文新闻语料库文本分布情况

Table 4.4 Chinese news classification corpus text distribution

类别	culture	education	entertainment	history	it	military	reading	society&law
数量	2410	80	913	1302	283	1149	4558	3145

最终得到适合用于词向量训练的文本，中文新闻语料库经过预处理后得到大小为 58.5M，搜狗语料经过预处理后的大小为 2.83G。文本中的每一行代表原数据集中一篇文档，这种表示方法为计算整个训练数据集计算词的 idf 值提供基础。

4.4 词向量模型训练

经过预处理中文新闻语料已经符合训练词向量的条件，选取不同的配置参数训练数据模型，会得到不同的词向量文件。本文结合文献[39]对词向量模型，选取向量维度这一对词向量效果影响较大的参数进行配置。

使用 3.2.3 节中描述的配置对语料进行训练，选取向量维数为 50~500 维，以 50 递增对语料进行训练，训练得到不同配置参数下的词向量模型。最终得到 10 个分别包含 62839 个词的词向量模型，其中每个词所对应的向量均为指定维数的浮点型数组。

4.5 KKB-DC 算法

第 2 章中对相关技术作了介绍，本节将正式提出并描述 KKB-DC 算法。

算法以 word2vec 训练得到的词向量模型为基础，使用 TF-IDF 关键词提取算法从待分类文本中提取出 K 个关键词，由 2.6.3 节对文档频率与逆文档频率的介绍，TF-IDF 是一种统计方法，用以评估一个词对于一个文件集或一个语料库中的其中一份文件的重要程度，近年来不少人将 TF-IDF 用于特征的权值计算，并取得不错的效果。因此在本文中，认为这些关键词可以代表待分类文本参与到本文分类算法中；对于一个指定的关键词，结合已经训练出来的词向量模型，计算该关键词对应的词向量与各个类别在同一词向量模型所对应词向量之间的余弦值，使用 KNN 算法判断该关键词所对应的类别，其中每个类别选取 5 个词代表该类别参与到关键词类别判断中；一个关键词在分类活动中为文本分类产生的影响称为贡献（contribution），贡献的值为关键词与其所属类别对应词向量的余弦值。

本文从每一个待分类文本中选出 K 个关键词参与到待分类文本类别判断，在后续算法中也是对这 K 个关键词进行分析、处理。因此本文约定，K 个关键词产生不同的贡献值给相应的类别，计算每个类别所得平均贡献，选出得到平均贡献最大的类，则该类即为待分类文本所属的类。

以下是算法的具体描述：

待分类文本经过预处理之后：

(1)结合已经保存的由训练语料得到的 idf 数值和待分类文本中所有词的 tf 值，计算待分类文本中每个词的 tf-idf 值，按照从大到小的顺序抽取出该文本的 K 个词作为关键词；

(2)对每个关键词，如果词向量模型中包含该关键词，则用其对应的词向量分别与类别集合中每个类在词向量模型中对应的词向量进行余弦值计算，其中每个类别选取 5 个词代表该类别（这 5 个词可以人工归纳，也可以通过 word2vec 工具计算得出，本文综合两者考虑，对每个类别挑选出最能代表该类的词语参与到分类计算中），然后使用 KNN 算法找出该关键词所属的类别，并记录关键词与该类中词所对应词向量计算所得余弦值的最大值。本文中使用的 KNN 算法中的 k 为 5；如果词向量模型中不包含该关键词，则跳过该关键词进行下一个关键词的类别判断；

(3)把步骤(2)中计算所得余弦值作为该关键词对文本分类所做的“贡献”保存起来，并且记录该关键词所属类别；

(4)在对每个关键词进行类别判断并保存相应信息之后，把分在相同类别的关键词的“贡献”相加，除以分在该类别的关键词个数，得到该类所得平均贡献；

(5)最后一步选取所有类别中所得平均贡献最大的类别，为待分类文本的类别。

以上是对本文提出的基于词向量和 K-NN 分类算法（KKB-DC）的核心描述。算法流程图和伪代码分别图 4.3 如表 4.5 所示：

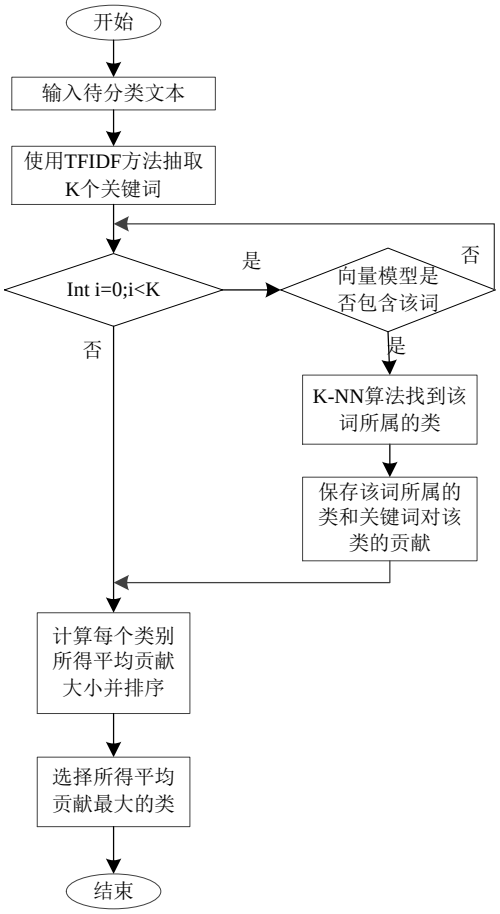

图 4.3 KKB-DC 算法流程图

Fig. 4.3 Flow chart of KKB-DC algorithm

表 4.5 KKB-DC 算法伪代码

Table 4.5 Pseudocode of algorithm KKB-DC

算法：KKB-DC 算法

```
输入：待分类文本 d，类别集合 classes, 词向量模型 Model
输出：文本所属分类 finalClass
(1) begin
(2) keywords = getKeyWordsByTFIDF(d);// 步骤 1，使用 TF-IDF 方法抽取出该文本的 K 个关键词
(3) foreach keyword[i] in keywords
(4) if(keyword[i] in Model)// 步骤 2，如果关键词在词向量模型中，继续向下执行，否则返回上一步
(5) begin
(6)     keyClass = getNearestClass(keyword[i]);//关键词与类对应的词向量进行余弦值计算，使用 K-NN 算法得到与该关键词最近的类的距离
(7)     save(keyword[i], keyClass);// 步骤 3，保存该关键词对文本分类所做的“贡献”和类别
(8) end
(9) foreach class[i] in classes
(10) begin
(11)    contribution[i] = getAverageContribution (classes);// 步骤 4，得到待分类文本属于每个类的概率；
(12) end
(13) finalClass = Max(contribution); //步骤 5，返回所有类别中所得平均贡献最大的类别
(14) end
```

如图 4.3 所示，如果某一关键词不被包含在词向量模型中，则跳过该关键词进入下一个关键词的类别判断。为避免关键词不被包含在词向量模型中，一方面训练词向量模型的时候要使用足够大的语料，使其能够包含尽可能多的词；另一方面避免使用极少数的关键词参与分类过程。

4.6 分析指标

为了对文本分类效果进行分析评估，本文使用准确率（precision）、召回率（recall）、综合准确率与召回率考虑的 F 值[60]。为便于表述，将结合表 4.6 进行描述：

表 4.6 文本分类结果

Table 4.6 text classification results

	实际属于此类	实际不属于此类
分类器判断属于此类	a	b
分类器判断不属于此类	c	d

（1）准确率

准确率指的是分类器判定属于此类中，实际属于此类文档的比例，计算公式如：

$$P = \frac{a}{a + b}$$

（2）召回率

召回率表示在实际属于这个类别的文档中，分类器判定属于该类的比例，计算公式如：

$$R = \frac{a}{a + c}$$

（3）F 值

准确率和召回率从两个角度对分类效果进行评价，但有时候两个会互补影响。为了综合考虑准确率与召回率，对两者这种取得较为全面的评估结果。这种评估方法称为 F 值，计算公式如：

$$F = \frac{2 * P * R}{P + R}$$

（4）宏平均

宏平均指所有类别的测试结果的算数平均值。

第 5 章 KKB-DC 算法的实验与探究

5.1 实验环境

本章所有进行的实验均在如下环境中进行：

电脑型号：华硕 X450J

处理器： Intel(R) Core(TM) i5-4200H CPU @ 2.80GHz

内存： 12GB (4+8GB) 1600MHz DDR3

硬盘： 128GB 2.5 英寸 Serial ATA 固态硬盘

操作系统：Windows® 7 专业版 64 位（简体中文）

开发平台：Eclipse Neon.2 Release(4.6.2)

Java Development Kit：1.8.0_92

Apache Maven：3.3.9

5.2 实验设计

本文选择使用中文新闻语料库进行与 KKB-DC 算法相关的实验。原因是该语料附加到数据库以后可以很方便的获取新闻类别信息，有助于训练集和测试集的选择。

搜狗语料是 2012 年由搜狗公司收集整理的，国际，体育，社会，娱乐等 18 个频道的新闻数据，提供 url 和正文信息。与中文新闻语料相比，搜狗语料没有明显类别区分标志，虽然可以参考 url 进行分类，但是不少 url 如今已经失效，根据 url 判断分类比较困难，因此本文仅使用搜狗语料进行词向量模型的训练，而不参与分类实验中测试集的选择上。从语料规模上，相比于搜狗语料的 2.83G，实验处理规模相对较小的中文新闻语料库速度更快，效率更高。

这里要注意一点，中文新闻语料库是 2010 年收集的，时间节点在搜狗数据集的前面，因为本文提出的算法有一定的语义要求，所以时间的先后顺序也会影响到最后分类结果。

本章设计实验探究可能会对本文提出算法 KKB-DC 的影响因素，分析本文所提分类算法的性能。

实验安排如下：

（1）在使用中文新闻语料训练的词向量模型下，探究 KKB-DC 算法在二分类和三分类实验中的表现，并对相应的结果进行分析；

（2）探究不同语料训练所得的词向量对文本分类的结果，这里特指使用大规模搜狗语料训练所得词向量对中文新闻语料测试集分类情况评估；

（3）维数是影响词向量质量的重要因素，设计实验探究在中文新闻语料提供的数据下，分类表现效果最好时向量的维数；

为便于表达，我们约定某一关键词在分类过程中的分数，指的是经过计算得到与该词最近的类的余弦值。关键词与类别之间的余弦值表明了该关键词分为某一类别的可能

性，本文所提 KKB-DC 算法选取最大余弦值所对应的类。某一关键词的分数，也表示该关键词最终被判别为某一类别的可能性大小。

5.3 分类算法影响因素探究

5.3.1 二分类实验

本实验探究在其他条件相同的情况下，关键词个数对二分类实验结果的影响。

实验所采用测试数据来自中文新闻语料测试集的娱乐、军事两个类别，这两个类别表 4.4 所中的 entertainment 和 military，在接下来的章节中，均使用中文类名作为类别标识。其中实验选取文化、民俗、传承、底蕴、内涵等词代表文化这一类别，军队、武器、演习、攻击、侦查等词代表军事这一类别。

使用由中文新闻语料的训练集训练所得的词向量模型，词向量的维数选取最常用的 200 维[39]。测试集每个类别随机抽取 500 条数据，同样经过上一章节中的预处理等一系列操作，进行分类实验。

最终得到分类结果的准确率按照类别统计，如图 5.1 所示：

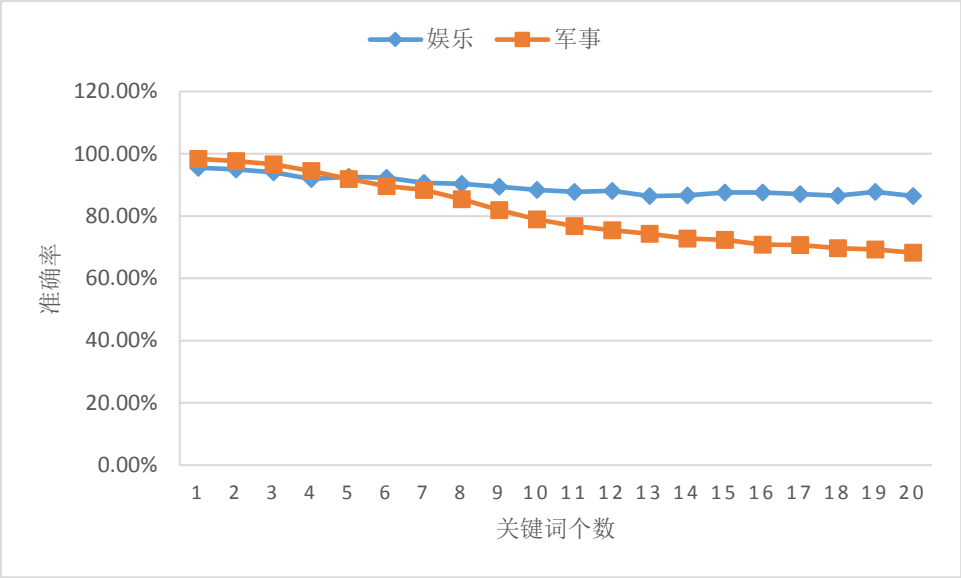

图 5.1 关键词数量不同条件下的二分类准确率

Fig. 5.1 Two categories accuracy with different number of keywords

由图 5.1 不难看出：（1）在关键词数较少的时候，二分类实验结果准确率较高；（2）但是随着关键词个数的增加，准确率逐渐递减。以下是对此现象的分析：

（1）在关键词个数较少时，KKB-DC 算法使用 TF-IDF 方法从待分类文本中提取出的关键词能够很好的代表该文本，即能够尽可能的保留待分类文本的信息。值得指出的是，在关键词个数小于 7 的时候，二分类实验中两个类别的分类准确率均超过 89.0%，因此可以看出 KKB-DC 在关键词个数较少的时候分类效果较好。

（2）随着关键词数量增多，分类准确率反而减少。出现这一现象的原因是，KKB-DC 算法按照 tf-idf 值抽取关键词时，越在后面的关键词，它的 tf-idf 值相对之前较小，

在文档分类过程中对分类结果的影响就越弱，甚至会对分类结果产生干扰，因此关键词数越多反而会影响文本分类的结果。

5.3.2 三分类实验

与二分类实验设置相同，本实验探究在其他条件相同的情况下，关键词个数对分类实验结果的影响。

实验所采用测试数据来自中文新闻语料测试集的娱乐、军事、社会三个类别，分别对应表 4.4 所中的 entertainment、military 和 society&law。其中娱乐、军事两个类别选词与二分类相同，实验选取社会、法律、犯罪、权力、道德等词代表社会这一类别。

最终得到分类结果的准确率按照类别统计，如图 5.2 所示：

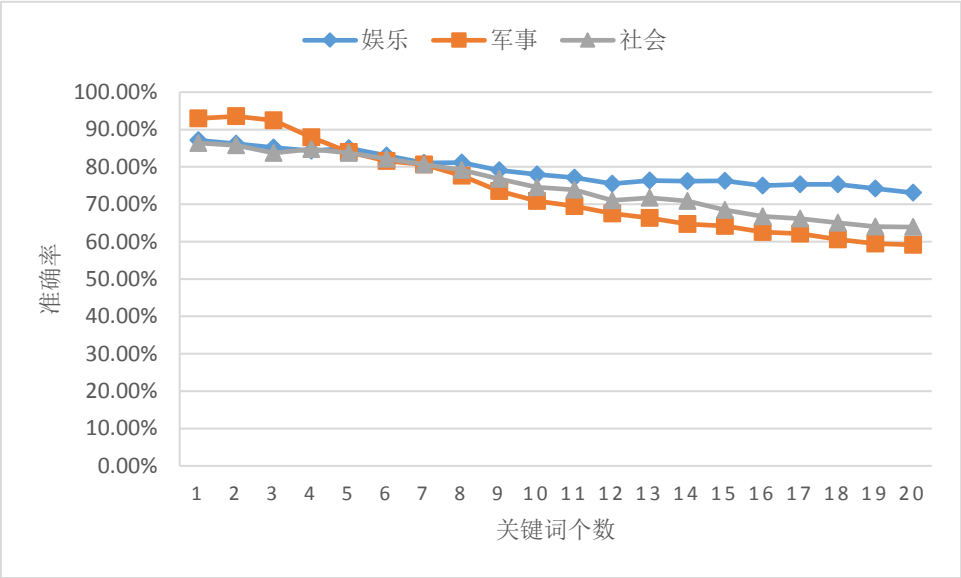

图 5.2 关键词数量不同条件下的三分类准确率

Fig. 5.2 Three categories accuracy with different number of keywords

由图 5.2 可以看出：（1）关键词个数较少时分类准确率较高；（2）随着关键词个数的增加，准确率逐渐递减；（3）但是整体准确率比二分类实验偏低。以下是对此现象的分析：

针对（1）和（2）两条，对比图 5.1 和图 5.2 可以发现，二分类和三分类的实验结果趋势整体是一致的。由图 5.1 可知，二分类结果准确率在关键词数量不超过 7 的时候，分类准确率均在 91%以上；由图 5.2 可知，二分类结果准确率在关键词数量不超过 7 的时候，分类准确率均在 80%以上。

（3）三分类整体准确率比二分类偏低。产生这一现象的原因是：随着待分类别数量的增加，KKB-DC 算法分类过程中不同类别之间会产生影响。举例如在二分类中某个关键词分类类别 1 的概率为 70%，分类类别 2 的概率为 50%，则会将此关键词分到类别 1 下；但是在三分类中，如果算法判别该关键词属于类别 3 概率为 80%，那么在三分类中就会把该词分到类别 3 中。

同时也应注意到不管是在二分类实验还是三分类实验中，关键词数量为 1 或 2 的时候，分类准确率都非常高，然而高准确率并不意味着分类效果非常好。在只有极少数关

键词个数的时候，一般情况下会选择待分类文档中得分最高的关键词所属的分类作为最终结果，但是因为中文新闻语料规模的限制，不可能把所有的词汇都包含在内，有可能会出现从一个待分类文本选出一个关键词而这个关键词在训练的词向量中没有对应的表示，这种情况的存在使得 KKB-DC 算法在分类过程会过滤掉一部分文本，这些文本不能正常的参与到分类活动中来。出现这一限制的根本原因是训练语料没能充分包含所有词汇。但是从另外一方面去分析，仅仅取出一个关键词或者两个关键词来代表一篇文本，这不符合本文所提算法的思想，即从文本中抽取的信息应尽可能的保留待分类文本的语义信息，显然仅使用一个或者两个关键词做法本身也是不合理的。

5.3.3 搜狗语料实验

本节实验探究不同训练语料训练得到的词向量模型对分类实验结果的影响。

上述实验所用词向量是由中文新闻文本语料训练集整理的语料训练得到，该语料规模较小，为 58.5M。本实验使用经过预处理的大小为 2.83G 的搜狗语料训练得到的词向量，与 5.3.1 和 5.3.2 节进行对比实验。实验设置与前两节相同，每个类别仍使用 500 条的测试集进行文本分类，探究在不同关键词个数下，不同语料对分类准确率的影响。

为便于区分数据，使用搜狗语料训练得到的词向量分类的结果使用“SG_”开头进行标记，最终得到的结果如图 5.3、图 5.4 所示：

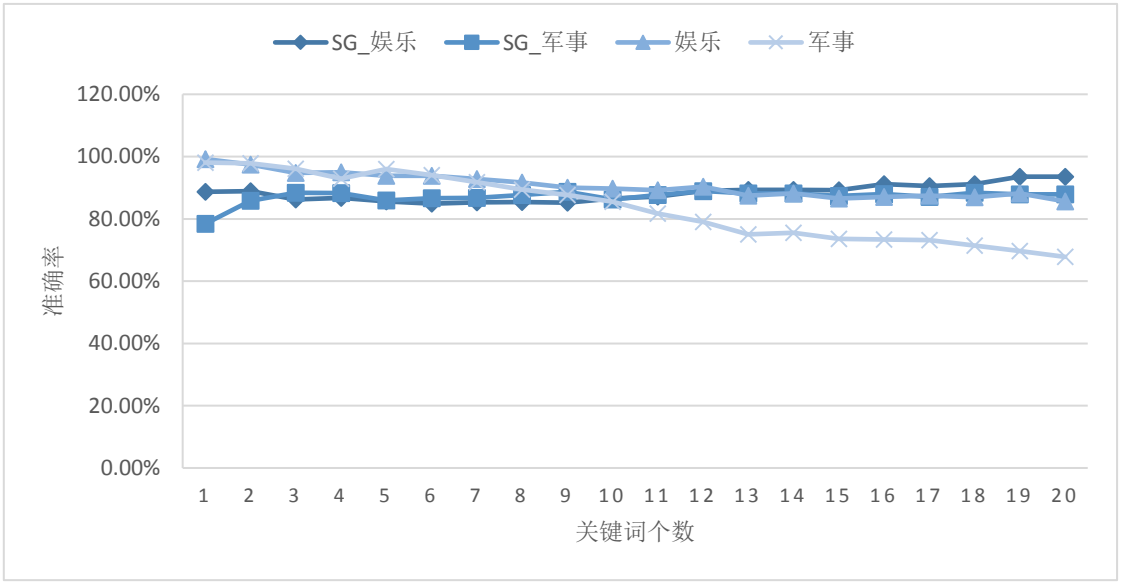

图 5.3 不同词向量二分类实验结果对比

Fig. 5.3 Two categories accuracy compared with different distributed representations

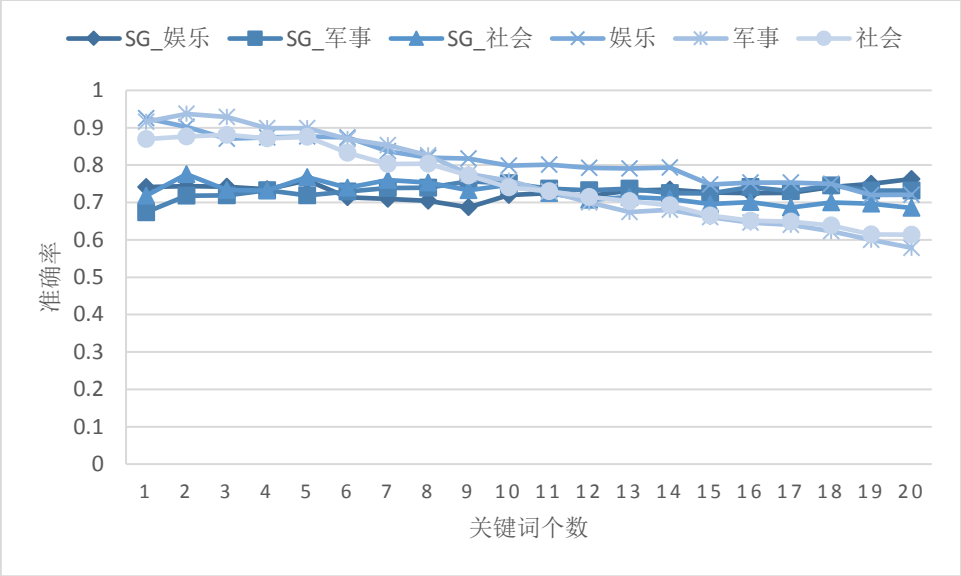

图 5.4　不同词向量三分类实验对比

Fig. 5.4　Three categories accuracy compared with different distributed representations

分析图 5.3 和图 5.4，可以得出：

（1）搜狗语料训练的词向量（以下简称：搜狗向量）整体表现平稳。随着关键词数量的增长，实验结果稳中有升，但趋势不明显。这表明大规模语料训练得到的词向量模型包含足够多词的语义信息，在关键词数量超过一定数量之后，继续增加关键词分类结果影响很小；

（2）如图 5.3，对比两种语料的表现，搜狗向量在关键词数量大于 11 个之后表现比中文新闻文本语料要好。分析可知，训练搜狗向量的语料规模所得词向量模型远比中文新闻语料训练的词向量模型所蕴含的信息大，关键词对应的词向量蕴含更多信息，因此对最终分类所造成的干扰就越少；而中文新闻语料训练所得词向量模型在面对更多关键词的时候，由于关键词词向量表现力受语料规模约束，因此关键词个数越多，分类过程中带来的干扰也就越多。

（3）如图 5.4，在关键词数量极少的情况下，搜狗向量分类效果变化比较大，这是因为只使用极少数量的关键词不足以表示整个文本，由此本算法中不建议使用较少关键词进行文本分类。

同时也应注意，在关键词数量极少的情况下二分类结果达到 98%以上，这个结果不意味着算法所选的词能够替代所在文本的所有信息，分析同 5.3.1 节，关键词数量较少的情况下，测试数据集有相当一部分没有能够参与到正常的分类活动中，这是因为在新的测试数据集中出现了新的词语，而这些词是不被包含在最初训练词向量的语料当中。值得注意的是，尽管实验中所使用的搜狗语料规模相对较大，但是待分类的测试数据集是从中文新闻语料集中随机抽取出来的，即使经过统一的数据清洗、中文分词、去停用词等这一些列预处理操作，因数据来源不同，仍有可能出现从待分类文本抽取出的关键词在词向量模型中没有对应向量这一现象。

由以上分析可以得出结论：训练词向量模型的语料规模和来源对 KKB-DC 算法性能有十分重要的影响。训练语料数据规模越大，算法分类效果越稳定；训练语料与待分类文本在同一个领域并且收集时间越近，分类效果越好。

这对选取训练语料提供一个方向：选择数据尽可能多的，同一来源且尽可能是最近一段时间的语料进行词向量模型的训练。对于需要做新闻分类的门户网站，使用本网站最近三个月或者半年的大规模数据进行训练，即可保证对新的待分类文本进行准确分类。

5.3.4 词向量维度选择

文献[39]总结词向量维度是决定词向量模型质量的关键因素，因此本节设计实验观察词向量维度对实验结果的影响。

由对以上几个小节的实验分析，本实验选取关键词数量为 5，实验采用的仍是娱乐、军事、社会三个类别的数据，使用由中文新闻语料的训练集训练所得的词向量模型，实验设置与 5.3.3 节相同。

最终得到结果按照类别统计，如图 5.5 所示：

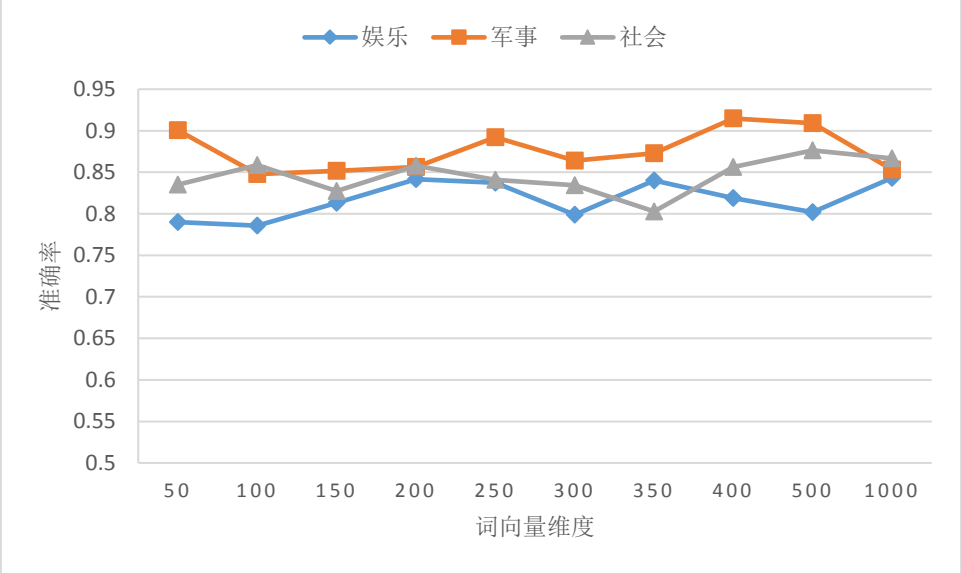

图 5.5 不同词向量维度对分类结果的影响

Fig. 5.5 Infect of categories accuracy with different word dimension

结合图 5.2 与图 5.5 进行分析，不同维度的词向量对分类结果影响较大：

（1）维度偏小不足以表达训练语料中的语义信息，如图 5.5 所示，词向量为 50 维的时候，不同类别分类结果差距较大；

（2）反之，词向量维度偏大如 400 维，维数过多导致语料信息存储分散，也存在分类结果相差较大的情况。

结合本文使用的训练语料，选取使用 200 维训练词向量模型是较好的选择。

5.4 本章小结

本章针对上一章节所提的 KKB-DC 算法设计实验，探究不同影响因素对分类实验结果的影响。分别探究在二分类、三分类实验中算法所选关键词个数对实验结果的影响，

算法使用的由不同数量级的语料训练所得词向量模型在相同实验配置下的对比，以及最佳词向量维度的选择。

第 6 章 KKB-DC 算法的改进

6.1 概述

第 4 章介绍了本文提出的基于词向量的 KKB-DC 算法，并且第 5 章通过实验证明了算法的可行性。但是仔细观察图 5.1 和图 5.2 可以看出，KKB-DC 中使用中文新闻语料训练词向量模型，在选取关键词个数较少的时候分类结果准确率较高，但是随着关键词个数的增加，分类准确率却下降的比较迅速。

结合图 5.4, 发现，使用由更大规模的搜狗语料训练的词向量，发现在关键词较多的时候表现更加稳定，由 5.3.3 的分析可知，使用更大规模语料训练的词向量模型可以保留更多词的语义信息，算法中词向量模型可以通过增加训练语料的规模进行优化。

另一方面，KKB-DC 算法可以在所使用的词向量模型固定的情况下，从待分类文本中所提取的关键词改进算法。KKB-DC 算法中使用关键词与类别之间的余弦值作为该关键词对该分类的“贡献”，既然关键词在待分类文本中有着不同的 tf-idf 值，因此可以把关键词对待分类文本的重要程度，即关键词的 tf-idf 值引入到分类算法当中。则本文实验框架修改之后如图 6.1 所示：

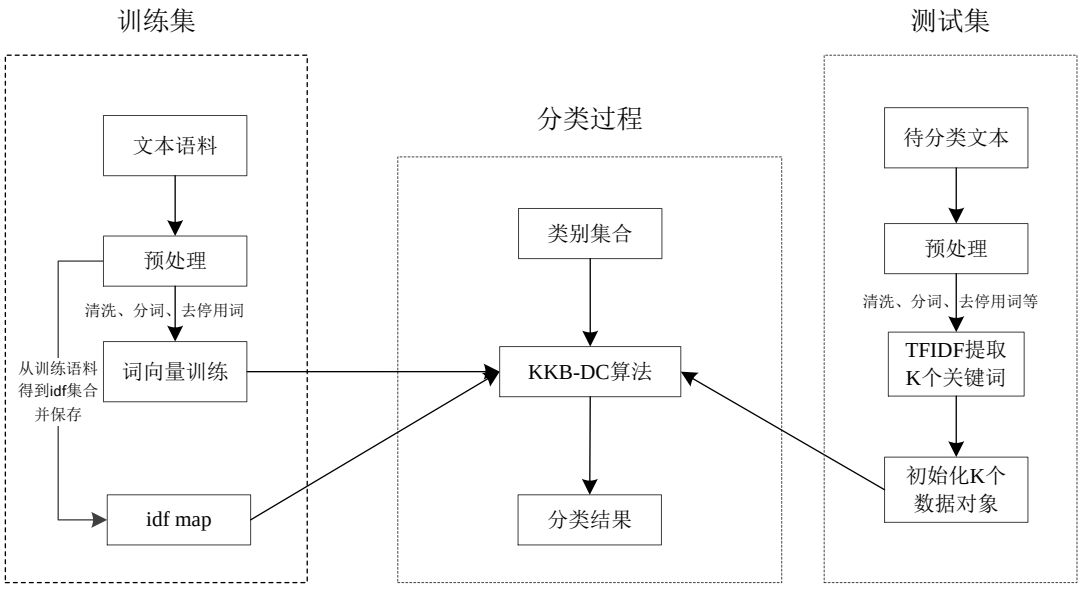

图 6.1 实验框架修改

Fig. 6.1 Modified experiment framework

6.2 改进算法描述

经实验分析，本文将待分类文本中关键词的重要程度引入算法的权值优化当中，这样不仅充分利用由训练语料所得的词的语义信息，也考虑到了待分类文本中不同词的表现力。因此可修改 4.5 节中贡献的计算方法：

一个关键词在分类活动中为文本分类产生的影响称为贡献，贡献的值大小为关键词与其所属类别对应词向量的余弦值乘以该关键词的 tf-idf 值。其中，该关键词的 tf-idf 值为从 idf map 中取出的该关键词的 idf 值与该关键词在待分类文档词频的乘积。

引入待分类文本 tf-idf 值之后的算法描述如下：待分类文本经过预处理之后，

(1)结合已经保存的由训练语料得到的 idf 数值和待分类文本中所有词的 tf 值，计算待分类文本中每个词的 tf-idf 值，按照从大到小的顺序抽取出该文本的 K 个词作为关键词；

(2) 对每个关键词，如果词向量模型中包含该关键词，则用其对应的词向量分别与类别集合中每个类在词向量模型中对应的词向量进行余弦值计算，其中每个类别选取 5 个词代表该类别（这 5 个词可以人工归纳，也可以通过 word2vec 工具计算得出，本文综合两者考虑，对每个类别挑选出最能代表该类的词语参与到分类计算中），然后使用 KNN 算法找出该关键词所属的类别，并记录关键词与该类中词所对应词向量计算所得余弦值的最大值。本文中使用的 KNN 算法中的 k 为 5；如果词向量模型中不包含该关键词，则跳过该关键词进行下一个关键词的类别判断；

(3)把步骤(2)中计算所得余弦值与该关键词的 tf-idf 值相乘作为该关键词对文本分类所做的“贡献”保存起来，并且记录该关键词所属类别；

(4)在对每个关键词进行类别判断并保存相应信息之后，把分在相同类别的关键词的“贡献”相加，除以分在该类别的关键词个数，得到待分类文本属于该分类的概率；

(5)最后一步选取所有类别中所得平均贡献最大的类别，为待分类文本的类别。相应的表 4.5 中伪代码只需把第 7 行“save(keyword[i], keyClass);”改为“save(keyword[i] * tfidf, keyClass);”即可。

6.3 改进算法对比

为对比 KKB-DC 算法改进前后的分类效果，本节使用与 5.3.1、5.3.2 节相同的实验设置，重新进行二分类和三分类实验，实验结果如下：

6.3.1 二分类实验对比

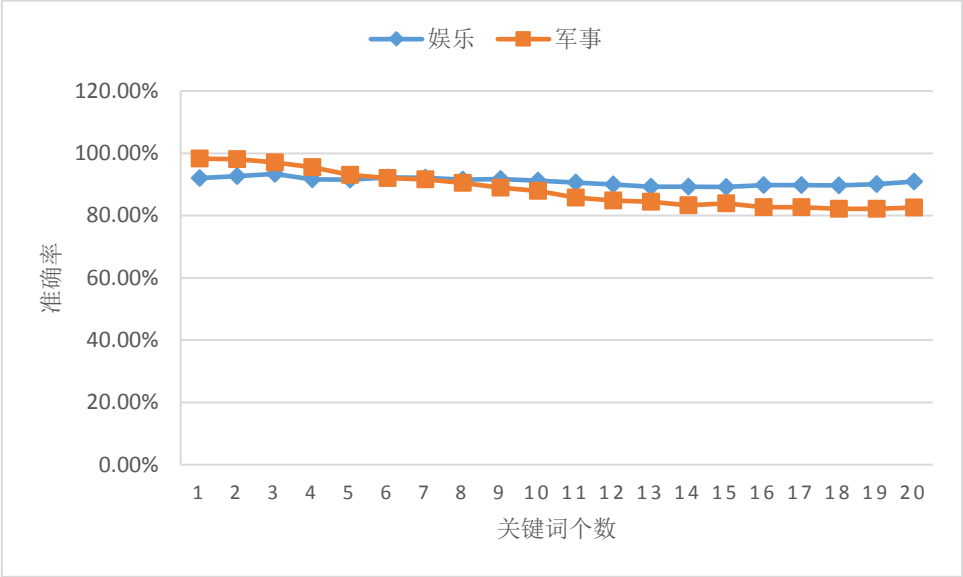

图 6.2 关键词数量不同条件下的二分类准确率

Fig. 6.2 Two categories accuracy with different number of keywords

与图 5.2 对比发现，图 6.2 同样是在关键词数较少的时候，二分类实验结果准确率较高；随着关键词个数的增加，准确率逐渐下跌。不同的是算法改进之前的二分类准确率在关键词个数为 20 的时候跌至 68.22%，而改进后的算法准确率则始终保持在 82%以上。这说明把待分类文本关键词的 tf-idf 值引入到算法当中是可行的，能够更加充分的利用待分类文本的信息，提高算法的稳定性。

6.3.2 三分类实验对比

改进之后的算法仅仅在二分类实验中表现良好还不能完全说明引入关键词的 tf-idf 值是正确的，接下来进行三分类实验在算法改进前后的对比分析，算法改进之后三分类实验结果如图 6.3 所示：

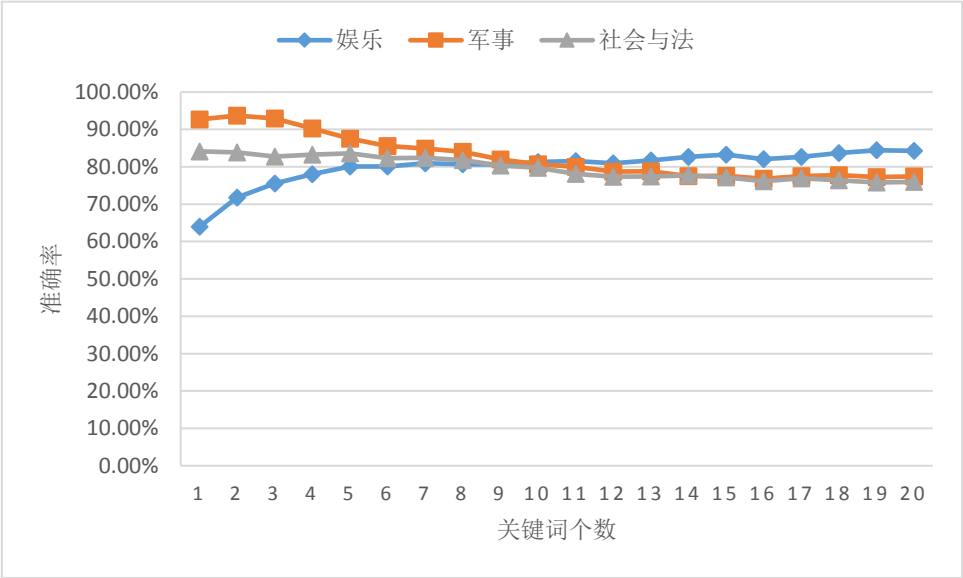

图 6.3 关键词数量不同条件下的三分类准确率

Fig. 6.3 Three categories accuracy with different number of keywords

与图 5.3 对比可以看出，图 6.3 在改进算法之后，关键词数较少的时候，三分类实验结果其中两类准确率较高，娱乐这一类准确率较低；随着关键词个数的增加，准确率逐渐下跌，但是趋势不明显。值得指出的是算法改进之前的三分类准确率在关键词个数为 20 的时候跌至 59.14%，而改进后的算法准确率则始终保持在 75%以上。

针对算法改进之后，三分类在关键词个数较少的情况下分类情况不稳定这一现象进行分析，同 5.3.2 节原因一样，在只有极少数关键词个数的时候，因为中文新闻语料规模的限制，不可能把所有的词汇都包含在内，有可能会出现从一个待分类文本选出一个关键词而这个关键词在训练的词向量中没有对应的表示，这种情况的存在使得 KKB-DC 算法在分类过程会过滤掉一部分文本，这些文本不能正常的参与到分类活动中来。由此，本文认为在关键词个数过少的情况下，使用任何关键词选择算法选出的关键词均不足以代表原待分类文本，因此对如图 5.2、图 5.3、图 6.2 所示关键词较少时分类准确率较高这一现象持保留态度。

综上实验分析，本章提出改进的 KKB-DC 算法能够在二分类和三分类的条件下取得更加稳定的分类结果。尤其是二分类结果，部分情况下准确率可以达到 94.64%。对于本算法具体总结如下：

（1）针对本实验，受语料限制，在关键词较少的情况下分类相对较好，这表明算法能够较好的提取出能够表示文本信息的关键词，而这些关键词也能够代表其所在文本的语义信息；

（2）在关键词数量极少的情况下，虽然准确率比较高，但是可信度较低，原因是受语料限制，不能充分覆盖所有的字、词，在关键词个数极少的时候能够参与准确分类活动的并非所有待测文本，因此本文对于关键词数量极少的情况仍有较高准确率这一结果暂持保留态度；

（3）实验也证明，本算法对词向量质量要求比较高。即使使用由大规模语料训练得到的搜狗向量，在关键词数量较少的情况下效果依然不如由测试集所在的语料训练得出的词向量效果更好，一是这两种语料来源不尽相同，另外一个比较重要的是两种语料的收集时间有一定差异。因此在与测试数据集语义信息的耦合方面存在差异；

（4）在本实验所用语料条件下，实验得出最适合参与算法的词向量的维数是 200 维，关键词个数取 4 到 7 个为宜。

6.4 分类算法对比实验

为证明本文所提算法的有效性，本节使用不同的文本分类方法与本文所提的 KKB-DC 算法进行对比实验。

6.4.1 KKB-DC 与 LibSVM、LibLinear 分类模型对比实验

LibSVM 和 LibLinear 是台湾大学林智仁(Lin Chih-Jen)教授等[61]开发设计的软件包，LibSVM 主要是用来进行非线性 SVM 分类器的生成；LibLinear 是应对大规模的数据分

类而创建的，Linear 分类器的训练比非线性分类器的训练计算复杂度要低很多，时间也少很多，而且在大规模数据上的性能和非线性的分类器性能相当。

本节使用清华大学自然语言处理实验室推出的中文文本分类工具包 THUCTC[62]（THU Chinese Text Classification），能够自动高效地实现用户自定义的文本分类语料的训练、评测、分类功能，THUCTC 对于开放领域的长文本具有良好的普适性，不依赖于任何中文分词工具的性能，具有准确率高、测试速度快的优点，包含 LibSVM 和 LibLinear 两个分类模型的实现。本节将使用这两种分类模型与本文所提 KKB-DC 文本分类算法的进行对比。

实验参数设置如下：

（1）对比实验选取二字串 bigram 作为特征单元，即 n-gram 的 n 取值为 2，特征降维方法为 Chi-square，权重计算方法为 tf-idf，分类模型使用的是 LibSVM 模型和 LibLinear 模型，所采用的训练集与测试集同样是中文新闻语料库中随机选取，每个类别选取 500 条数据，训练集占总文件数比例为 70%；

（2）参考实验为改进后的 KKB-DC 算法，待分类文本抽取的关键词个数设置为 5，使用由中文新闻语料的训练集训练得到的 200 维词向量模型；

使用以上参数设置进行实验，结果如表 6.1 所示：

表 6.1 三分类实验在不同算法下准确率对比

Table 6.1 Three categories accuracy with different algorithm

	LibSVM/%	LibLinear/%	KKB-DC/%
娱乐	100	100	87.68
军事	85.45	85.45	89.88
社会	97.18	97.18	87.56

通过对比发现，在特征单元、特征降维方法、权重计算方法等相同的情况下，THUCTC 工具包实现的 LibSVM 模型与 LibLinear 模型分类结果相同。需要指出，在娱乐这一类别的分类中两者准确率都达到了 100%。但是相比之下，本文所提 KKB-DC 方法表现更加稳定，准确率在 87%~90%之间，其中军事类别比对比实验采取的方法提高了 4.43%。分析表现稳定原因是 KKB-DC 算法不完全依赖训练数据集的词的统计信息，而且充分利用训练所得词的语义信息，因此在做分类决策时不会仅凭待分类文本的词统计信息做出判断，从而表现出稳定的分类能力。

6.4.2 KKB-DC 与 TFIDF+SVM、LDA+SVM 算法对比

本节使用中文新闻语料中的娱乐、军事、社会三个类别的数据进行实验。

本文提出的 KKB-DC 算法进行分类实验，每个类别随机选取 500 条数据参与分类，词向量模型训练采用默认配置，向量维度设置为 200 维，关键词个数选择 5；

对比算法为文献[63]，该文献分别使用 TF-IDF 提取出待分类文本的关键词，使用 LDA 得到待分类文本的主题模型向量，然后结合 SVM 分类器，得到 TFIDF+SVM 和

LDA+SVM 两种分类方法。这两种对比分类方法，同样是每个类别随机抽取 500 条数据参实验，训练数据集与测试数据集数据比例为 9:1。

最终分类结果如表 6.2 所示：

表 6.2 分类结果

Table 6.2 Result of different algorithm

	Precision/%	Recall/%	F-measure/%
娱乐	87.47	89.57	88.51
军事	88.26	81.34	84.65
社会	83.20	87.47	85.29

将对比结果采用宏平均值表示，如图 6.4 所示。

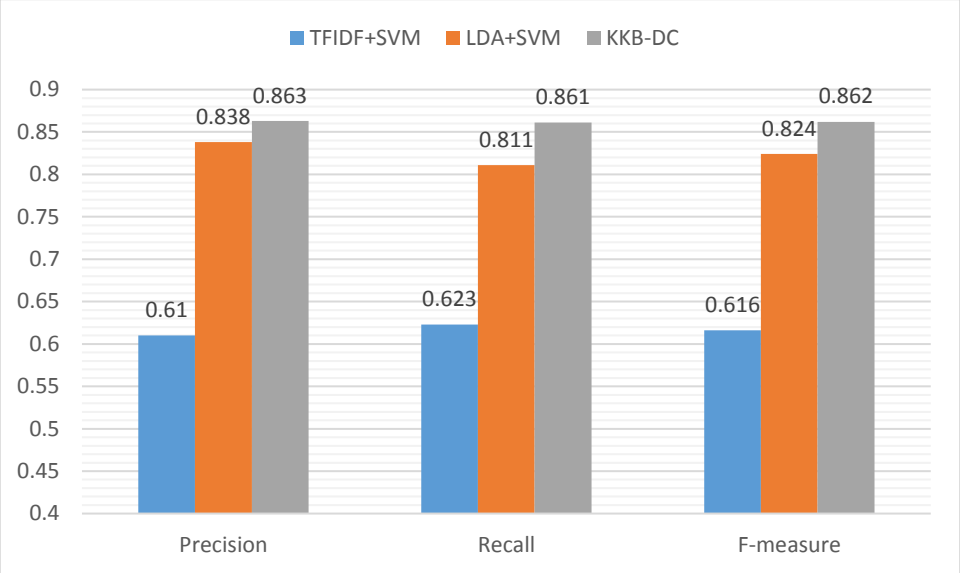

图 6.4 三种方法的准确率、召回率和 F 值的对比

Fig 6.4 precision recall and F measure between three algorithm

从图 6.4 中结果可以看出，在使用中文新闻数据集进行的文本分类中，无论是准确率、召回率还是 F 值，本文提出的 KKB-DC 算法的分类效果好于文献[63]中基于 TFIDF 的 SVM 分类算法和基于 LDA 的 SVM 分类算法，证明文本所提 KKB-DC 算法的有效性。

6.5 本章小结

针对算法在关键词数量较多时分类准确率下降较快这一现象，本章对所提算法进行优化，通过引入待分类文本关键词的 tf-idf 值来优化算法中关键词对其分类的贡献；然后设计实验与优化之前的算法进行对比；最后使用优化后的算法与现有分类算法进行对比，证明本文所提算法的有效性。

第 7 章 总结与展望

7.1 全文总结

本文通过文本分类技术的研究与分析，包括数据清洗的方法、中文分词技术、文本表示模型、特征选择方法、分类算法、语言模型等，提出在现有的分类算法不能有效利用原始语料中的语义信息，语义信息的丢失会限制分类器的分类性能的提升，因此本文重点关注文本的语义信息的利用和分类算法性能的提升上。

本文整理使用网上公开的数据集进行相关的分类实验，使用谷歌开源的 Word2vec 工具对训练语料进行训练，结合现有分类技术，提出了基于词向量和 K 近邻算法提出新的文本分类算法，本文的主要成果如下：

（1）提出了一种基于词向量和 K-NN 分类思想，以 tf-idf 和词对应的词向量之间的余弦值为权重，关键词投票决定分类的方法——KKB-DC，该算法的优点是前期实验得出适合算法使用的训练词向量模型的配置参数，模型训练完成之后，再进行分类任务直接加载调用词向量模型即可，而且在训练数据达到一定规模之后，在选取词向量维度确定的情况下，总的词数逐渐趋于稳定，所训练的词向量模型也会固定大小，这对大规模文本分类提供了一个高效的分类途径；

（2）通过实验证明该算法对中文文本分类的可行性，通过组织实验探究不同维度词向量、关键词个数以及测试文本数量对分类效果的影响，根据实验结果分析实验数据，得到相应的结论。针对算法在关键词数量较多时分类准确率下降较快这一现象，对提出算法进行分析、改进，使得分类效果更佳稳定。实验证明在算法的权值优化中引入关键词的 tf-idf 值，可以有效解决关键词个数选取较多时分类准确率下降较快这一问题。；

（3）本文通过现有的分类算法进行对比实验，分析本文所提 KKB-DC 算法的表现性能。与清华大学提供 THUCTC 工具包实现的分类算法相比，结果显示使用本文提出的算法，虽然整体分类效果不如已有 LibSVM、LibLinear 分类模型，但是在个别类别的分类上优于前者，并且在现有规模的语料下分类表现稳定；与 TFIDF+SVM 和 LDA+SVM 两种分类方法相比，在准确率、召回率、F 值等方面，本文所提方法均优于对比实验，从而证明本文所提算法的有效性。

7.2 工作展望

本文对于文本分类的研究还非常有限，该算法中有些地方仍需要进一步的改善。以下是未来研究可能涉及到的关键点：

（1）实验处理中文新闻语料所得数据集大小仅为 58M，而词向量的训练语料一般是要求比较大的。本文尽可能在预处理阶段对数据集做的充分处理，但是数据量相对较小，无法涵盖所有词汇，导致在测试分类阶段有些文本找不到关键词，无法参与分类，致使少数文本的关键词不能全部参与分类计算。因此可使用更大规模数据集上进行改进，以生成质量更高、词汇覆盖范围更广的词向量模型；

（2）实验选取的语料分别在 2010 年和 2012 年收集，若对最近几年产生的文本数据进行分类，可能出现一定的偏差。原因同上所述，随着社会的发展，尤其是新闻文本中会迅速产生大量新的词汇，如果不及时做到更新训练语料，也将对分类结果产生较大影响。

（3）受语料限制，本文所选分类类别数为二分类、三分类，如果进一步丰富训练词向量的语料，则可以得到更高质量的词向量，并且对有望扩充至更多类别的分类。

参考文献

[1] 互联, 中心. 第 39 次《中国互联网络发展状况统计报告》[R]. 北京:中国互联网络信息中心, 2017.

[2] 王仁武. Python 与数据科学[M]. 华东师范大学出版社, 2015.267

[3] Luhn H P. Auto-encoding of documents for information retrieval systems[M]. IBM Research Center, 1958

[4] Maron M E, Kuhns J L. On relevance, probabilistic indexing and information retrieval[J]. Journal of the ACM (JACM), 1960, 7(3): 216-244.

[5] Salton G, Wong A, Yang C S. A vector space model for automatic indexing[J]. Communications of the ACM, 1975, 18(11): 613-620

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

[7] 祝晓鲁, 白振兴, 贾海燕. 自动文本分类技术研究[J]. 现代电子技术,2007,30 (3):121-124

[8] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating er-rors[J]. Nature, 1986, 323(6088):533-536

[9] 周练. Word2vec 的工作原理及应用探究[J]. 科技情报开发与经济，2015（25）02：145-148

[10] Collobert R, Weston J. A unified architecture for natural language processing:deep neural networks with multitask learning[C]// International Conference. DBLP, 2008:160-167.

[11] Turian J, Ratinov L, Bengio Y. Word representations: a simple and general method for semi-supervised learning[C]//Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, 2010: 384-394

[12] Glorot X, Bordes A, Bengio Y. Domain adaptation for large-scale sentiment classification: A deep learning approach[C]//Proceedings of the 28th international conference on machine learning (ICML-11). 2011: 513-520

[13] Bengio Y. Deep learning of representations for unsupervised and transfer learning[J]. ICML Unsupervised and Transfer Learning, 2012, 27: 17-36.

[14] Word2vec. google[EB/OL].https://code.google.com/p/Word2vec/

[15] Word2vec code. google[EB/OL]. http://Word2vec.googlecode.com/svn/trunk/

[16] Mikolov T, Yih W T, Zweig G. Linguistic regularities in continuous space word representations[J]. In HLT-NAACL, 2013

[17] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013

[18] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]//Advances in neural information processing systems. 2013: 3111-3119

[19] 周水庚， 关佶红， 胡运发． 隐含语义索引及其在中文文本处理中的应用研究[J]. 小型微型计算机系统，2001, 22(2):239-243.

[20] 朱靖波， 陈文亮． 基于领域知识的文本分类[J]． 东北大学学报(自然科学版)，2005, 26(8):733-735.

[21] 黄秀丽， 王蔚． 一种改进的文本分类特征选择方法[J]． 计算机工程与应用，2009, 45(36):129-130.

[22] 张玉芳， 王勇， 刘明, 等． 新的文本分类特征选择方法研究[J]． 计算机工程与应用, 2013, 49(5):132-135.

[23] 熊富林, 唐晓晟, 邓怡豪. Word2vec 核心架构及其在中文处理中的应用[J]. 2014

[24] 王明亚. 基于词向量的文本分类算法研究与改进[D]. 华东师范大学，2016

[25] 百度百科. 中文信息处理[EB/OL]. http://baike.baidu.com/item/中文信息处理?sefr=cr,2016

[26] 百度百科. 中文分词[EB/OL].http://baike.baidu.com/item/中文分词?sefr=enterbtn, 2016

[27] Zheng X, Chen H, Xu T. Deep Learning for Chinese Word Segmentation and POS Tagging[C]//EMNLP. 2013: 647-657

[28] Li S, Zhou G, Huang C R. Active learning for Chinese word segmentation[C]//In COLING (Posters. 2012

[29] Wang L Z H, Mansur X S M. Exploring representations from unlabeled data with co-training for Chinese word segmentation[J]. 2013

[30] Sun X, Wang H, Li W. Fast online training with frequency-adaptive learning rates for chinese word segmentation and new word detection[C]//Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1. Association for Computational Linguistics, 2012: 253-262

[31] Qian X, Liu Y. Joint chinese word segmentation, pos tagging and parsing[C]//Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics, 2012: 501-511

[32] Li S, Zhou G, Huang C R. Active learning for Chinese word segmentation[C]//In COLING (Posters. 2012

[33] 张华平. NLPIR 汉语分词系统. [EB/OL]. http://ictclas.nlpir.org/

[34] 李岩. 基于深度学习的短文本分析与计算方法研究[D]. 北京科技大学, 2016

[35] Salton G, Buckley C. Term-weighting approaches in automatic text retrieval[J]. Information processing & management, 1988, 24(5): 513-523

[36] 唐明, 朱磊, 邹显春. 基于 Word2Vec 的一种文档向量表示[J]. 计算机科学, 2016, 43(6):214-217

[37] 江大鹏. 基于词向量的短文本分类方法研究[D]. 浙江大学,2015

[38] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. Journal of Machine Learning Research, 2003, 3:993-1022.

[39] Wu Q, Deng X, Zhang C, et al. LDA-based model for topic evolution mining on text[C]// International Conference on Computer Science & Education. IEEE, 2011:946-949.Lai S, Liu K, He S, et al. How to generate a good word embedding[J]. IEEE Intelligent Systems, 2016, 31(6): 5-14

[40] Mikolov T, Le Q V, Sutskever I. Exploiting Similarities among Languages for Machine Translation[J]. Computer Science, 2013

[41] 维基百科. 主成分分析 [OL]. [2016-7-11] https://zh.wikipedia.org/wiki/主成分分析

[42] Troussas C, Virvou M, Espinosa K J, et al. Sentiment analysis of Facebook statuses using Naive Bayes classifier for language learning[C]// Fourth International Conference on Information, Intelligence, Systems and Applications. IEEE, 2013:1-6

[43] 维基百科. 语言模型. [OL]. [2013-3-12]. http://zh.wikipedia.org/zh-cn/语言模型

[44] Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model[J]. Journal of machine learning research, 2003, 3(Feb): 1137-1155

[45] Cortes C, Vapnik V. Support-vector networks[J]. Machine learning, 1995, 20(3): 273-297

[46] 李航. 统计学习方法[J].清华大学出版社, 北京, 2012

[47] 范小丽, 刘晓霞. 文本分类中互信息特征选择方法的研究[J].计算机工程与应用, 2010,46(34):123-125

[48] Church K W, Hanks P. Word association norms, mutual information, and lexicography[J]. Computational linguistics, 1990, 16(1): 22-29

[49] 维基百科. TF-IDF [OL]. [2017-3-9].https://zh.wikipedia.org/wiki/TF-IDF

[50] Robertson S. Understanding inverse document frequency: on theoretical arguments for IDF[J]. Journal of documentation, 2004, 60(5): 503-520

[51] Witten I H, Frank E, Hall M A, et al. Data Mining: Practical machine learning tools and techniques[M]. Morgan Kaufmann, 2016

[52] Chau A L, Li X, Yu W. Support vector machine classification for large datasets using decision tree and Fisher linear discriminant[J]. Future Generation Computer Systems, 2014, 36: 57-65

[53] Tomas Mikolov.Word2vec project [EB/OL].[2014－09－18]. https//code.google.com/p/Word2vec/

[54] Guthrie D, Allison B, Liu W, et al. A closer look at skip-gram modelling[C]//Proceedings of the 5th international Conference on Language Resources and Evaluation (LREC-2006). 2006: 1-4

[55] Word2vec. Deeplearning4j[EB/OL]. https://deeplearning4j.org/deeplearning4j

[56] 面向文本分类研究的中英文新闻分类语料. 数据堂[EB/OL]. http://more.datatang.com/data/13484

[57] 搜狗全网新闻数据. 搜狗公司[EB/OL]. https://www.sogou.com/labs/resource/ca.php

[58] Niu K. 停用词表[EB/OL].https://github.com/Niukun/NK_TextClass/blob/master/trunk/NK_TextClass_3.0/source/ stopWords

[59] Qiu S.停用词表[EB/OL].https://github.com/JNU-MINT/ TextBayesClassifier/中文停用词表(1208 个).txt

[60] Powers D M. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation[J]. 2011

[61] Chang C C, Lin C J. LIBSVM: a library for support vector machines[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2011, 2(3): 27

[62] 中文：孙茂松，李景阳，郭志芃，赵宇，郑亚斌，司宪策，刘知远. THUCTC：一个高效的中文文本分类工具包. 2016

[63] Wu X, Fang L, Wang P, et al. Performance of using LDA for Chinese news text classification[C]//Electrical and Computer Engineering (CCECE), 2015 IEEE 28th Canadian Conference on. IEEE, 2015: 1260-1264

致谢