# **文本分类和文本聚类**

# 巩卫参 喻云飞 熊啸楠 周月阳

摘要：随着当今世界信息化时代的迅猛发展,大量的数据信息呈现出爆炸式的增长态势,而且随着互联网的进步,这些海量数据的传播速度也日益加快。对于这些数据的处理是目前亟待解决的问题，通过文本预处理和文本表示模型，可以将文本转化为计算机可以识别的数据，为文本分类和文本聚类提供条件。文章第四章对文本分类进行了综述，按照文本分类流程，从预处理、特征表示、特征提取、文本分类机器学习算法以及深度学习算法作了介绍。机器学习算法包括Rocchio算法、KNN算法、决策树、朴素贝叶斯和支持向量机。深度学习包括fastText、TextCNN、TextRNN和TextRCNN等。文本聚类作为涵盖了数据挖掘、机器学习和自然语言处理等领域的一种重要的文本挖掘技术在这样的时代背景下应运而生。聚类分析是探索性数据挖掘的主要任务，也是统计数据分析的常用技术，用于许多领域，包括机器学习、模式识别、图像分析、信息检索、生物信息学、数据压缩和计算机图形学。关于文本聚类的算法有基于网络、层次、划分、密度等算法，也有基于神经网络的算法。

关键词：文本分类 文本聚类

Absrtact: With the rapid development of the information age in the world today, a large number of data information shows an explosive growth trend, and with the progress of the Internet, the speed of transmission of these massive data is also increasing. The processing of these data is an urgent problem to be solved. Text can be transformed into computer-recognizable data through text preprocessing and text representation model, which provides conditions for text classification and text clustering. Chapter 4 summarizes text categorization. According to the process of text categorization, it introduces pre-processing, feature representation, feature extraction, text categorization machine learning algorithm and deep learning algorithm. Machine learning algorithms include Rocchio algorithm, KNN algorithm, decision tree, Naive Bayesian and support vector machine. Deep learning includes fastText, TextCNN, TextRNN and TextRCNN. As an important text mining technology covering data mining, machine learning and natural language processing, text clustering emerges as the times require. Clustering analysis is the main task of exploratory data mining and a common technology of statistical data analysis. It is used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, bioinformatics, data compression and computer graphics. There are some algorithms about text clustering, such as network, hierarchy, partition, density, and so on. There are also some algorithms based on neural network.

Key words: text categorization text clustering

|  |  |
| --- | --- |
| 小组成员 | 分工情况 |
| 巩卫参 | 文本预处理和文本表示模型 |
| 喻云飞 | 文本分类 |
| 熊啸楠 | 文本聚类 |
| 周月阳 | 文本分类和文本聚类的应用 |

# 引言

# 随着科学的日益发展，人们越来越难以离开互联网，人们逐渐习惯了在互联网上工作生活和娱乐，在这个过程中，产生了大量的数据。对这些数据进行分类和聚类，可以更加容易地发现这些数据中蕴藏的价值。文本分类和聚类已成为文本信息处理领域的一个重要分支。利用文本分类和聚类技术，我们可以更加高效的分析和处理文本，文本分类和文本聚类可以有效的解决信息爆炸的问题 。目前，文本分类和文本聚类在情感分析，信息化图书馆等方面都有重要的应用。

# 有越来越多的研究者开始投入到文本分类和聚类的研究中来。文本分类和聚类技术极大的提高了用户查询和检索的效率。通过对文本预处理，去掉文本中无意义的字符，对句子进行分词并将词语还原为它的一般形式，之后进行文本表示，将文本变为计算机可以识别的数据，方便计算机对文本进行操作。在此基础上对文本进行分类和聚类，我们可以从中得到我们希望得到的数据。

# **2文本预处理**

在对文本进行分类和聚类以前,需要对文本进行一系列的处理,提高文本的质量,使分类和聚类的效率更高.在对文本进行与处理之前，我们首先要想办法获取文本，常用的方法有两种，一个方法就是去寻找一些第三方提供的语料库，比如wiki。但事实上，很多情况中所研究或开发的系统往往是应用于某种特定的领域，我们的要求无法在这些开放语料上得到满足。这种时候就需要使用另一种方法，使用爬虫去主动的抓取想要的信息。可以使用如pyspider等python框架可以非常轻松地编写出自己需要的爬虫，从而让机器自动地去获取大量数据，从而继续我们的研究。文本预处理主要包括处理文本标记,分词处理,提取词干,移去停用词4步。

## **2.1处理文本标记**

文本中经常含有一些没有实际意义的标记，例如数字，标点符号。这些标记无法作为分类和聚类提供帮助。所以需要将它们去掉，例如：“今天天气很好！！！”在这句话中，我们需要将后面3个感叹号去掉。

## **2.2分词处理**

分词处理主要是针对中文文档来说，因为在英文文档中，词与词之间有空格和标点符号作为间隔的标志，因此不必对英文文档再进行分词。对于中文文档来说，分词需要一定的算法来实现，目前常用的中文分词方法有3种。

第一种是字符匹配法，这种方法会将待分析的字符串和一个充分大的数据库进行匹配，如果匹配成功，则认为匹配成功的几个字符是一个词语。如果匹配失败，就不会认为这是一个词语。按照扫描方向的不同,可以分为正向最大匹配(从左到右),逆向最大匹配（从右到左）。正向最大匹配就是将一段字符串进行分割，其中分割的长度有限制，然后将分割的子字符串与字典中的词进行匹配，如果匹配成功则进行下一轮匹配，直到所有字符串处理完毕，否则将子字符串从末尾去除一个字，再进行匹配，如此反复。逆向匹配与此类似。例如：对”探索数学的奥秘”这句话进行分词。

假设我们的词典中包含下列词语

探索

数

数学

的

奥秘

正项最大匹配过程：

探索 第一个词匹配成功

数学 第二个词匹配成功

的 第三个词匹配成功

奥秘 第四个词匹配成功。

正向最大匹配的最后结果为

探索 数学 的 奥秘

逆项最大匹配过程：

奥秘 第一个词匹配成功

学的 第二个词匹配失败

的 第二个词匹配成功

数学 第三个词匹配成功。

探索 第四个词匹配成功

逆向最大匹配的最后结果为

探索 数学 的 奥秘

因为中文的一些特性，实际使用时逆向匹配的准确率比正向匹配要高一些。统计结果表明，单纯使用正向最大匹配的错误率为1/169，单纯使用逆向最大匹配的错误率为1/245。但这种精度还远远不能满足实际的需要。实际使用的分词系统，都是把机械分词作为一种初分手段，还需通过利用各种其它的语言信息来进一步提高切分的准确率。

第二种方法是统计法，上下文中，相邻的字同时出现的次数越多，就越可能构成一个词。即在一篇文章中，如果几个字总是按照一定的顺序在一起出现，就认为这几个字组成了一个词。主要统计模型为：N元文法模型（N-gram）、隐马尔科夫模型(Hidden Markov Model, HMM)。N-gram模型基于马尔科夫假设即：假设在一段文本中，第n个词的出现只与前面n-1个词相关，而与其他任何词都不相关。基于这样一种假设，可以评估文本中每一个词出现的概率，整句的概率就是各个词出现概率的乘积。HMM模型用来描述一个含有隐含未知参数的[马尔可夫过程](https://baike.baidu.com/item/%E9%A9%AC%E5%B0%94%E5%8F%AF%E5%A4%AB%E8%BF%87%E7%A8%8B/2952385" \t "https://baike.baidu.com/item/%E9%9A%90%E9%A9%AC%E5%B0%94%E5%8F%AF%E5%A4%AB%E6%A8%A1%E5%9E%8B/_blank)。其难点是从可观察的参数中确定该过程的隐含参数。然后利用这些参数来作进一步的分析。

第三种方法是理解法，让计算机模拟人对句子的理解，从而进行分词。这种方法需要一个非常完备的规则库。其基本思想就是在分词的同时进行句法、[语义分析](https://baike.baidu.com/item/%E8%AF%AD%E4%B9%89%E5%88%86%E6%9E%90" \t "https://baike.baidu.com/item/%E4%B8%AD%E6%96%87%E5%88%86%E8%AF%8D/_blank)，利用句法信息和语义信息来处理歧义现象。它通常包括三个部分：分词子系统、句法语义子系统、总控部分。在总控部分的协调下，分词子系统可以获得有关词、句子等的句法和语义信息来对分词歧义进行判断，即它模拟了人对句子的理解过程。这种分词方法需要使用大量的语言知识和信息。由于汉语语言知识的笼统、复杂性，难以将各种语言信息组织成机器可直接读取的形式，因此目前基于理解的分词系统还处在试验阶段。

## **2.3提取词干**

词干提取是去除[词缀](https://baike.baidu.com/item/%E8%AF%8D%E7%BC%80/6140318" \t "https://baike.baidu.com/item/%E8%AF%8D%E5%B9%B2%E6%8F%90%E5%8F%96/_blank)得到[词根](https://baike.baidu.com/item/%E8%AF%8D%E6%A0%B9/6140303" \t "https://baike.baidu.com/item/%E8%AF%8D%E5%B9%B2%E6%8F%90%E5%8F%96/_blank)的过程。英语中，同一个词通常会有很多种不同的形式，比如动词经常会有一般过去时，现在进行时等时态。名词通常会有单数，复数等形式，词干提取通常就是指将这些不同的形式的单词进行提取，得到词语的词干。1980年7月，马丁·波特提出了一种算法，出版在《程序》杂志上 。该算法被广泛应用，成为英文词干提取中一个事实上的评判标准，被称为“波特词干算法”。Porter 博士因为其在词干提取和信息检索中的成就获得了2000年的托尼·肯特思奖。目前有很多种不同的词干提取方法，这些方法的性能和[准确率](https://baike.baidu.com/item/%E5%87%86%E7%A1%AE%E7%8E%87/5165407" \t "https://baike.baidu.com/item/%E8%AF%8D%E5%B9%B2%E6%8F%90%E5%8F%96/_blank)还有对词干提取中遇到的问题的解决上有所不同。词干提取算法无法达到100%的准确程度，因为语言单词本身的变化存在着许多例外的情况，无法概括到一般的规则中。

1. 查找法。从数据库中查找出相应单词的词尾变化。这种方法实现起来比较简单，但要求数据库中要存储大量的单词和这些单词的变化形式，因此数据库会非常庞大。而且一旦有新词出现，就需要对数据库进行更新。
2. 后缀去除算法。记录一定的规则，通过这些规则来提取词干，例如如果记录了这样一条规则，词尾是“ed”的英文单词需要将“ed”去掉，那么所有以“ed”结尾的单词就会被去除“ed”,这种方法不必存储大量的数据，但是后缀去除算法的解决方案在那些具有大众化词缀但有包含异常的词性上受到了限制。因为不是所有的词类都有这样一个好制定的规则集。
3. 随机算法。在一张词根表上训练有影响的词根形式从而开发出词根可能性模型。这种模型是典型的复杂语言规则，自然相似性等的表达形式。停顿词表现为输入有影响的已经经过训练的词根，根据规则，使用模型产生词根形式。

词干提取：

protecting → protect

protected → protect

protects → protect

protection → protect

和词干提取类似的一个过程是词形还原，词干提取主要是指通过去除前后缀，来获得词语的词干。词形还原主要是指通过变换的形式将词语还原为其一般形式，例如将形容词的比较级和最高级还原成他的一般形式。

词形还原：

Better → good

Best → good

## **2.4移除停用词**

停用词是指在信息检索中，为节省存储空间和提高[搜索](https://baike.baidu.com/item/%E6%90%9C%E7%B4%A2/1806" \t "https://baike.baidu.com/item/%E5%81%9C%E7%94%A8%E8%AF%8D/_blank)效率，在处理[自然语言](https://baike.baidu.com/item/%E8%87%AA%E7%84%B6%E8%AF%AD%E8%A8%80/4146019" \t "https://baike.baidu.com/item/%E5%81%9C%E7%94%A8%E8%AF%8D/_blank)数据（或文本）之前或之后会自动过滤掉某些字或词，这些字或词即被称为Stop Words（停用词）如“a”、“an”、“that”。这些词会对分类没有太大的用处，通常将这一类词语去掉，以提高程序的效率。

# **3文本表示模型**

经过文本预处理，我们还要将文本转化为计算机可以识别的形式，这一步就叫做文本表示。文本表示主要解决文本的表示和计算问题，这里的表示问题是指应该使用什么来作为文本的特征，计算主要是指文本的权重计算和文本之间的相似度应该如何计算常用的文本表示模型有以下几类。

## **3.1 布尔检索模型**

布尔检索模型是一种非常简单的信息检索模型。[布尔](https://baike.baidu.com/item/%E5%B8%83%E5%B0%94" \t "https://baike.baidu.com/item/%E5%B8%83%E5%B0%94%E6%A8%A1%E5%9E%8B/_blank)模型是基于集合论和布尔代数的一种简单检索模型，这种模型会根据查询表达式来返回满足表达式的文档。文档有一组特定词语来进行描述，这组词语的权重只有0和1.这里0代表文档中出现过这个词，1表示这个词没有在这篇文档中出现过。用户的查询表达式为and,or,not连接索引词组形成的布尔表达式。如果用户的查询表达式返回true，则认定当前文档符合用户的要求，将这篇文档返回给用户。例如

如果现在有4个标引项t1,t2,t3,t4。

其中包含t1的文档有d1,d2,d3

其中包含t2的文档有d2,d3

其中包含t3的文档有d2

其中包含t4的文档有d2,d3

假设用户的查询表达式为（t1 And t2）Or(t3 And t4).

此时(t1 And t2)表示用户查询的文档中要同时包含t1和t2两个词语，我们可以看到，目前满足条件的文档有d2,d3。（t3 And t4）表示文档中要同时包含t3和t4两个词语。满足条件的只有d2，因为两个表达式之间使用Or进行连接的，所以d2和d3都是满足用户要求的文档。

布尔检索模型结构简单，易于实现，容易理解，查询速度快。但是无法按照文档与用户查询的相关性进行排序，而且有时难以将用户的需求转化为表达式。只能找到哪些文档是符合用户要求的。为此。有人提出了一种扩展布尔检索模型。在原有布尔模型的基础上对文档向量和用户查询表达式中的标引项进行加权，还允许对布尔表达式中的连接符加权。以实现对文档的相关性进行排序。

## **3.2向量空间模型**

向量空间模型由美国的Salton教授于20世纪70年代提出，已成功地应用于著名的SMART文本检索系统。这个模型对于查询和计算文档与用户查询的相关性具有比较好的操作性。通过将文档和用户查询都表示为一个具体的向量，通过对向量进行一些运算，来表示两个向量之间的相似度，进而表示了文档和用户查询之间的相关性。

对于一个文档集，可以将每一篇文档都用很多个特征项表示，如果将每个特征项都看成是文档空间中的一维，则由这些特征项组成的集合就定义了一个多维向量空间，文档集中的任一文档都可以表示为这个空间的一个向量，则称这个空间为文档空间。

向量空间模型将对文本的处理转化为向量空间中的向量运算。将每一篇文档都按照t个独立属性（s1,s2,s3,...st）标引成一个t维文档向量i=(Xs1,Xs2,Xs3,...Xst)这里的Xin可以理解为（该词条在文档中出现频数的某种函数）。Xij=0表示文档和这个词条毫无关系。Xij=1表示文档和这个词条完全相关。该模型主要涉及到两个问题，一是通过什么方法来构建一个向量来表示文档中的词项，二是如何构建一个向量来表示用户查询的词项，另外还需要选择一种方法来计算两个向量之间的相似度。

针对文档中每个词项的权重，可以考虑两种赋值方法，一种是自动赋值权重，基于词项在整个文档中出现的频率。另一种是人工赋值权重。实验表明，自动赋值在查询性能上并不比人工赋值差。为了给每篇文档建立一个对应的向量，可以考虑如下定义：



在一篇文章中，如果一个词在本文章中出现概率较高，而在其他文档中出现概率比较低，那么我们可以认为这样的词比较能够体现出本篇文章的特征，可以考虑将这类词的权重设的高一些。可以通过



来计算每个词项的权重。

用户查询向量和文档向量之间的相似度计算也是关键的一环。常用的匹配方法有：

1.内积相似度计算（包括余弦相似度）

向量空间模型一般使用查询向量与文档向量的夹角余弦值来计算用户查询和文档间的相似度



2.距离相似度计算

通过计算两个向量之间的距离来作为两个向量的相似度

3.基于项匹配个数的相似度计算

4.基于概率向量的相似度计算

在实现向量空间模型或其他索引模型时，通常使用倒排索引来避免在每篇文章中使用顺序查找来查找查询词。

倒排索引，也常被称为反向索引、置入档案或反向档案，是一种索引方法，被用来存储在全文搜索下某个单词在一个文档或者一组文档中的存储位置的映射。它是文档检索系统中最常用的数据结构。通过倒排索引，可以根据单词快速获取包含这个单词的文档列表。

|  |  |
| --- | --- |
| 词项 | 文档和频率 |
|  | (,1)→(,2) |
|  |  |
| .............. |  |
|  | (,) |
| ............... |  |
|  |  |

表3-1 倒排索引表

从表1可以基本看出倒排索引表的结构。对于每一个词项，指向的逻辑连接表被称作倒排表。倒排表包含所有包含这个词项的文档和这个词项在该文档中出现的频率等信息。在表1中，包含term1的文档有文档1和文档10，并且在文档1中出现了1次，在文档10中出现了2次。

向量空间模型可以通过计算向量之间的相似度来计算文档之间的相似度，因此可以通过相似度来对文档的相关性进行排序。但是向量空间模型会赋予那些在用词规律和查询类似的文档较高的相关性，而且需要注意的是，向量空间模型每个向量空间模型的每个词条间要尽可能相互独立。但是实际使用过程中各个词条之间可能会或多或少的有一定的相关性。基于向量空间模型的检索系统一般比较复杂。大型商业检索系统很少使用这种模型。

## **3.3 概率检索模型**

对于某个特定的提问，文献集中的某一篇文章是否是相关文章可以看做是一个随机事件，每个文献是否是相关文献的概率各不相同。

概率检索模型基于对已有反馈结果的分析，根据贝叶斯原理为当前查询排序。概率检索模型与贝叶斯分类的思想非常接近，但还是有一定区别的。

贝叶斯分类原理：对于给出的待分类项，求解在此项出现的条件下各个类别出现的概率，哪个最大，就认为此待分类项属于哪个类别。

朴素贝叶斯分类的正式定义如下：

1、设x=\{a_1,a_2,...,a_m\}为一个待分类项，而每个a为x的一个特征属性。

2、有类别集合

C=\{y_1,y_2,...,y_n\}。

3、计算

P(y_1|x),P(y_2|x),...,P(y_n|x)。

4.如果

P(y_k|x)=max\{P(y_1|x),P(y_2|x),...,P(y_n|x)\}，则x \in y_k。

概率检索模型并不是用来分类的，而是用来计算某个文档属于属于“相关”或者“不相关”的概率大小为文档排序。

概率检索的目的是估计P（R/q,d）即文献d对于用户查询表达式来说被认为是相关文献的概率。一种简单的做法是每一篇文献都根据文档中是否包含标引词将文献表示为一个二值向量d={d1,d2,d3,...},di=0表示文档中没有出现过这个词语，di=1表示文档中出现过这个词语，用R表示当前文档为相关性文档，用表示当前文档为不相关文档，对一篇文献计算P（R/q）和P（/q）来判断哪个文献是相关文献，哪个是不相关文献。

为了决定文献相关的阈值，需要设置一个判断规则，最简单的判断规则是如果P(R/q)>P(/q),则认为文档是相关的，否则就认为文档是不相关的。检索结果按照相关型大小来排序。

概率检索模型需要解决的两个基本问题：参数估计和独立性假设。

参数估计需要良好的训练数据集，没有良好的数据集，很难估计参数。独立性假设并不完全符合实际情况。

概率论模型的基本准则是：文本按照与查询的概率相关性大小排序，排在前面的文本是最有可能被获取的文本，即如果文本满足如下公式则该文本被获取：

P(Rel|Doc)>=P(Notrel|Doc)

其中，p(Rel | Doc)表示文本Doc与查询有关的条件概率；p(Notrel | Doc)表示文本Doc与查询不相关的[条件概率](https://wiki.mbalib.com/wiki/%E6%9D%A1%E4%BB%B6%E6%A6%82%E7%8E%87" \o "条件概率)。根据[贝叶斯规则](https://wiki.mbalib.com/wiki/%E8%B4%9D%E5%8F%B6%E6%96%AF%E8%A7%84%E5%88%99" \o "贝叶斯规则)，上述公式可以改写成



所获取的文本可以利用上述公式的左端进行排序。

利用概率论模型的关键是如何计算左端式子中的概率。如果查询由一个词组成，或者组成查询的词之间相互独立，那么查询词的[权重](https://wiki.mbalib.com/wiki/%E6%9D%83%E9%87%8D" \o "权重)可以由下式计算：



式中，N是文本库中文本的数量；n是文本库中包含该词的文本数量；R是与查询相关的文本数目(相关的文本数目在训练文本库中是已知的)；r是与查询相关而且包含该词的文本数目。这样查询q可以表示为IMG_259，其中，wi由上述公式计算得到。文本d表示为IMG_260，其中，如果词存在于文本中，则对应的xi = 1，反之则xi = 0。查询q与文本d之间的相似度函数定义为



利用概率论模型的典型系统有Kwok与Robertson等，它们在利用样本计算词权重的过程中考虑了词在文档中出现的[频率](https://wiki.mbalib.com/wiki/%E9%A2%91%E7%8E%87" \o "频率)，其中Kwok系统还考虑了词在整个样本集合中出现的频率。概率论模型的效果要明显优于[布尔模型](https://wiki.mbalib.com/wiki/%E5%B8%83%E5%B0%94%E6%A8%A1%E5%9E%8B" \o "布尔模型)，但比[向量空间模型](https://wiki.mbalib.com/wiki/%E5%90%91%E9%87%8F%E7%A9%BA%E9%97%B4%E6%A8%A1%E5%9E%8B" \o "向量空间模型)略差。

概率论模型的一个特例是[贝叶斯网络](https://wiki.mbalib.com/wiki/%E8%B4%9D%E5%8F%B6%E6%96%AF%E7%BD%91%E7%BB%9C" \o "贝叶斯网络)，由于该模型适合于超文本系统，因此在超文本信息成为当前[信息获取](https://wiki.mbalib.com/wiki/%E4%BF%A1%E6%81%AF%E8%8E%B7%E5%8F%96" \o "信息获取)主流信息的情况下，该模型的应用越来越广泛。

## **3.4语言模型**

一个语言模型通常构建为字符串s的概率分布p(s)，这里的p(s)实际上反映的是s作为一个句子出现的概率。统计语言模型是一个单词序列上的概率分布，对于一个给定长度为m的序列，它可以为整个序列产生一个概率 P(w\_1,w\_2,…,w\_m) 。其实就是想办法找到一个概率分布，它可以表示任意一个句子或序列出现的概率。

语言模型目前在自然语言处理相关应用非常广泛，如语音识别(speech recognition) , 机器翻译(machine translation), 词性标注(part-of-speech tagging), 句法分析(parsing)等。传统方法主要是基于统计学模型，最近几年基于神经网络的语言模型也越来越成熟。

语言模型的基本思想为通过计算文章生成查询的可能性大小，来对文章的相关性进行排序。最简单的语言模型是一元语言模型，它只需要每个词的分布；更复杂的语言模型则需要更多的上下文信息来预测一个词。

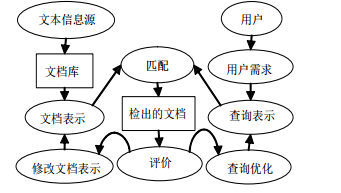


图3-1 语言模型应用于信息检索的一般图形表示

建立文档或查询模型，首先需要选择建模对象，然后对选择好的对象建立模型的基本框架，最后选择合适的方法进行平滑(用来调整模型，使模型能尽可能表示文档或查询的方法)优化检索。

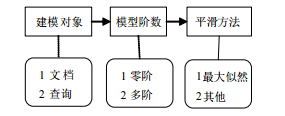


图3-2 建模过程

选择建模对象需要考虑是对文档建模，对用户的查询建模，还是对两者都建模。

模型可以根据马尔科夫链的阶数分为一元语言模型和多元语言模型。

一元语言模型假设词与词之间是相互独立的，一个词出现的概率与这个词前面的词没有存在必然联系。这是最简单的语言模型。

多元语言模型(其中N>=2)假设词与词之间是相互关联的，一个词出现的概率与这个词前面的词存在一定的关联。根据与前面词的个数的多少，把多元语言模型分为二元语言模型、三元语言模型等。

文献[1]中Song 和Croft指出，把一元语言模型和二元语言模型混合后的效果比只使用一元语言模型则好 8%左右。不过，VictoLavrenko中指出，Song和Croft使用的多元模型得到的效果并不是一直比只用一元语言模型好。同样，David R.H.Miller 指出一元语言模型和二元语言模型混合后得到的效果也要好于一元语言模型。

为了解决某些词项在文档中可能没有出现，此时概率为0的问题，我们需要进行平滑处理，所谓平滑就是概率为零时，选择一个值来代替0.常用的方法有最大似然估计法，播入方法中的 Jelinek-Mercer，Dirichlet，absolute discounting，二阶段平滑方法以及其它模型中的翻译模型以及 PLSI 方法，也有少数采用了 Good-Turing 方法。

考虑朗读一篇文章，如果说话人读取了文档中的所有词语，那么他说出查询中的词语的概率是多少呢？在形式上，相似度可以简单的定义为



这里需要为查询构造一个概率模型。一种方法是对任何词项的出现与否使用伯努利事件进行建模。此时可以将生成整个查询看做观察到查询中的词项和没有观察到查询外的词项的联合事件。生成查询的概率为所有查询中出现的词项和没有出现的词项的概率的乘积。

可以通过多种方式来估计模型中的 一种方法是





对于查询中没有出现的词项，使用平滑因子。如果该词在查询中存在，则在小规模文档模型上估计w的值，并引入风险函数作为混合参数。词项权重计算公式为

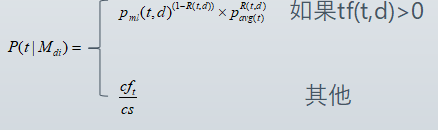


图3-3 词项权重计算公式

对于一个由T个词按顺序构成的句子IMG_256，p(s)实际上求解的是字符串IMG_257的联合概率，利用贝叶斯公式，链式分解如下：



从上面可以看到，一个统计语言模型可以表示成，给定前面的的词，求后面一个词出现的条件概率。

我们在求p(s)时实际上就已经建立了一个模型，这里的p(\*)就是模型的参数，如果这些参数已经求解得到，那么很容易就能够得到字符串s的概率。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 特征项权重 | 结构化查询 | 语义扩展 | 匹配方法 | 相关性反馈 | 其他 |
| 布尔模型 | 0或1 | 显式  And,Or,Not |  | 精确匹配  BIR |  | 结构化查询 |
| 向量空间模型 | 实验获得  标准化 | 隐含  And | 字典  聚集分析 | 模糊匹配  基于距离 | 权重更新  Rocchio方法 | 特征项权重 |
| 传统概率模型 | 相关性文档估计  获得困难 | 隐含  And |  | 模糊匹配  基于概率 | 重新估计权重 | 相关性反馈 |
| 语言模型 | 直接估计  容易获得  不需要要标准化 | 隐含  And,Or,Not | 查询语言模型 | 模糊匹配  基于概率分布或Bayes决策 | 查询语言模型 | 综合框架 |

表3-2几种文本表示模型总结

## **3.5几种文本表示模型的总结**

布尔模型，向量空间模型，概率模型，语言模型各有优缺点，使用时应该就具体情况进行分析和比较，再决定使用哪个模型。

## **3.6文本特征选择方法**

从文档中提取的特征数量比较大，有必要对提取到的文档特征进行筛选，选出那些最能代表文档类别概

念的特征。这一过程就是特征选择。

1.文档属性选择方法基本上是基于信息论和统计分析。特征选择的具体步骤为

2.从训练文档库中提取所有得到的特正项，构成文档特征集合F;

3.对集合F中的每一项用某一种方法进行打分。

假设需要选取N个文档分类属性，则从F中的选取分值高的N个项，构成最终的分类属性集FS.

文本中能观察到的量其实只有两个：词频和文档频率，所有的方法一律以这两个量为计算基础。简单综合这两者的TF-IDF选择出来的特征不具有类别区分度。以文档频率为基础的特征选择算法有文档频次方法（直接依据文档频率大小排序的方法）、卡方检验、信息增益、互信息等。

# 4文本分类综述

## **4.1文本分类概述**

随着互联网发展的不断壮大，人们上网的途径也变得越来越多样，中国网民的数量不断攀升，根据中国互联网络信息中心在今年8月份发布的报告中显示，中国网民已达到8亿人次。而网络中无时无刻存在着数据传输。

在如此庞大的数据流量下，包含了大量的文本信息，在大量文本的背后包含了许多隐含且值得研究的问题，比如，网络热点、网络舆情。此类信息已与人们日常生活息息相关，研究该类问题意义重大。然而，由于网络信息量过于庞大，仅仅依靠人力来分类识别各类文件信息变得不切实际。因此，文本分类将能辅助人们进行管理庞大的文本信息，使其更能快速精准的获取自身所关注的问题。

文本分类主要包含了对模型进行训练的过程以及对新文本进行分类的过程:



图4.1 文本分类训练过程

在训练过程中，主要分为以下五步:

1）训练集的获取。

2）对文本信息进行预处理。

3）对文本经行表示。

4）对高维文本信息经行降维，实现特征提取。

5）采用机器学习算法或深度学习算法训练模型。

文本分类的分类过程如下:



图4.2 文本分类分类过程

当对待分类样本进行特征表示后，投入相应的分类器，在分类器训练得足够好的情况下，将得到分类结果。

## **4.2文本预处理**

所谓文本预处理其主要目的就是对文本进行一系列操作，比如，分词、处理文本标记、去掉停用词等，以此来减少阻碍分类的干扰因素，提高分类准确率。

对于文本所用语言的不同，还可以将预处理流程分为英文和中文两部分。

### 4.2.1中文文本预处理

1）除去数据中非文本部分。

由于训练集有可能来源于我们通过爬虫爬取的语料数据，其中必然包含了一些对文本分类无关的html标签，因此需要去掉。可以使用python中的正则表达式(re)来删除较少量的非文本内容，也可以使用其他的工具，比如[beautifulsoup](http://link.zhihu.com/?target=http%3A//www.crummy.com/software/BeautifulSoup/)来辅助进行。

2）处理中文中的编码问题

一般来说存储数据使用utf8，而使用中文数据进行相关处理时，则使用GBK进行中文编码。

3）中文分词

中文分词简单来讲就是将一个句子分成一个个的汉字或单词。中国汉字博大精深，对于一个较为复杂多义的句子，有时候人也很难进行分词划分，对于计算机来说则更加困难。

比如:

研究生命的起源。

如果按照以下进行划分:

研究\生命\的\起源

它表明这句话的意义是在研究某个生命的起源。

而如果按照以下进行划分:

研究生\命\的\起源

它的意思是研究生的生命的起源。

可见，对于同一个句子断句不同，其意义也可能不同。

分词方法主要有以下几种:

1）双向最大匹配法

2）基于字标注的平均感知机分词方法

3）统计与字典相结合

4）基于深度学习的分词方法

现在分词系统中较为流行的有张华平老师的汉语词法分析系统，即ICTCLAS分词系统和jieba分词等。

4）去掉停用词

对于中文中一些出现频率较高但没有实际意义的语气词，如”啊”、”吧”、”和”等，以及一些标点符号的引用，需要我们进行去掉。

5）词语替换

主要是针对将一些方言和网络用语替换成大家常用的语言，比如将”俺们”替换成”我们”、将”禾斗匕匕”替换成”科比”、将”灰机”替换成”飞机”等等。

### 4.2.2英文文本预处理

1）除去数据中非文本部分。

同中文一样，由于训练集可能来源于爬虫爬其，对于其中的非文本部分，如html标签，需要去掉。可以使用python中的正则表达式(re)来删除较少量的非文本内容，也可以使用其他的工具，比如[beautifulsoup](http://link.zhihu.com/?target=http%3A//www.crummy.com/software/BeautifulSoup/)来辅助进行。

2）拼写检查

由于英文中会出现一些拼写错误，例如将”friend”写成”friand”,因此需要我们在预处理时找到其中的错误拼写，并进行相应的正确拼写。

3）词干提取和词形还原

词干提取和词型还原在英文文本与处理中扮演着重要作用。两者看上去很相似，即都是要找到词的原始形式。但是词干提取的容错率较高，它在还原词干的时候可能会得到拼写错误的词。比如"taking"的词干可能得到的是"tak", 并不是一个词。而词形还原容错率较低，他所还原的词最后得到的都是一个拼写正确的词。

4）大写转小写

对于英文中的大小写词在分析时可能会识别成两种不同的词，尽管它们意思相同，因此，需要在文本预处理阶段将大写词变为小写词。

5）去掉停用词

同中文中一样，英文也包含了很多无意义或电脑无法识别的词，比如，”a”、”an”、”the”

等等以及一些标点符号。

## **4.3文本表示**

文本表示的目的是把文本预处理后的文本转换成计算机可理解的方式，良好的文本表示模型将使文本更好的从文本空间映射到向量空间，将文本向量化表示后从而使得文本可计算化。文本表示的选择很大程度上将决定最终分类质量的高低，因此，文本表示是决定文本分类质量结果因素中很重要的一部分。

### 4.3.1词袋模型

词袋模型是最基础的文本表示模型，它忽略每个词出现的顺序，把待分类文本看作一袋子单词。

词袋模型相对比较简单，然而由于其忽略了上下文之间的关系，其表示准确度不算太高。比如以下两个句子:

“我/喜欢/踢/足球”

“我/不/喜欢/踢/足球”

可以看出，第一句话和第二句话中的单词基本相同，将其表示在向量空间上后，两个向量仅在某一个维度上不同，因此，会认为这两句话是相似的。然后，这两句文本严重不相似。

词袋模型使用较广泛，但还是存在一些缺点:

1. 由于大部分文章主要只含有1000个词左右，而语料库却能达到10万，映射到向量就是10万维向量中只有1000个维度有值，导致矩阵高纬度、高稀疏度。
2. 将单词看作词袋，忽略上下文的关系，致使无法解读一些词在上下文关系中的真正含义。
3. 不考虑词与词之间的关系，对于一些同类词不能识别。

### 4.3.2 N-gram模型

N-gram又被称为一阶马尔科夫链。它的主要思想是将字符串看作字符流，对其用滑动窗口的方式进行截取操作，形成长度为N的字节片段序列，称之为gram，然后再统计各个gram的出现频率，形成文本的向量特征空间。各个gram就相当于词袋模型中的单词，在向量特征空间中就是特征向量维度。

N-gram的优点有如下几点:

1. 由于它将字符串当作字符流进行处理，因此在处理中文和英文时误差差别较词袋模型误差差别有了明显改善。
2. 提高容错率。
3. 不需进行语言学处理。

### 4.3.3 [Word2Vec](https://www.cnblogs.com/june0507/p/7600988.html)

针对N-gram的不足，Bengio等人引入了前向反馈神经网络来拟合词序列的条件概率，即Neural Network Language Model。

NNLM模型采用了以下的思想，其通过词表中的每一个word都对应着一个连续的特征向量，并且假定输入词向量序列后能得到这段序列的联合概率。以此来计算词向量的权重和概率模型的参数。

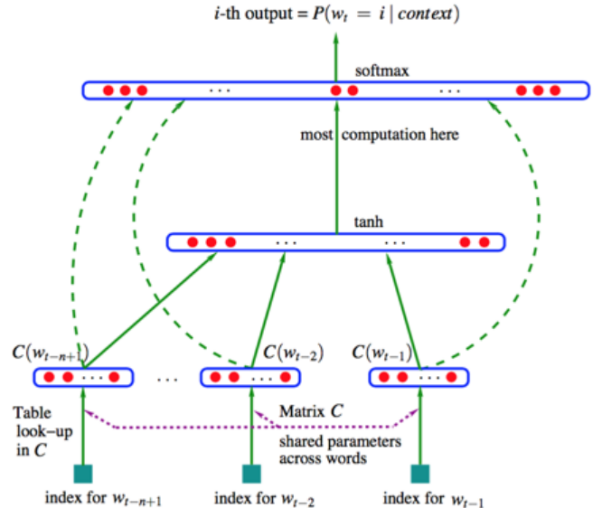


图4.3 NNLM模型网格结构

NNLM模型也存在一些问题。比如只能处理定长的序列，缺少灵活性。并且训练时间过长，对于真实语料库，要想训练完成基本不可能。

Word2VEC是由谷歌在2013年开源的用于计算词向量的工具，它结合了深度学习的思想，采用三层神经网络，即输入层-发射层-输出层，将词作为特征，把文本映射到高维空间向量。

在此处，[Word2Vec](https://www.cnblogs.com/june0507/p/7600988.html)指的是其背后用于计算word vector的CBOW模型和Skip-gram模型。

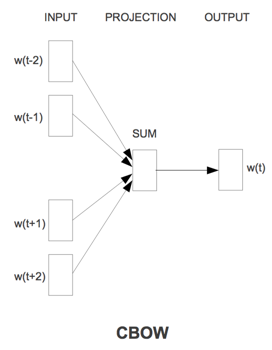


图4.4 CBOW模型

CBOW模型将NNLM模型做了如下改造:

1. 移除非线性的hidden layer，对中间层embedding layer与输出层softmax layer连接。
2. 忽略上下文的序列信息，并将 furture words加入上下文。

Skip-gram模型是计算输入向量与输出向量之间的余弦相似度，再进行softmax归一化。

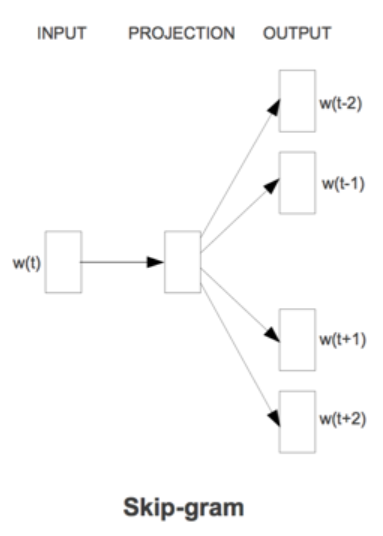


图4.5 Skip-gram模型

## **4.4特征提取**

特征提取的目的主要是针对上一步对文本表示后的优化工作，由于通过前面表示后的向量维度过高，计算量过大，所以特征提取的目的就是降低特征向量的维度以及筛选出和类关系性大的特征来辅助提高分类的准确率。

其方法思想总结来看如下:依靠下文所介绍的几种特征评价方法来构造特征评价函数，紧接着对特征表示后的原始向量的各个特征进行独立评估，由此，每个独立特征都获得一个属于它自身的评估分值，然后对所有的特征按照其评估分值的大小进行排序，选出排名靠前的特征、即最能代表该文本信息的特征。

### 4.4.1词频 (Term Frequency, TF)

词频方法比较简单，它仅仅考虑以某个词在该训练集中的次数为依据来对特性进行打分。

词频公式如下：

其中，w为考察词，n为w在语料中的次数，N为语料中总的次数。

可以看出，词频方法的优点就是简单，复杂度较低，随着训练集增加而线性增加。

尽管优点突出，但缺点也很明显，它仅仅考虑此出现的频率，而忽略了类别和特征之间的关系。比如，某些没有意义的高频词以及一些尽管频率较低，但却有实际意义，对分类很重要的专有名词。

### 4.4.2词频-逆文件频率 (Term Frequency-Inverse DocumentFrequency, TF-IDF)

针对上述词频缺点的改进，引入权重的计算，当文本词出现在很多的类别下时, 该值很小;反之, 当该文本词出现在较少的类别下时, 该值较大。

其计算公式如下:

其中， 是该次在文档中的出现次数，分母则是文档中所有字词的出现次数之和。

语料库中文件总数， 为包含词的文档数目

每个词的权值为:

=

### 4.4.3卡方检验（CHI）

卡方检验源自于统计学的方法，统计样本的实际观测值与理论推断值之间的偏离程度，实际观测值与理论推断值之间的偏离程度就决定卡方值的大小，卡方值越大，越不符合；卡方值越小，偏差越小，越趋于符合，若两个值完全相等时，卡方值就为0，表明理论值完全符合。

卡方检验公式如下:

其中，E 表示理论值，x 表示实际观测值，上式计算出了总的偏差程度，并设置一个阈值，小于某个值就认为假设成立，两个变量不相关；反之，大于了预设阈值，认为假设不成立，表示两个变量相关。

在特征提取中，用特征T与类别C不相关来做原假设，则的统计值为:

N表示样本集中文档总数，A表示每个词的正文档出现频率，B表示每个词出现的负文档频率，C表示正文档不出现的频率，D 表示负文档不出现频率。

### 4.4.4互信息 (MI)

在文本特征提取方法中，互信息通过判断特征项与类别的关联程度来设计评价标在特征提取中，互信息常用公式如下所示:

其中，指特征词 w 在整个文本训练集中出现的概率，P()指类别Ci在整个文本集中出现的概率；P ( w,)代表文本类别中出现特征项 w 的文本数目在整个训练集中的比重。

## **4.5早期文本分类方法**

在得到所需要的特征以后，就是文本分类中最为重要的一步，即分类器的训练。文本分类类似于其他分类。分类不同于聚类，分类对于待分类物体有明确的标签，即样本属于A类或者B类或者其它类。

早期的文本分类主要有两种方法:词匹配法和知识工程法。

词匹配法：

词匹配法，仅仅根据文档中是否出现了与类名相同的词或与类名意义接近的词，以此来确定该文本是否属于这一类。

知识工程：

在词匹配法的基础上提出知识工程，知识工程需要依赖各种专业人士的帮助，为每个类别设定相应的推到规则。当该文本符合或大部分符合这项规则时，可判定该文本属于这一类别。

知识工程一定程度上来讲较词匹配法大幅度提高了分类的准确度。但缺点也更加突出，比如由于需要提前设定推导规则，因此将耗费大量的人力和物力。还有一个最致命的缺点就是知识工程对于不同领域的文本分类没有可推导性。例如，专家已经针对某数学领域创建了分类系统，当需要对其他领域，如政治领域经行分类时。之前所创建的系统将不再适用，因此需要重新再创建，浪费了之前的成果。

## **4.6机器学习方法**

### 4.6.1 Rocchio算法

Rocchio算法的基本思想使用训练集为每个类构造一个原型向量。对于某些词一旦出现属于这个分类的可能性就会增加,而另一些词一旦出现属于这个分类的可能性就会降低,那么累计这些正面的,和负面的影响因素,最后由文档分离出的词向量可以得到对于每个类的一个打分,打分越高属于该类的可能性就越大。



图4.6 Rocchio算法流程

1. 计算每类文本集的原型向量，计算方法为训练集中所有文本的算数平均。
2. 文本到来后经过预处理，然后将文本表示为特征向量。

3）计算新文本特征向量与每个类原型向量的相似度。

4）比较每个类原型向量与新文本向量的相似度，将文本分到相似度最大的文本类别中。

在文本分类中，Rocchio算法基本的思路是把一个类别里的样本文档各项取个平均值，可以得到一个新的向量，称为“质心”，质心就成了这个类别最具代表性的向量表示。再有新文档需要判断的时候，比较新文档和质心相似度。

稍微改进一点的Rocchio算法不仅考虑属于这个类别的文档（称为正样本），也考虑不属于这个类别的文档数据（称为负样本），计算出来的质心尽量靠近正样本同时尽量远离负样本。

优点：易于实现、训练和分类简单、分类时间效率高。

缺点：精度一般、不能用在非线性空间的分类上。

### 4.6.2 KNN算法

KNN算法的基本思想是每个样本都可以用它的 Ｋ 个邻居样本来代表，通过相似度的计算来找到与待分类样本相似度最大的 Ｋ 个邻居样本，然后根据这Ｋ个邻居样本所属的类别采用投票法来对待分类样本所属的类别进行判定。

训练样本集中每个数据都存在标签，即我们知道样本集中每一数据与所属分类的对应关系。输入没有标签的新数据后，将新数据的每个特征与样本集中数据对应的特征进行比较，然后算法提取样本集中特征最相似数据（最近邻）的分类标签。一般来说，我们只选择样本数据集中前k个最相似的数据，这就是k-近邻算法中k的出处，通常k是不大于20的整数。最后，选择k个最相似数据中出现次数最多的分类，作为新数据的分类。

如果待分类样本的 Ｋ 个邻居样本中的大多数样本属于某个类别，则判定待分类样本也属于这个类别。



图4.7 KNN算法计算流程

其中，余弦相似度计算公式为：

待分类样本ｄ属于每个类别的权重：

其中，为训练集中某一个文本，表示向量中第ｊ维特征值的权重，d为待分类向量，表示向量ｄ的第ｊ维特征值的权重。

函数ｙ( , )为 类 别 属 性 函 数，属于 其值为1，否则为0。

其优点如下:

1）简单有效

2）重新训练代价低

3）可用于非线性分类

4）准确度高，对数据没有假设，对outlier不敏感

5）适合类域的交叉或重叠较多的待分样本集

其缺点如下:

1）计算量大

2）需要大量内存

3）样本类别不平衡

### 4.6.3决策树算法

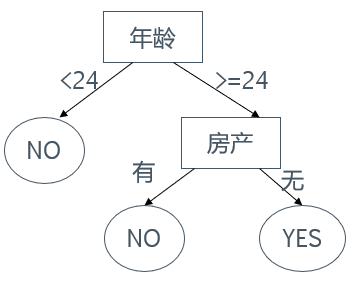


图4.8 决策树示例

决策树如图3.8所示，以某银行调查人群贷款意向为例。将人群按照各种已经制定好的标签进行判断，判断成立则进入该标签，再以此进行判断，直到得到结果。可以看出，上图中的决策树包含了3条信息。

1. 对于年龄小于24的客户，银行会认为他们没有贷款意向。
2. 对于年龄大于等于24且有房产的客户，银行会认为他们没有贷款意向。
3. 对于年龄大于等于24且没有房产的客户，银行会认为他们有贷款意向。

可以看出，决策树的构建在分类过程中起到了关键作用。其构建过程是由浅入深，首先选取最具有代表性的属性作为节点，然后按属性值进行划分，递归构造树形结构。再对这个树进行剪枝，直到到达合适的规模和分类效率。通常，使用信息增益来进行划分属性选择。

信息增益源于信息熵，对于特征T，其信息增益为:

T为特征，C为分类类别，H为熵

决策树最大的优点就是结构清晰，容易理解。在构建树的过程中如果出现问题很容易调试，在相对短的时间内能够对大型数据源做出可行且效果良好的结果。

决策树通常在训练集上表现出很好的分类效率，在待测分类中效率往往不好。因此，模型容易出现过拟合问题，要想提高准确度，需要比较好的简直策略。

### 4.6.4朴素贝叶斯算法

朴素贝叶斯分类方法是源于贝叶斯理论，给定待分类文本，计算其最大后验概率，值越高，说明该文本越有可能属于这个类别。

贝叶斯公式如下:

其中，是指文本属于类别的后验概率，表示先验概率。

朴素贝叶斯算法分类的效率比较稳定，但是只适用于小规模的数据并且假设各个特征值之间相互独立。

朴素贝叶斯算法优点如下:

1）所需参数少

2）对缺失数据不敏感

3）算法简单

4）源于古典数学理论，有着坚实的数学基础，以及稳定的分类效率

朴素贝叶斯算法优点如下:

1）需要知道先验概率

2）模型假设属性之间相互独立，这个假设在实际应用中往往是不成立的

3）在属性个数比较多或者属性之间相关性较大时，NBC模型的分类效率比不上决策树模型。而在属性相关性较小时，NBC模型的性能最为良好

### 4.6.5支持向量机算法

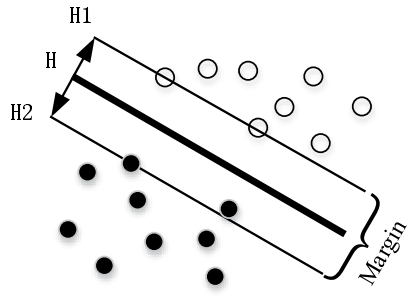


图4.9 SVM示意图

支持向量机的示意图如上所示，它分类的思想是在不同的类之间能找出最大分类的超平面。如上图中的H线，用来区分两个类。用方程Y=AX+B来表征该超平面方程。其中X为特征向量，A为系数向量，B为常量。SVM是一种有监督的学习模型，在处理二分类问题上可以说是现有算法中的最好的一种。

文献[4] 基于基本支持向量机算法，提出了一种新的文本分类算法。结合SVM算法和KNN算法，提出了SVM-KNN文本分类算法。SVM-KNN算法可以通过反馈和改进分类预测概率来提高分类器的性能。测试了SVM-KNN算法的实际效果，并在相关的中文网页分类测试系统中验证了性能。

文献[5]探讨了内核支持向量机在文本消息领域的应用。基于训练文本消息集中存在的使用模式对文本消息的作者进行分类。在确定未知样本的作者时，将准确率从57％提高到了96％。

SVM优点如下:

1）可以解决小样本分类问题。

2）可以提高泛化性能。

3）可以解决高维问题。

4）可以解决非线性问题。

5）可以避免神经网络结构选择和局部极小点问题。

SVM缺点如下:

1）对缺失数据敏感。

2）对非线性问题没有通用解决方案。

3）对缺失数据敏感。

4）对非线性问题没有通用解决方案。

## **4.7深度学习方法**

### 4.7.1 fastText

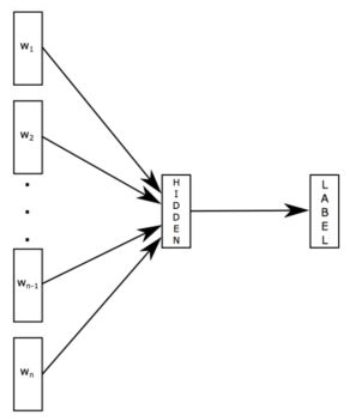


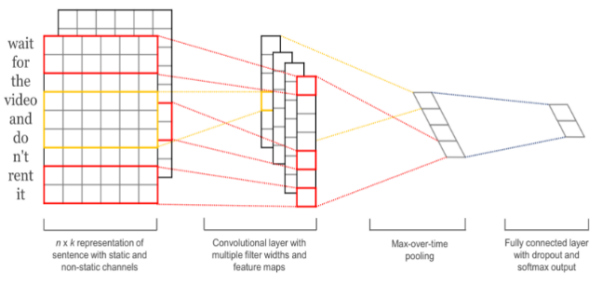
图4.10 fastText示意图

文献[6]对fastText进行了介绍。其主要思想是将文本中的单词进行lookup得到词向量，求向量平均，然后直接接 softmax 层预测标签。当标签较多的时，论文最后一层采用了层次softmax的方法，既根据label的频次建立哈夫曼树，每个标签对应一个哈夫曼编码，每个哈夫曼树节点具有一个向量作为参数进行更新，预测的时候隐层输出与每个哈夫曼树节点向量做点乘，根据结果决定向左右哪个方向移动，最终落到某个标签对应的节点上，以此减少计算量。

举个例子，fastText能够学会”医生”、”工人”、”白领”、”教师”指代的是特定的工种。并且能够将这些数值存在相关文档中。然后，当用在提出一个请求（假设是“我的孩子的老师是谁？”），它能够马上在fastText生成的文档中进行查找并且理解用户想要问的是有关教师的问题。

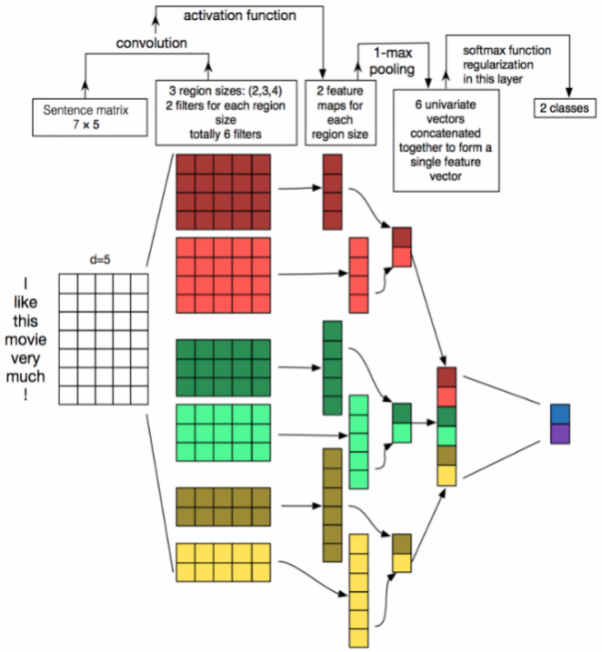
fastText的特点就是快，训练速度比其他方法快一个量级。

### 4.7.2 TextCNN

图4.11 TextCNN示意图

文献[7]中提出了TextCNN算法，首先对句子经行截断，使句子的长度固定为n，然后将单词embedding成固定m维度向量，这样句子表示成(n,m)(n,m)大小的矩阵，就像上图中的像素。然后经过一维卷积层，得到输出channel。第三层使1-max pooling层，使不同长度句子变成定长表示，最后再接softmax层，得到其分类的概率。

原理如下：

图4.12 TextCNN详细原理图

### 4.7.3 TextRNN

TextRNN利用CNN进行文本分类，其实还是运用卷积核去寻找n-gram的特征。卷积核的大小是超参。利用RNN去处理文本信息，因为RNN对处理时间序列有优势，它利用前后不同时刻的输出链接来确保了“记忆”的留存。但RNN循环的机制和形式过于简单从而导致梯度消失和梯度爆炸。RNN的变种LSTM/GRU在一定程度上减缓了梯度消失和梯度爆炸问题。

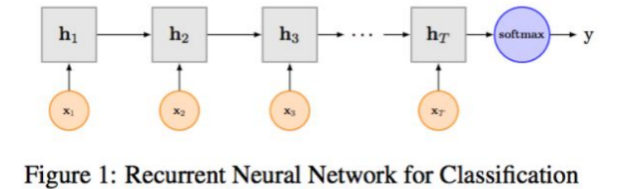
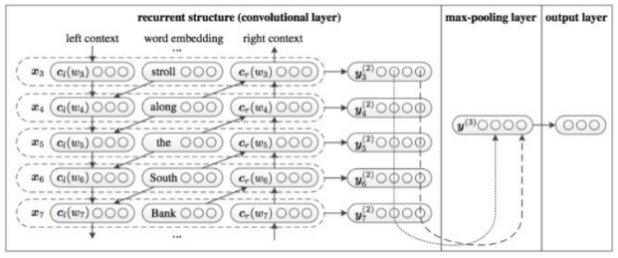


图4.13 TextRNN示意图

利用RNN进行文本分类时，将句子的每个字或者词对应于RNN的一个时刻，隐层的输出作为下一时刻的输入，最后用隐层输出捕获句子的抽象特征，再用softmax进行分类。

### 4.7.4 TextRCNN

图4.14 TextRCNN 示意图

文献[8]提出了TextRCNN算法。利用前向和后向的RNN来算出每个词的前向和后向的上下文的表示。

将词的表示变成词向量和前后向上下文向量连接起来的形式。

最后接和TextCNN相同的卷积层，pooling层即可。

## 4.8评估标准

作如下约定:

1) a表示正确地标注测试集文本为类别ci 的文本数量；  
  2) b表示错误地标注测试集文本为类别ci 的文本数量；  
  3) c表示错误地排除测试集文本在类别ci 之外的文本数量；  
  4) d表示正确地排除测试集文本在类别ci 之外的文本数量。

模型在类别上的召回率:

模型在类别上的准确率:

模型在类别*ci* 上的F1值定义如式:

# **5文本聚类**

文本聚类是将无类别的文本数据聚合成若干个组或类别的过程，与文本分类不同，没有明显监督信息或分类标准，是无监督无指导的聚簇过程，数据处理事先未知聚簇结果为几个类别，不需要训练样本，完全根据样本的距离计算或者其他计算标准自动形成几个簇，一个簇内的文本数据具有较强的相似性，而与其他簇内的文本数据没有或者具有很小的相似性。

聚类分析是基于相似性，将一组对象进行分组，使同一组中的对象彼此（在某种意义上）比其他组中的对象更相似。

聚类分析本身不是一种特定的算法，而是需要解决的任务。可以通过各种算法来实现，这些算法在理解集群的组成以及如何有效地找到集群方面存在显著差异。集群的流行概念包括集群成员之间的距离较小的组、数据空间的密集区域、间隔或特定的统计分布。因此，聚类可以被描述为多目标优化问题。适当的聚类算法和参数设置（包括参数，例如要使用的距离函数、密度阈值或期望聚类的数量）取决于单个数据集以及结果的预期使用。

## **5.1聚类算法**

聚类是数据分析的核心构建块，旨在从原始数据集（例如可以有效地关联、比较和解释的特定组）中提取其他隐藏的结构和关系。迄今为止，已经提出了大量的可视化交互式聚类分析技术，然而，要获得有用的聚类通常需要几轮的用户交互来微调数据预处理和算法。按照逻辑划分的聚类算法包括基于网格、基于层次、基于划分、基于密度等，而还有基于神经网络的SOM法。

聚类分析首先也是要将文本通过预处理，将所有文本的单词个数作为矩阵的列数，文本的数目作为矩阵的行数，使用TF-IDF作为每个单词在其所在文本的权值，形成的矩阵通过转置，每一列的列向量就是每一个文本的特征，即转为VSM向量空间模型。

### 5.1.1基于网络的算法

基于网络的聚类算法典型特点是将处理对象由原始数据点转化为自行划分的网格单元，将数据空间划分成为有限个单元（cell）的网格结构,所有的处理都是以单个的单元为对象的。基于网络的算法有STING算法、CLIQUE算法、WAVE-CLUSTER算法。

STING是一种基于网格的多分辨率的聚类技术，它将输入对象的空间区域划分成矩形单元，空间可以用分层和递归方法进行划分。这种多层矩形单元对应不同的分辨率，并且形成了一个层次结构：每个高层单元被划分成低一层的单元。关于每个网格单元的属性的统计信息（如均值，最大值和最小值）被作为统计参数预先计算和存储。

根据属性的相关统计信息进行划分网格，而且网格是分层次的，下一层是上一层的继续划分。在一个网格内的数据点即为一个簇。

STING算法的优点在于其网格结构有利于增量更新和并行处理，效率高，产生聚类的时间复杂度为O(n)，缺点在于其聚类质量取决于网格结构的最底层的粒度。除此之外，STING在构建一个父亲单元时没有考虑到子女单元和其他相邻单元之间的联系。所有的簇边界不是水平的，就是竖直的，没有斜的分界线。降低了聚类质量。

而CLIQUE(Clustering In QUEst)算法综合了基于密度和基于网格的聚类方法, 给定一个多维数据点的集合，数据点在数据空间中通常不是均衡分布的。CLIQUE区分空间中稀疏的和“拥挤的”区域（或单元），以发现数据集合的全局分布模式。接着，如果一个单元中的包含数据点超过了某个输入模型参数，则该单元是密集的。在CLIQUE中，簇定义为相连的密集单元的最大集合。

CLIQUE算法能自动发现最高维中所存在的密集聚类；它对输入数据元组顺序不敏感；也不需要假设（数据集中存在）任何特定的数据分布；它与输入数据大小呈线性关系；并当数据维数增加时具有较好的可扩展性。

WaveCluster(Clustering with Wavelets)采用小波变换聚类，它是一种多分辨率的聚类算法，它首先通过在数据空间上加一个多维网格结构来汇总数据，然后采用一种小波变换来变换原特征空间，在变换后的空间中找到密集区域。

WaveCluster的计算复杂度是O(n)，能有效地处理大数据集合；发现任意形状的簇；成功地处理孤立点；对于输入的顺序不敏感；不要求指定诸如结果簇的数目或邻域的半径等输入参数；在实验分析中，WaveCluster在效率和聚类质量上优于BIRCH，CLARANS和DBSCAN；实验分析也发现WaveCluster能够处理多达20维的数据。

### 5.1.2 基于层次的算法

基于层次算法对给定数据对象集合进行层次的分解。根据层次的分解如何形成，层次的方法可以分为凝聚的和分裂的。凝聚的方法，也称为自底向上的方法，一开始将每个对象作为单独的一个组，然后相继地合并相近的对象或组，直到所有的组合并为一个（层次的最上层），或者达到一个终止条件。分裂的方法，也称为自顶向下的方法，一开始将所有的对象置于一个簇中，在迭代的每一步中，一个簇被分裂为更小的簇，直到最终每个对象在单独的一个簇中，或者达到一个终止条件。该类方法采用相应不同的相似性计算方法，层次聚类方法可以是基于距离的或基于密度或连通性的。常见的算法有BIRCH算法、CURE算法、CHAMELEON算法等。

BIRCH算法是利用了一个树结构来进行快速聚类，其结构类似于平衡B+树，树的每一个节点是由若干个聚类特征(Clustering Feature，简称CF)组成。每个节点包括叶子节点都有若干个CF，而内部节点的CF有指向孩子节点的指针，所有的叶子节点用一个双向链表链接起来。

在聚类特征树中，一个聚类特征CF是这样定义的：每一个CF是一个三元组，可以用（N，LS，SS）表示。其中N代表了这个CF中拥有的样本点的

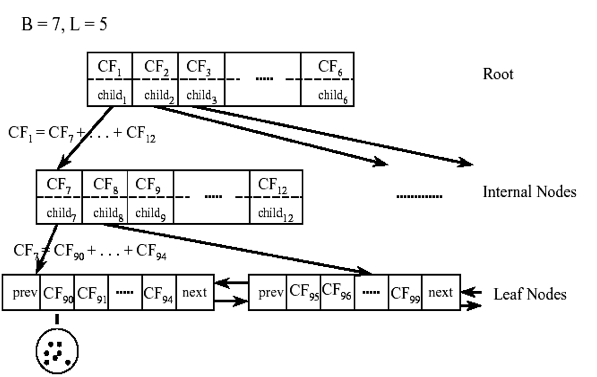
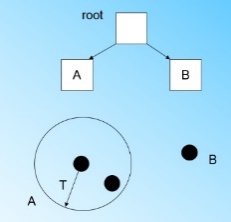
数量，这个好理解；LS代表了这

图 5-1 CF三元组

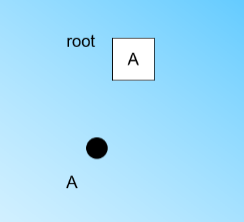
个CF中拥有的样本点各特征维度的和向量，SS代表了这个CF中拥有的样本点各特征维度的平方和。

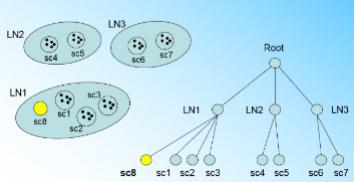
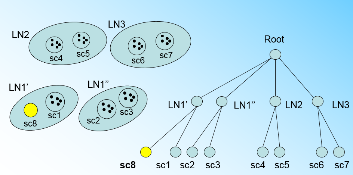
CF满足线性关系，也就是CF1+CF2=(N1+N2,LS1+LS2,SS1+SS2)，在CF Tree中，对于每个父节点中的CF节点，它的(N,LS,SS)三元组的值等于这个CF节点所指向的所有子节点的三元组之和。如图5-1所示，根节点的CF1的三元组的值，可以从它指向的6个子节点（CF7 - CF12）的值相加得到。

对于CF Tree，有几个重要参数，第一个参数是每个内部节点的最大CF数B，第二个参数是每个叶子节点的最大CF数L，第三个参数是针对叶子节点中某个CF中的样本点来说的，它是叶节点每个CF的最大样本半径阈值T，也就是说，在这个CF中的所有样本点一定要在半径小于T的一个超球体内。

对于上图中的CF Tree，限定了B=7， L=5， 也就是说内部节点最多有7个CF，而叶子节点最多有5个CF。

关于生成CF树，先定义好CF Tree的参数： 即内部节点的最大CF数B， 叶子节点的最大CF数L， 叶节点每个CF的最大样本半径阈值T，一开始CF树是空的，导入第一个样本点，生成一个CF三元组A，再导入第二个样本点，如果这个样本点和第一个样本点A，在半径为T的超球体范围内，就将第二个样本点页加入CF三元组A，如果这个节点不能融入刚才前面的节点形成的超球体内，就用一个新的CF三元组B，来容纳这个新的值。如果一个三元组的CF数超过了叶子最大CF数L就要对其进行分裂，将此节点分成两个节点，按照此规则建立起CF树，如图5-2所示。





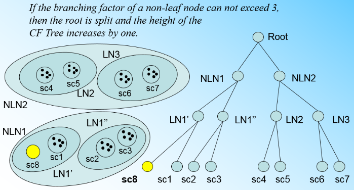


图5-2 CF树建立步骤

将所有的训练集样本建立了CF Tree，一个基本的BIRCH算法就完成了，对应的输出就是若干个CF节点，每个节点里的样本点就是一个聚类的簇。

BIRCH算法的优点在于节约内存，所有的样本都在磁盘上，CF Tree仅仅存了CF节点和对应的指针。聚类速度快，只需要一遍扫描训练集就可以建立CF Tree，CF Tree的增删改都很快。可以识别噪音点，还可以对数据集进行初步分类的预处理。而由于CF Tree对每个节点的CF个数有限制，导致聚类的结果可能和真实的类别分布不同，这是BIRCH的缺点，还有对高维特征的数据聚类效果不好。

### 5.1.3 基于划分的算法

划分方法首先根据给定要构建划分的数目k创建一个初始划分，然后采用一种迭代的重定位技术，尝试通过对象在划分间移动来改进划分。

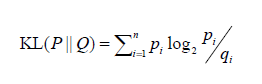
著名的k-means算法属于基于划分的聚类算法，它不仅是最简单的聚类算法，也是最普及且最常用的。

首先从n个 数据对象任意选择 k 个对象作为初始聚类中心；而对于所剩下其它对象，则根据它们与这些聚类中心的相似度（距离），分别将它们分配给与其最相似的（聚类中心所代表的）聚类；然后再计算每个所获新聚类的聚类中心（该聚类中所有对象的均值）；不断重复这一过程直到标准测度函数开始收敛为止。一般都采用均方差作为标准测度函数. k个聚类具有以下特点：各聚类本身尽可能的紧凑，而各聚类之间尽可能的分开。

换言之，k-means算法就是为了构造K个分组，首先初始选择K个簇心，把每个数据分配到与之最相似的组中，所有的点都分配到相应的组后重新计算每组的质心，又重复将每个数据进行分组，不断循环直到数据的划分不在变化或满足一个收敛条件。

k-means算法在做比较相似度的时候通常是用欧几里得的距离相似性，即数据点会分配到离其最近的质心上，这种情况下聚类的结果与初始样本的分组相关，初始分组会对结果影响很大。所以通常在初始分组时，会先在n个点中选择两个相距最远的两个点，然后在剩下n-2个点中找距离那两个点最远的一个点作为下一个聚类中心，直到找到k个点。

除了利用欧几里得的距离相似性，还可以使用KL即交叉熵，根据概率分布的差异去进行分组。针对P={p1,p2,p3,…,pn},Q={q1,q2,…,qm}，P和Q的交叉熵定义为：



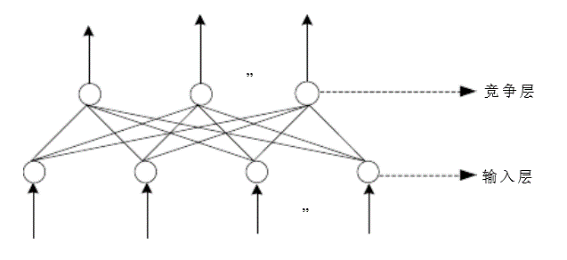
这个交叉熵就可以用来对比两个文本之间的相似性。在一个多指标评估系统中，KL交叉熵就可以解决指标权重分配的问题。

K-means算法具有很好的可伸缩性，其计算复杂度为O(nkt)，其中，t是循环的次数。K-means聚类算法的不足之处在于它要多次扫描数据库，此外，它只能找出球形的类，而不能发现任意形状的类。还有，初始质心的选择对聚类结果有较大的影响，该算法对噪声很敏感。

基于划分的算法线性复杂度，聚类的效率高，在中小规模的数据库中发现球状簇很适用，但并不适用于非凸面形状的簇。

## **5.2基于神经网络的算法**

基于神经网络的算法是通过神经网络自动寻找样本中的内在规律和本质属性，自组织，自适应地改变网络参数与结构。SOM网络是一种自组织特征映射网络，是一种无监督的学习算法，SOM的主要目的是将任意维度的输入信号通过计算映射转变为一维或者二维的离散映射，并且以自适应的方式实现这个过程。

生物学研究表明，在人脑的感觉通道上，神经元的组织原理是有序排列的。当外界的特定时空信息输入时，大脑皮层的特定区域兴奋，而且类似的外界信息在对应的区域是连续映像的。生物视网膜中有许多特定的细胞对特定的图形比较敏感，当视网膜中有若干个接收单元同时受特定模式刺激时，就使大脑皮层中的特定神经元开始兴奋，输入模式接近，与之对应的兴奋神经元也接近；在听觉通道上，神经元在结构排列上与频率的关系十分密切，对于某个频率，特定的神经元具有最大的响应，位置相邻的神经元具有相近的频率特征，而远离的神经元具有的频率特征差别也较大。大脑皮层中神经元的这种响应特点不是先天安排好的，而是通过后天的学习自组织形成的。

在生物神经系统中，存在着一种侧抑制现象，即一个神经细胞兴奋以后，会对周围其他神经细胞产生抑制作用。这种抑制作用会使神经细胞之间出现竞争，其结果是某些获胜，而另一些则失败。表现形式是获胜神经细胞兴奋，失败神经细胞抑制。自组织（竞争型）神经网络就是模拟上述生物神经系统功能的人工神经网络。

自组织（竞争型）神经网络在网络结构上一般是由输入层和竞争层构成的两层网络，如图5-3所示。竞争型神经网络构成的基本思想是网络的竞争层各神经元竞争对输入模式响应的机会，最后仅有一个神经元成为竞争的胜者。这一获胜神经元则表示对输入模式的分类。

图5-3 SOM结构

在竞争学习的过程中，神经元与输入层之间的连接的权值会随着输入模式(刺激)的变化选择性的进行调整，调整后的神经元(获胜神经元)的位置彼此之间变得有序．使得对于不同的输入，在网格上建立有意义的坐标系。因此，在自组织映射网络的输出层神经元构成的网格中，每个神经元的空间位置(坐标）表示着一个输入模式包含的一个内在的统计特征。

自组织映射网络实现步骤为：

1. 对网络权值进行初始化，选择较小的初始值

2.竞争过程：对每个输入信号，网络中的神经元计算他们各自的判别函数的值．判别值最大的特定神经元成为本次的获胜神经元。

3.合作过程;：获胜的神经元决定兴奋神经元的拓扑邻域(即：获胜神经元周围的空间位置内的神经元),提供相邻神经元的合作基础。

4.权值调节：通过对获胜神经元及其周围的兴奋神经元的权值进行调节，以增加它们对输入信号的判别函数值，随着权值的不断调整，获胜神经元对与相似的输入信号会有更强的响应(即：判别函数值越大)。

针对竞争过程，假设网络中输入信号(数据)空间的维度数为 m,从中随机选择一个输入信号(向量)记为x, x = [x1,x2,x3,.....xm]T，输出层中的每个神经元与输入层是全连接的结构，所以每个神经元的权值向量和输入空间的维数相同，神经元 j的权值向量记为：wj=[wj1,wj2,wj3,......wjm]T　j=1,2,3,4...L 其中 L 是输出层网络中神经元的总数，竞争过程就是找到与向量x最佳匹配的权值向量wj. 最佳匹配的意思是：对于　j=1,2,3,4..... L, 比较每一个神经元对应的权值与输入向量 x 的内积，选择最大者，对应的神经元作为获胜神经元．(假设所有神经元的阈值相同，偏置为负)。

最小化欧几里得几何距离就对应最大化内积，欧几里得几何距离最小意味着向量 x 与权值向量 wj的相似度越高。

针对合作过程，在竞争过程中产生的获胜神经元位于兴奋神经元拓扑邻域的中心位置。一个获胜神经元倾向于激活它紧接着的邻域内的神经元而不是隔得很远的神经元，所以对于获胜神经元的拓扑邻域按照侧向距离光滑地缩减。具体地，用hj,i表示以获胜神经元为中心的拓扑邻域且包含这一组兴奋(合作)神经元，j表示一个输出神经元，设 di,j表示获胜神经元和兴奋神经元之间的距离。假设拓扑邻域 hj,i是一个单峰函数，与di,j大小有关，获胜神经元与兴奋神经元之间的距离越小，兴奋神经元受到的刺激(影响)就越大。拓扑邻域 hi,j也可以表示兴奋神经元受到影响的程度。

单峰函数hj,i满足在di,j=0和dj,i=0处在获胜神经元处达到最大值，hi,j的幅值随着距离di,j的增加而减小，距离趋于无穷大时幅值趋于０。邻域函数依赖于获胜神经元和兴奋神经元在输出空间上的位置距离，不依赖于原始输入空间的距离度。

在SOM网络中还有一个特征是拓扑邻域的大小随着时间收缩．这要求拓扑邻域函数hi,j的有效宽度σ随着时间减小来实现．对于σ依赖与时间 n 流行的选择是：σ(n)=σ0exp(−n/τ1)，σ0为σ的初始值，τ1为时间常数。

在网络进行学习的初始阶段，拓扑邻域 hi,j应该包含以获胜神经元为中心的所有神经元，然后随着时间 n (即:迭代的次数增加)慢慢收缩。宽度 σ(n)以指数下滑，拓扑邻域也以相应的方式收缩。hj,i会减少到仅有围绕获胜神经元的少量邻居神经元或者减少到只剩下获胜神经元。

在权值调节中，自组织网络的神经元的权值 wj随着输入向量 x 的变化而改变，wj有一个自适应过程，在n轮的迭代中，n 时刻确定拓扑邻域的有效半径内所有的兴奋神经元,每一轮的训练中，拓扑邻域 hj,i(n)和有效半径(随着时间 n 衰减)，权值wj(n+1)=wj(n)+η(n)(x(n)−wj(n))，j包括n 时刻拓扑邻域的有效半径内所有的兴奋神经元(含获胜神经元)，η 为学习效率参数，学习效率也是随着时间逐渐衰减的：η(n)=η0exp(−n/τ2)。

权值更新的公式，实际上是在将获胜神经元和拓扑邻域内的兴奋神经元的权值向量向，向输入向量x移动，随着训练数据的重复出现，拓扑邻域内的网络权值向量的分布会趋于服从输入向量的分布，网络中的相邻神经元的权值向量会很相似。

SOM网络就是通过针对输入向量x寻找其获胜神经元，通过不断迭代更新权值直到特征映射不再发生明显变化，最后的输出层中每一个神经元表示着一个输入模式包含的一个内在的统计特征。

未来的聚类分析会成为大数据分析的重要基础，海量的数据都需要经过聚类的过程，未来会存在更多无先知信息的数据，利用神经网络的无监督聚类分析会越来越多。

**6 文本分类与聚类的应用**

高速发展你的Internet是一个庞大而充满混沌的互联网络，它为信息发布提供了广阔的舞台。然而随着计算机网络技术和Internet的迅速发展，给信息的查询和检索带来了极大的麻烦。通过不懈努力，人们逐渐掌握了多种技术对不断膨胀的信息进行收集和整理。各种海量信息的复杂性和非结构性，对信息的处理带来了极大地挑战。由于不同的领域对网络信息拥有各种不一样的需求，比如准确而全面的信息查询和检索系统（查全率，查准率）、精简的文摘资料（自动文摘系统，可视化）、个性化的信息服务（订制服务）、信息的过滤、加工和整理（文本挖掘、发现潜在信息）等等。而文本分类与聚类对于这些信息获取的需求也有着不可忽视的作用，如合理组织检索结果、提供可视化的多文档摘要、加速检索过程、个性化信息服务等等。文本分类与聚类还存在着一些问题，比如文本对象的高维性（算法不可实现，时间复杂度大）和稀疏性（降低分类和聚类性能）、同义词和近义词问题（潜在语义索引）、效率和精确度之间的搭配问题、参数优化问题等。虽然文本分类与聚类由于存在的这些问题还无法大规模的应用，但是在某些特定的领域它已经大放异彩。

首先对文本分类的应用进行综述。

**6.1 文本分类的应用**

文本分类就是给定分类体系，将文本集中的每个文本分到某个或者某几个类别中，文本分类过程可以分为手工分类和自动分类。前者最著名的的实例是yahoo的网页分类体系，是由专家定义了分类体系，然后人工将网页分类。由于这种方法需要大量人力，现实中已经采用的很少了。自动文本分类算法大致可以分为两类：知识工程方法和机器学习方法。知识工程方法指的是由专家为每个类别定义一些规则，这些规则代表了这些类别的特征，自动把符合规则的文档划分到相应的类别中。上个世纪90年代后，机器学习方法成为主导。机器学习方法与知识工程方法想比，能够达到相似的精确度，但是减少了大量的人工参与。所以，目前文本分类的应用大多也以基于机器学习方法的文本分类为主。

下面举几个文本分类应用的例子加以描述。

**6.1.1 文本分类在新闻自动分类与推荐的系统中的应用**

随着电子商务的不断发展壮大，商品的种类和数量也在不断的增加，不论你是何种身份，用户或者是管理员，在面对成千上万种商品的时候，总会显得很无措。使用者需要花费大量时间在类别判断和浏览那些无用的商品信息上，当前在这个追求效率的时代，必然会导致大量用户的流失。为了提高使用者的使用效率，减少使用者在无用信息上的逗留时间，自动分类和推荐系统逐渐被大众所熟知。

从目前的淘宝来看，淘宝平台上有着数以千万不同种类的商品，另外同种商品还有众多的卖家，而且卖家的商品也参差不齐。使用者往往会花费大量时间在商品的种类和信息上。当自动分类和推荐系统的出现后，只需要填写相应的商品信息，系统便可自动将其进行分类，用户也再不需要去花费太多的时间在寻找商品上，购物平台可以根据用户的购买等一系列隐性或显性特征进行分析，从而得 到与该用户相似度较高的其他用户或者预测到该用户的消费需求，并将结果推送给用户，这使得用户可以能够快速有效的寻找到自己想要的商品信息。

在自动分类和推荐系统中，最关键的技术就是对包含各种各样信息的文本的分类，这也是最难的，下面我就举一个具体的新闻自动分类与推荐系统的实例来说明文本分类在其中的应用，在这一系统中，只对涉及到文本分类的部分加以说明，而涉及到推荐的部分则不再加以描述。

在新闻自动分类与推荐系统中，包括四个部分：新闻文本的处理、关键词的提取、类别的判断以及新闻的个性化推荐。而在这四个部分中，前三个部分都属于文本分类的范畴，可见文本分类在其中的重要性。

系统流程从发布新闻开始，首先使用Ansj分词工具对新闻进行处理，包括去停用词以及用户自定义词的提取。计算每个词的TF-IDF值，将其值高的名词作为该新闻的关键词，然后通过训练好的SVM模型对该新闻的类别做出判断，并且将分类的结果反馈给使用者，如果使用者对推荐的结果不满意可以手动修改类别。由于训练好的SVM模型是根据现有数据库中的新闻建立的，这就使得当新闻量较少的时候，很难反馈出较为合适的类别推荐给用户。这时，为了能够提升分类结果的准确率，在对新闻进行分类之前，我们将会不定时的对现有的新闻集合进行分词处理，将其中的一 些TF-IDF值较大的名词组成我们的属性值集合，并将这些新闻集合处理成 LibSVM所需的测试集合。分类好的新闻会根据不同的类别展示给用户查看，其中我们也会根据用户的行为信息计算出与该用户相似度较高的用户，并向其推荐与之相似用户看过的但是该用户没有看的新闻。新闻自动分类过程如图6-1所示。

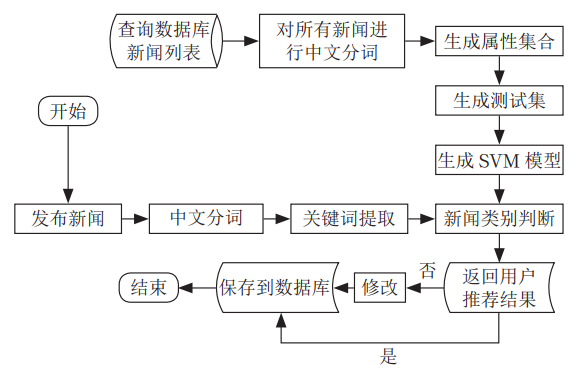
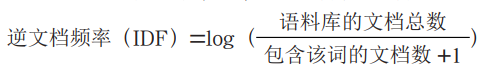


图 6-1 新闻文本自动分类图

在新闻文本自动分类的实现中，主要涉及利用TF-IDF算法和筛选名词提取新闻关键词和利用LibSVM实现新闻的分类两个步骤。首先在TF-IDF算法和筛选名词提取新闻关键词这一步骤中，讲问题分为两类，“军事新闻”和“教育新闻”。 在每个新闻类别中都有相应的关键字。如在军事新闻中有关键字：海军、空军、导弹、国防部等。首先采用Ansj分词技术对新闻进行分词并保存每一个词的词性，再用TF-IDF算法计算出每一个词的TF-IDF值，按照降序排列取出 TF-IDF值大于0.1的名词作为该新闻的关键词。例如，对《解放军这款大杀器发

射管“通红”，为何还有东西向后射？》这则新闻进行分析，首先会对该新闻的内容进行分词操作，通过计算提取其中的关键词，一般都会认为某一个词在这篇文章中出现的次数越多则说明这个词在文章中的重要程度越高。所以需要先找出该篇文章频数最多的那个词，因此先用Ansj对这篇文章进行分词操作，保存每一个词的词性并统计其频数，从而计算出该词在整篇文章中出现的频率（TF）。在此期间，需要设置一些停用词，因为在对一篇文章分词结束之后，发现很多频率高的词，如：的，我，我们等，毫无意义，设置停用词就是为了将这些毫无意义的词去掉，提高获得高频率词的准确度。在滤掉停用词后发现在其中有些词汇，如：中国，空降兵，反坦克火箭系统，他们的频率是一样多的，但人为的来看这三个词的重要性并不一样，可区分新闻类别的区分度也不一样，所以要进一步的筛选出更重要的关键词，这时需要对其逆文档频率（IDF）进行计算，其计算方法如下：



从上面的式子可以得知，如果一个词在其他新闻中出现的次数很少，但是在该篇文章中却大量出现，则就认为这个词具有很强的区分度，可以做关键词。最终将上面所求得的每个词的频率（TF）和逆文档频率相乘后，按照降序排列并取出其TF-IDF 值大于0.2的名词作为该新闻的关键词。

在利用LibSVM实现新闻的分类这一步骤中，比较关键的就是LibSVM，LibSVM 是台湾大学林智仁（Lin Chih-Jen）副教授等开发设计的一个简单、易于使用和快速有效的SVM模式识别与回归的软件包；该软件有一个特点，就是对SVM所涉及的参数调节相对比较少，提供了很多的默认参数，利用这些默认参数就可以解决很多问题；并且提供了交互检验（Cross Validation）的功能。该软件可以解决C-SVM分类、V-SVM分类等问题，包括基于一对一算法的多类模式识别问题。 我们把利用TF-IDF所收集的两个新闻类别的关键词作为分类的属性。我们规定 -1表示表示军事新闻，1表示教育新闻。比如现在有100条军事类的新闻，我们选取其中的80条作为训练新闻集，剩下的20条作为测试的新闻集。我们将每一条新闻进行分词操作，保存其词的词性，并在属性词集中查找是否存在改词，如果分析出的词出现在属性词集中，则将该属性标为1，否则记为0。利用这种方法将训练集合中的新闻转化为 LibSVM 所需要的数据格式，接着使用LibSVM对其进行训练获得相应的模型。最后将测试的新闻集合也按照上述分词处理，转化成LibSVM 所需要的数据格式，将其与得出的模型进行对比分析，得出预测结果。

这个系统还有一个基于协同过滤算法的推荐这一实现步骤，但其与文本分类无关，所以不再描述具体细节。

对与文本自动分类测试，测试的数

据来自于复旦大学李荣陆用于文本分类研究所使用的语料库，该训练集共 9804 篇文档，分为20个类别。训练语料和测试语料基本按照1︰1来划分的。本文提取了其中军事类和教育类的新闻，共137 条新闻。每个分类各取10条新闻作为测试数据，剩下的作为训练数据。对系统自动分类的准确率进行统计，可得出准确率受到关键词条件（即TFIDF的取值和是否自取名词）的影响。经过测试，可知当关键词获取条件为只取 TF-IDF> 0.25 的名词时，其准确率最高，准确率高达85%。虽然随着TF-IDF的数值继续增加，其准确率却没有出现明显的下降，但是由于会出现关键词较少等情况不宜采纳。具体实验数据如图6-2所示。

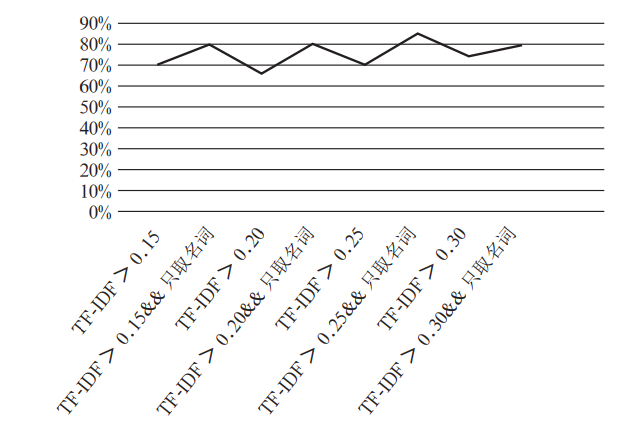


图 6-2 新闻文本自动分类实验结果

**6.1.2 文本分类在法律文书处理领域的应用**

早期的法律文书处理的系统主要以专家系统为主，它存在以下缺点：支持严密逻辑推理，但基于大量专家手写规则；适用于限定领域，难以大规模扩展；规则集的覆盖与冲突；模糊信息处理等。因而总结了一些早期工作的经验，如机器并不能代替法律实务工作者；应成为信息化中的重要组成（两高、律所.....）；以法律文书为主要研究对象等。法律与人工智能的结合的发展前景是可观的，一些一流院校已经开设了相关项目，如Stanford CodeX、北大法律人工智能中心、清华法学院计算法学专业、意大利等欧洲院校的法律信息学专业等。初创企业有LawGeex,Premonition,Ross Intelligence,...2017年前九名的创业公司融资总量就达到了两千九百万美元。目前应用还停留在民商领域为主、帮助普通人理解专业用语、熟悉业务流程、为法律或商务专业人士节约时间等方面。法律文本处理的应用的主要框架是首先由法条、法规、诉讼、非诉文书、仲裁文书、侦查相关文书、法律实务文书等这些标注数据进行法律知识体系构建、信息抽取、语义分析、文本分类、文本生成、信息检索、文本推理、篇章分析、论辩分析等处理，最终可以得到罪名预测、量刑预测、资源推送、法律文书生成、民事调解、智能问答等应用，这些应用可以为公安、检查、法院、律师、用户等带来帮助。其中在处理法律领域的标注数据时，可以看到文本分类是其关键的一环。文本分类典型任务有判决结果预测和事实认定、情节认定。其中判决结果预测包括罪名分类、法条分类、刑期预测，而事实和情节认定则包括主从犯分类、是否从重、是否从轻；重复、虚假事实判定、是否批捕、是否起诉、是否减刑。同时，文本分类也存在着一些挑战，如分类目标（罪名、法条等）多样、区别细微；训练数据质量参差不齐、训练数据不均衡、长尾效应显著；需常识、语义等多种类型的推理技术协助等。

用一个具体的实例来说明文本分类在法律文本处理系统中的作用。法律文本如下：被告人吴某甲以外出打工为名，将贵州省XX县XXX乡村民吴某诱骗出门，后与王某（在逃）、曹某（已判刑）将吴某带到江苏省丹阳市XX镇XXX村以2900的价格卖给张某为妻，三人共同分赃。案发后被告人吴某甲向吴某家人退还人民币600元。其中，可以提取到的文本中核心的特征包括被告人吴某甲、诱骗出门、以2900的价格卖给张某为妻，然后得到情节：主犯、退还部分赃款，然后根据情节的特征向量将文本分类到刑法第十二条、刑法第二十五条第一款、刑法第六十七条第三款，最后根据情节和相关法条的特征向量将文本分类到罪名:拐卖人口罪、刑期：有期徒刑3年、罚金：人民币2000元，从而最终达到判决结果预测的目的。

**6.2 文本聚类的应用**

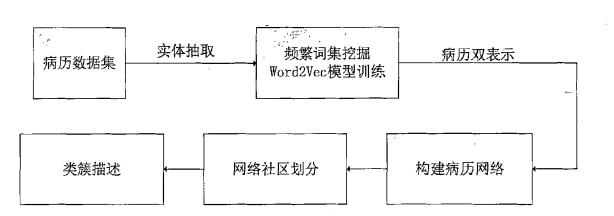
文本聚类其实就是将文本集合分组成多个类或簇，使得在同一个簇中的文本内容具有较高的相似度，而不同簇中的文本内容差别较大。根据聚成的簇的特点，聚类技术通常分为层次聚类和划分聚类，前者比较典型的例子是凝聚层次聚类算法，后者比较典型的例子是k-means算法。但是，近年来，由于神经网络、深度学习的发展，聚类算法作为典型的无监督算法在各个领域都有应用。

**6.2.1 文本聚类在电子病历分析中的应用**

在我国，由于基于电子病历的EMR系统起步较晚，再加上中文病程描述的复 杂性，使得基于电子病历的医疗分析研究面临很多的问题。首先在我国还没有统 一的术语规范标准，其次由于中文语言的丰富性，使得病例中关于病人病程描述中的特征词抽取变得极为困难。随着我国医疗领域中对电子病历应用的不断探索，目前临床决策支持诊断系统也成为研究的一个热点，这些研究中的重要组成部分依然是电子病历。同时为了将不同病种的电子病历做归类或组织化管理，方便医生进行病历分析。国内学者将文本聚类算法应用于电子病历聚类中通过聚类算法将已有的电子病历数据集根据病种来进行划分。文本聚类技术的引入对于病历归类和临床决策支持都有着重大意义。而在电子病历聚类过程中，病历间的相似性度量是最重要的一个步骤，为了能够更加精准的度量两份病历之间的相似性，有学者提出根据电子病历来计算病人间相似性的方法，将患者的诊断、人口数据、生命体征等信息加入到相似度计算中。

举一个具体的应用于电子病历的文本聚类的例子及它的实现过程和结果。

最开始的步骤就是对数据集进行清洗，这就需要结构化电子病历，病人数据部分全部记录在word文档中，利用 python语言将内容进行解析并保存在数据库和文本文件中，由于病人的个人信息属于个人隐私范围，只保留病人的诊疗信息。电子病历聚类方法的整体框架主要是利用FIC聚类算法思想进行数据的组织化管理，并根据电子病历数据特征提出一种基于频繁词集和词嵌入模型的电子病历混合聚类算法FIC-Hybrid。该算法为了充分挖掘数据中的上下文关系，在文本表示中引入Wbrd2vec模型，提出基于频繁词集与Word2Vec双表示的文本表示模型。在计算文本间相似性时采用加权集成的策略，使得该算法更适应病历数据。以文本信息作为组织管理的依据，舍弃了图像的信息。设计框架如图6-3所示。

图 6-3 电子病历聚类算法流程图

下面对应用的实现的主要过程加以说明。首先在病历实体抽取过程中，依据北京市广安门医院所积累的命名实体词库来进行特征实体的抽取，通过将实体词库作为自定义词库导入分词工具中来抽取实体词。在电子病历双表示模型中，主要用到病例中的诊断信息、刻下症与病程治疗记录这三类信息。对于频繁词集表示的文本，由于每一类的记录信息都是频繁词集表示的形式，因此我们将三类记录信息合在一起。而对于Wbrd2vec向量表示，我们将病历根据信息的类别进行拆分，将一份病历由诊断、刻下症、病程治疗记录三个向量进行表示。在电子病历的相似性计算中，别将不同表示模型计算出的相似性进行集成，从而得到一个基于频繁词集与词嵌入模型相结合的集成相似值。同时根据病历数据中诊断信息、刻下症与病程治疗记录自身的特点，对这三个维度的Word2Vec向量进行加权集成，将三个特征的信息融合在一起，形成基于病历特征的集成相似性计算方法。在电子病历网络构建中，根据上面过程计算出的两份病历间的相似性来构建病历网络，设置相似度阈值，将相似度大于阈值的边权重设为1，否则设为0，只保留权重为1的边。在病历网络划分与类簇描述中，用FIC-K中的K-rallk-D算法对病历网络进行划分，对于划分好的病历簇，同样统计簇中出现频率较高的频繁词集作为描述词，通过实验分析，将频率高于10的频繁词集作为类簇描述词。

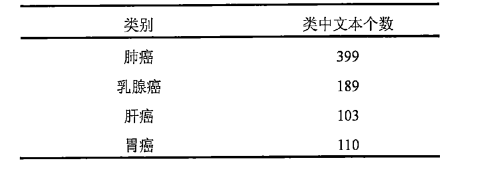
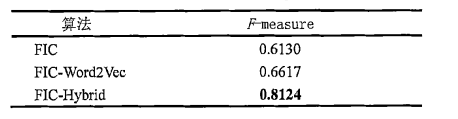
对于实验，实验数据来自北京御方堂医院，共收集了2543份肿瘤电子病历数据。 由于一些病例中病程记录的缺失，最终按照病种筛选出801份病历，这些病历来自肺癌、肝癌、乳腺癌和胃癌四类病种。其详细信息如表6-1所示。

表 6-1 病例数据集

实验结果如表6-2所示。

表 6-2 电子病历聚类算法精度比较

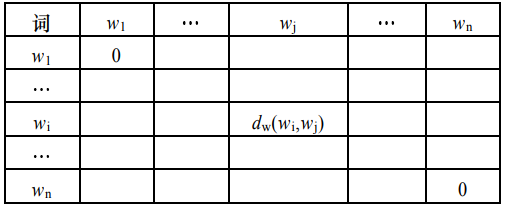
**6.2.2 文本聚类在自动文摘中的应用**

自动文摘是指利用计算机自动地从原始文献中生成准确全面地反映文献中心内容而且语言简洁连贯的摘要。在某种意义上信息检索变得比信息本身还重要。作为自然语言处理的一个重要分支的计算机自动文摘已成为Internet信息时代的必然需求。

基于文本聚类的自动文摘系统的整体框架采用多层B/S结构从逻辑上分为 Web服务器 应用服务器和数据库服务器由 Web服务器提供系统的输入/输出服务应用服务器为 Web服务器提供问题处理的逻辑运算服务数据库服务器管理系统的应用数据库语义知识库和领域知识库等系统，主要组成有自动分词和标注模块、基于统 计的单文档文摘句确定模块、基于句子聚类的段落文摘句确定模块、基于句子聚类的单文档文摘句确定模块、基于自然语言理解的文摘句平滑处理模块、文档文本聚类模块、多文档自动文摘模块和知识库规则库。

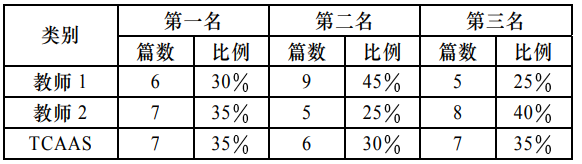
基于文本聚类的自动文摘系统的具体实现只对涉及到文本聚类的部分加以描述。文本聚类是自动文摘系统的核心功能模块。这一模块通过计算各个句子之间的距离，从而获得各个句子的聚合程度信息。某一段落中，与各个句子关系都较密切的句子，往往是本段落的中心句。为了计算各个句子之间的距离，需要对句子进行向量化表示。我们将句子向量的参数确定为句中所有的非功能词。实现过程包括：(1)构造词义距离表。句子由词组成，要计算两个句子的语义距离必然涉及到两个句子中相应词的词义距离。为此，构造词义距离表，其结构如表4-3所示.(2)句子加权。文本聚类中不需要对所有的句子都进行聚类分析，为此对重要的句子进行加权。句子加权中，以下4类句子得到的权值较高L：1)标题句,即含有标题中出现的有效词的句子；2)含有高频有效词的句子；3)文中重要位置的句子。如段首句、段尾句和第一自然段中的句子等；4)含有提示短语的句子。如含有“总之”、“因此”、“所以”、“论述了”、“表达了”等短语的句子。(3)句子的向量表示和聚类分析。根据功能词表和停用词表，首先将每个句子中的功能词和停用词进行显性标，然后对其余的字串进行基于 TTPFS 算法的分词处理和词性标注。对分词和标注后的句子进行向量化表示，得到每个句子的 n元组向量模型(w1,w2,... ,wn)。计算任意两个句子 si 和 sj 的语义距离 并把与其他句子语义聚合度在我们规定的阈值以内的句子作为文摘句。(4)基于自然语言理解的文摘句平滑处理。直接摘取的文摘句堆砌在一起，显然不够连贯、 流畅，还需进行润色平滑处理。自动文摘系统采用的是基于自然语言理解的文摘句平滑处理方法。

表 6-3 词义距离表



对于实验，在1998 年和2000 年人民日报电子版中检索了20篇有关塑料的文章，采用 Turing 测试法对自动文摘的结果进行测试。首先用 TCAAS对 20篇文章做出文摘，然后请两位中文系的教师为20篇文章各做出一篇人工文摘。将TCAAS自动生成的20篇文摘与两位教师做出的40篇文摘混在一起。最后，请中文专家对每一篇文章的3篇文摘进行严格的黑箱打分。其测试结果如表6-4所示。

表 6-4自动文摘系统的实验结果



从表6-4的实验结果可以看出，自动文摘系统的文摘效果优于至少一位教师的文摘效果的比例达到65%，优于至少两位教师的文摘效果的比例达到 35%。由此可见，自动文摘系统的文摘效果略优于人工文摘，已达到实用化的水平。

**7 总结与展望**

文本分类与文本聚类是文本挖掘领域的两个重要分支，它们各有优缺点。文本分类的效果普遍要比文本聚类要高，但是需要大量带标签的训练数据对分类器模型进行训练；相比之下，文本聚类不需要任何训练数据，但是最终的分类效果也普遍不如文本分类。

目前由于神经网络、深度学习的浪潮席卷人工智能领域，文本分类与聚类也无法幸免。近年来，涌现出了一大批基于神经网络的文本分类与文本聚类的算法，也冲击着传统的基于机器学习的模型的权威。在文本分类中，传统的机器模型在分类前首先要做特征工程，例如把文本转换成词袋，并转化为TF-IDF矩阵，然后再做分类。而使用神经网络模型可以使它自己提取特征并进行文本分类，并能获得优于传统机器学习模型的能力。文本聚类的情况也与文本分类的情况类似。

所以，对于未来文本分类与文本聚类的发展，我认为在相当的一段时间内，还是以基于神经网络的文本分类与聚类的研究为主。但是，神经网络自身也存在着瓶颈，如难解释性、局部陷阱等，所以基于神经网络的文本分类与聚类的研究未来也极有可能遭遇瓶颈。那时，可能基于神经网络的文本分类与聚类算法还得回归原始，到基于规则、基于机器学习的算法中汲取养分。

参考文献

[1]Ponte J M, Croft W B . A Language Modeling Approach to Information Retrieval[C]//Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval, 1998: 275-281.

[2]靳小波.文本分类综述[J].自动化博览,2006(S1):24+26+28-29.

[3]黄昌宁,赵海.中文分词十年回顾[J].中文信息学报,2007(03):8-19.

[4] [Yun Lin](https://ieeexplore.ieee.org/author/37085425048); [Jie Wang](https://ieeexplore.ieee.org/author/37085419363). Research on text classification based on SVM-KNN: Software Engineering and Service Science (ICSESS), 2014 5th IEEE International Conference on.

[5]Kretchmar M , Zhao Y . Text Message Authorship Classification Using Kernel Support Vector Machines[C]// Computational Science and Computational Intelligence (CSCI), 2014 International Conference on. IEEE, 2014.

[6]Joulin A , Grave E , Bojanowski P , et al. Bag of Tricks for Efficient Text Classification[J]. 2016.

[7]Vieira J P A , Moura R S . [IEEE 2017 XLIII Latin American Computer Conference (CLEI) - Córdoba, Argentina (2017.9.4-2017.9.8)] 2017 XLIII Latin American Computer Conference (CLEI) - An analysis of convolutional neural networks for sentence classification[C]// Computer Conference. IEEE, 2017:1-5.

[8] Siwei Lai, Liheng Xu, Kang Liu, Jun Zhao. Recurrent Convolutional Neural Networks for Text Classification. AAAI Conference.

[9]hu Huan, Zhang Pengzhou, Gao Zeyang K-means Text Dynamic Clustering Algorithm Based on KL Divergence

[10]孙磊磊. AP聚类算法研究及其在电子病历挖掘中的应用[D]. 2017.

郭庆琳, 吴克河, 吴慧芳, et al. 基于文本聚类的多文档自动文摘研究[J]. 计算机研究与发展, 2007, 44(z2).

[11]在线热点新闻推荐系统研究和实现[D]. 南京航空航天大学, 2015.

[12]马海群, 王今. 基于神经网络的数据安全话题文本分类研究——以新浪微博为例[J]. 图书馆, 2017(5):36-39.