摘 要

近几十年来，国外学者对英文文本聚类投入了大量研究工作，并取得了一些优秀的成果。与英文文本聚类相比，中文文本聚类技术研究和应用起步较晚，文本聚类效果普遍不太理想。针对此现状，本文对中文文本聚类技术进行深入研究。重点在于改进经典 SOM 算法，并应用于中文文本聚类中。本文研究工作主要涵盖以下四点内容：

（1）研究中文文本聚类技术，包括中文分词、停用词过滤、特征选择等中文预处理技术以及各种聚类分析算法。

（2）针对特征项维数灾难导致计算负载过大，在预处理中引入同义词合并技术，实现特征空间的语义降维，提高系统聚类速度和准确性。

（3）重点研究经典 SOM 算法，针对其聚类数目需预先输入、网络结构固定、初始化效果不理想以及聚类结果依赖样本输入顺序，提出一种改进的自增长 SOM 算法予以解决之。

（4）采用 C#.net 技术设计实现了基于改进的 SOM 算法的《中文文本聚类系统》平台。然后进行了系统测试评估，评估结果表明改进 SOM 算法可以改善系统聚类效果。

关键词：文本聚类；SOM；中文分词；同义词合并；特征选择；预处理

Abstract

In recent years, foreign scholars have done a lot research, and made som outstanding results on the English text clustering. Compared with the English text clustering, Chinese text clustering technology research and application started more lately, and text clustering results in general is not very satisfactory. For this situation, this paper had some depth-study of Chinese text clustering techniques. It focused on improving SOM clustering algorithm, and applied to Chinese text clustering. It started work in the following aspects:

(1) Explored the Chinese text clustering technologies, including Chinese pretreatment such as Chinese word segmentation, Stop words filter, and Feature selection and so on and a variety of commonly used clustering algorithm.

(2) The curse of dimensionality for feature items would cause the large calculation load. So in the Chinese pretreatment it introduced synonyms merger technology, to achieve to reduce Feature space dimension and improve speed and accuracy of clustering.

(3) Explored SOM neural network theory and analyze its defects.Beacuse its Number of clusters to be pre-input, fixed network and unsatisfactory initialization result, this paper proposed an improved and growth-self SOM algorithm to solve these problems.

(4) Using C#.Net technology designed and implemented a "Chinese Text Clustering System" platform with both research and application value.And then conducted a systematic testing and evaluation, evaluation results show that the improved SOM algorithm can optimize cluster results.

Key Words: text clustering; self-organizing neural network; Chinese word segmentation; synonyms merger; feature selection; pretreatment

目 录

第1章 引言

1.1 研究背景与意义

自二十世纪90年代开始，随着网络技术的迅猛发展，大量的信息以文本（新闻语料，电子邮件，网页，科学论文...）的形式涌现在互联网上。信息检索、文本过滤、信息提取、分类和聚类技术正在迅速成为关键部件的现代信息处理系统，帮助最终用户选择，可视化和塑造它们的信息环境。

信息检索[1]从海量信息数据中搜索满足用户需要的内容，并以良好的组织方式展现给用户。文本过滤和路由技术自动分发文档到适当的文件读者，安排新引入的文件在适当的文件夹或目录中，拒绝不良性的访问。自动文摘技术类似的信息提取技术存在更多的潜力，减少了用户阅读的文本或邮件负担。这些应用大多需要利用文件或文件片段的无监督聚类技术：文档集的自行组织可以促进文本内容索引或搜索，对用户搜索结果进行聚类可以让可视化变得更轻松；以无监督方式形成的子类也能改善文本分类等。

自组织神经网络（Self-Organizing Map Neural Network, SOM）是在标准前向网络（如BP网络）之后，最受关注的神经网络之一。它以无导师监督的方式进行网络训练，通过网络结构的自组织从大量输入数据中发现并抽取其内在的特征，在网络输出结点权向量空间上形成输入数据分布拓扑映射图，反映输入数据的某种分布规律，能自动对输入模式进行分类，具有自组织学习、拓扑结构保持、概率分布保持及可视化等特征[2]。然而SOM传统学习算法和固定的网络拓扑结构以及不能按需要方便地在合适的位置生成新节点等大大影响了网络的收敛速度以及聚类准确性，限制了SOM网络的广泛应用。

本课题针对经典SOM算法聚类数目需预先输入、网络结构固定、初始化效果不理想以及聚类结果依赖样本输入序列，提出一种改进的自增长SOM算法，实现聚类数目和网络结构动态生成，初始化聚类中心与实际样本分布情况基本吻合；在调整阶段采用并行学习策略，实现对样本训练集的并行调整，使得聚类结果与样本的输入顺序无关。采用C#.net设计实现了基于改进SOM算法的《中文文本聚类系统》平台，实验表明系统性能得到改善。

1.2 课题来源

本课题来源于2008年江西省自然科学基金项目《基于粗糙集_Kohonen神经网络的用户模式挖掘》。

1.3 国内外研究现状

早在上世纪60年代，国外就开始对西文的文本聚类技术的研究，并且随着不断地发展，文本聚类相关的研究论文和研发产品也相继问世。文本信息研究大师Salton等人提出了向量空间模型[3][4]，开创了现代意义上的文本信息处理领域。

在向量空间模型的基础上实现了不少具有实际价值的文本挖掘系统。例如：IBM公司开发的TextMiner系统[5]，是一个拥有主题抽取、文本分类和聚类以及信息检索多功能的文本挖掘平台；Xtramind公司研发的XM-Mindset（http://www.gqb.gov.cn）平台[6]，是一个大型文档挖掘系统，引入了多项的智能核心技术，在文本聚类模块的查全率、查准率、收敛速度和通用性方面表现的突出；哥伦比亚大学研发的多文档自动文摘系统Newsblaster（http://www.newsblaster.com）[7][8]，采用了文本聚类技术，将每天同一主题的新闻组织到同目录下，然后自动文摘相关技术生成一篇高质量的文档摘要；vivisimo（http://www.vivisimo.com）和infonetware（http://www infonetware.com）是当前比较流行的搜索引擎平台，它们采用文本聚类技术，对检索返回的结果进行聚类，然后以分类的形式展现给用户，让用户在较短的时间内找到感兴趣的内容；不少的阅读平台，根据用户的兴趣，将用户感兴趣的读物进行聚类，从而挖掘用户的兴趣模式，并向用户提供信息过滤和信息推荐服务，数字图书馆服务采用了文档聚类技术，将高维文本向量空间映射到二维空间，实现聚类结果的可视化[9]。

国内的文本聚类等文本信息处理技术的研究起步相对较晚，开始于上世纪90年代，主要集中在中科院、清华大学、东北大学、北京理工大学、厦门大学以及山西大学等高等院校的自然语言研究室。虽然已有不少学者涉足该领域，但发展水平相对滞后，主要原因有两点：一、起步较晚，研究基础差；二、中文的文本信息处理相对复杂。由于中文自身的特点，中文文本聚类与西文文本聚类具有很大的不同，主要体现在：1、中文的词语之间没有像英文中有空格作

为单词的分隔标记符；2、中文的字、词、短语及语句存在着多种多样的歧义性；3、根据上下文环境的不同，中文具有不同的语义。上述3点导致了中文的文本信息处理与英文存在着差异，必须采用针对汉语的处理方法。中文分词是进行中文信息挖掘的基础，其准确性直接决定了后处理的准确性，因此引起了国内众多学者的关注。目前国内出现了不少优秀的中文分词系统，给中文的信息处理技术的发展提供了良好的基础。最为著名是中科院自然语言研究所开发的ICTCLAS（http://ictclas.org/）中文分词系统[10]，以高达 98%的分词准确率，31.5Kbytes/s 处理速率等多项优势，成为当前最流行的中文分词方法。性能比较好中文分词系统还有织梦分词系统（http://www.dedecms.com/）、Abot 汉语分词系统（http://www.abot.cn/）、第三代智能分词系统 3GWS（www.nlp.org.cn/）和 RLCSS 分词系统等，其准确性都达到 95%以上。至于汉语的歧义性和多义性的消解还处在研究阶段，比较流行的解决方法是采用概率统计结合字典的方法。

目前常用的聚类分析算法[11~16]主要包括基于划分的算法、基于层次的算法、基于密度的算法、基于网格的算法和基于模型的算法。自组织映射神经网络（SOM）[17~20]是典型的基于模型的算法，具有鲁棒性强，伸缩性好，聚类准确度高等特点，得到众多学者的青睐，广泛应用于模式识别、图像分割以及数据挖掘等领域。SOM 应用于文本聚类的成功案例也越来越多，比如 SOMLIB[21]和 WEBSOM[21]分别成为素文本聚类和 web 文本聚类的典型成功案例。

但随着近年来众多学者的研究，发现 SOM 具有网络结构单一，缺乏解决学习效率和准确性之间冲突的良好机制等缺点，大大影响了网络的收敛速度和聚类的准确性。因此，在 SOM 网络结构动态自适应调整和学习策略改进方面引起了众多专家学者的重视，试图研究出结构动态生成、学习率自适应的快速 SOM 聚类算法和结构，但实际上都未能达到理想的效果。1992 年 Bezdek 针对 SOM 学习率方面存在的缺陷，提出一种模糊的自组织神经网络 FSOM[22][23]，该方法将模糊 C 均值[24][25]的学习策略应用于 SOM 的学习过程，因此 FSOM 获取了模糊 C 均值算法在目标最优时结束，且聚类结果与输入数据的序列无关；然而 FSOM 的结构是固定的，在进行聚类分析之前须预先输入固定准确的聚类数目，而聚类的数目在实际中通常是预先不知道的。针对 FSOM 的网络结构的缺陷，天津大学的王莉提出一种采用树型结构的聚类模型（TGSOM）[26]，可根据样本实际分布情况在适当的位置生成新结点，聚类数目随着网络结构的形成而自动确定。天津大学的耿新青提出了一种动态模糊自组织神经网络（DFKNN）[27]，

在 TGSOM 的结构基础上采用收敛快而准确的学习率的计算公式，并以模糊聚类中心作为对应的神经元的权值，提高了 TGSOM 收敛速度和聚类准确度。然而 DFKNN 算法几乎完全采用了 TGSOM 的调整策略，只对优胜结点及其子节点进行调整，但往往存在其他结点比其子节点更接近输入样本模式情况，从而造成了学习的不公平性，降低了聚类结果的准确性。因此对于 SOM 的改进还有待进一步研究。

1.4 论文组织结构

第 1 章　引言

主要介绍研究背景与意义、课题来源、以及文本聚类技术的国内外研究现状。

第 2 章　中文文本聚类技术

主要介绍文本聚类的相关概念及基本理论。包括文本表示、中文分词、停用词过滤、同义词合并、权重计算以及特征选择等中文文本处理技术，以及几种常用的聚类分析算法和文本聚类的评估依据。

第 3 章　自组织映射神经网络的研究与改进

主要介绍 SOM 拓扑结构、算法描述、网络特性以及存在的缺陷，重点针对 SOM 聚类数目须预先输入、网络结构固定、初始化效果不理想以及聚类结果依赖样本输入顺序，提出了一种改进的自增长 SOM 算法，并通过算法分析对比表明，本文算法确实能够以上方面予以改进。

第 4 章　文本聚类系统的设计与实现

在文本聚类技术理论研究以及改进经典 SOM 算法的基础上，采用 C#.net 设计实现了《中文文本聚类系统》平台。

第 5 章　实验结果与分析

在《中文文本聚类系统》平台上，进行多种条件组合的实验结果对比分析，进一步探讨改进 SOM 算法和同义词合并分别在中文文本聚类中的地位。

第 6 章　总结与展望

对所做研究工作进行总结，并提出了需要进一步研究的工作。

第 2 章　中文文本聚类技术

2.1 文本聚类概述

文本聚类将聚类技术应用于文本处理，实现文本自动分类的过程。与文本分类不同的是[28]：它没有预先确定的类别，仅以文本之间的相似关系来组织文本集合。文本聚类技术可以在信息检索前进行前处理，即对文本库进行预先自动分类处理，然后在相关的类别上进行检索以提高检索速度；同时也用于对检索结果的进行后处理，即对检索结果进行聚类，可以帮助用户得到最相关的内容，也可以发现单纯使用排序很难发现的有用文档。

2.1.1 文本聚类的过程

聚类分析通常处理的对象是存储在数据库或者数据仓库中的结构化数据。但文本聚类和分类等文本挖掘技术处理的对象是以字符串流的方式存在的无结构化数据。由于目前计算机对自然语言的理解还处于初级阶段，聚类分析技术无法直接处理无结构化数据，因此在进行聚类分析之前，需要进行文本预处理。该过程总体来说可以分为三个阶段：预处理、聚类分析和聚类结果评估，如图 2.1 所示：

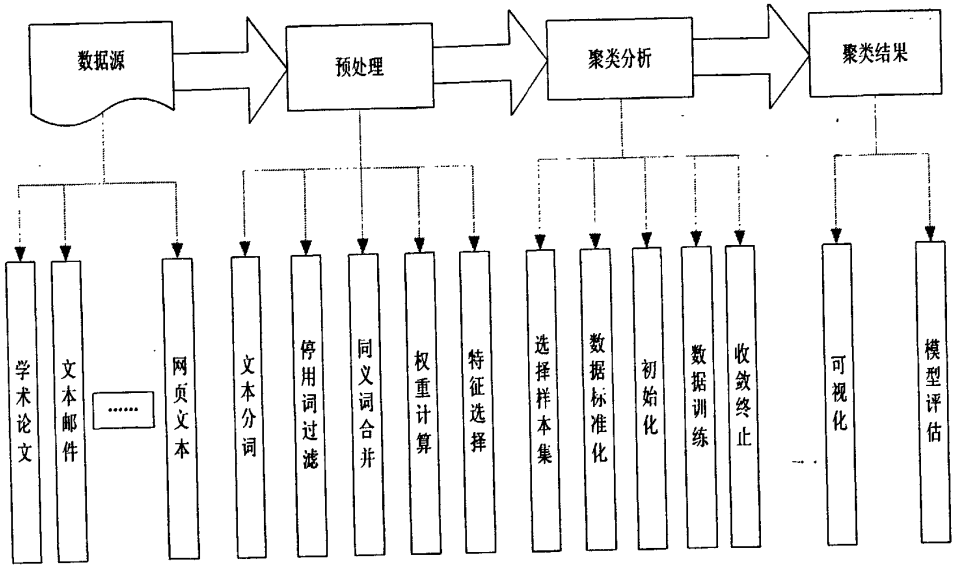

图 2.1　文本聚类的一般过程

其中预处理阶段的任务是将自然语言表示的文本，经过加工转换为计算机可以处理的文本-词条集合，文本中的字或词条被认为相互独立的。预处理主要包括文本分词、停用词过滤、词根还原或同义词合并、权重计算、特征选择或特征抽取等过程。

文本聚类分析在过程上与一般的数据挖掘的聚类基本相同，只是根据不同的应用领域，选取的聚类算法和相似度计算方法不同。聚类分析主要包括选择样本集、数据标准化、设定参数、初始化、数据训练以及收敛等过程。

聚类结果评估主要是采用相应的评估依据，评价聚类算法的准确性、收敛速度、伸缩性等。评估的方法主要有模型评估方法和可视化评估方法。模型评估是建立一个标准的数据模型，在模型的基础上评价聚类结果的准确性等，可视化评估是将聚类的结果以图表的方式展现给用户。

2.1.2 文本类的定义

目前关于文本类的定义还没有一个完全确切的衡量标准，下面给出了 3 种文本类的定义，都是基于分量之间的距离进行定义的，只是计算距离的方法有所不同。

假设集合 $D(n)$为文本集合，记为 $D(n)=\{t_i|i=1,2,...n\}$。其中 n 表示 D 中文本的数目，t_i 表示文本集合中第 i 个文本。d_{ij} 表示文本 t_i 和 t_j 两者之间的距离，ε 表示设定的一个阀值。

定义 2.1　D 为一个类当且仅当 D 中的任意两个文本 t_i 和 t_j 之间的距离小于等于阀值 ε，即有 $d_{ij}\le\varepsilon$。

定义 2.2　D 为一个类当且仅当 D 中的任意一个文本 t_i 与其他文本间的平均距离小于等于阀值 ε，即有 $\sum_{j=1}^{n} d_{ij}/(n-1)\le\varepsilon$ 其中 $j\neq i$。

定义 2.3　D 为一个类当且仅当 D 中的任意一个文本，至少 D 中存在另外一个文本 t_j，使得它们之间的距离小于等于阀值 ε，即有 $\forall i\exists j\ \ d_{ij}\le\varepsilon$。

描述一个含有 n 个文本对象的类 C 可以采用以下几个特征来表示。

类中心矢量：

$$Z_C=\sum_{i=1}^{n} d_i/n \tag{2.1}$$

类直径：可以采用多种方式来表示类直径，如下所示：

$$D_C=MAX(d_{ij}) \tag{2.2}$$

$$D_C=\sum_{i=1}^{n}(d_i-Z_C)(d_i-Z_C)^T \tag{2.3}$$

类的样本距离差：

$$A_C = \sum_{i=1}^{n} (d_i - Z_C)^T (d_i - Z_C) \tag{2.4}$$

类的样本协方差：

$$S_C = A_C/(n-1) \tag{2.5}$$

2.1.3 文本的相似度计算

文本的相似度被定义为文本之间内容的相似程度。文本的相似度通常是将文本表示成向量空间中的点，然后采用向量空间中点与点的距离来度量的。设文本 t_i 和 t_j 对应的文本向量分别为 $d_i=(w_{i1}, w_{i2}, ..., w_{in})$ 和 $d_j=(w_{j1}, w_{j2}, ..., w_{jn})$，$t_i$ 和 t_j 的相似度用 $sim(d_i, d_j)$ 表示，t_i 与 t_j 之间的距离用 d_{ij} 表示，度量 t_i 与 t_j 之间的相异程度。目前计算文本相似度和距离的方法比较多，代表性的有以下几种[28]。

（1） 余弦公式法

在文本相似度计算中，采用最多的是余弦公式法，随着夹角的减小，文本的相似度反而越高。其计算公式如下所示：

$$Sim(d_i, d_j) = \frac{d_i \bullet d_j}{|d_i| \bullet |dj|} = \frac{\sum_{k=1}^{n} w_{ik} \bullet w_{jk}}{\sqrt{\sum_{k=1}^{n} (w_{ik})^2} \bullet \sqrt{\sum_{k=1}^{n} (w_{jk})^2}} \tag{2.6}$$

（2） 内积法

采用向量的内积表示文本间的相似度，计算公式如下：

$$Sim(d_i, d_j) = <d_i \bullet d_j> \tag{2.7}$$

（3） 重叠系数法

$$Sim(d_i, d_j) = \frac{\sum_{k=1}^{n} w_{ik} \bullet w_{jk}}{\min(\sum_{k=1}^{n} w_{ik}, \sum_{k=1}^{n} w_{jk})} \tag{2.8}$$

（4） 雅可比系数法

$$Sim(d_i, d_j) = \frac{\sum_{k=1}^{n} w_{ik} \bullet w_{jk}}{\sum_{k=1}^{n} w_{ik} + \sum_{k=1}^{n} w_{jk} - \sum_{k=1}^{n} w_{ik} \bullet w_{jk}} \tag{2.9}$$

（5） 海明（Minkovski）距离法

$$d_{ij}(q)=\sqrt[q]{\sum_{k=1}^{n}\left|w_{ik}-w_{jk}\right|^{q}},(q>0) \tag{2.10}$$

当 q=1 时，海明距离公式变形为：

$$d_{ij}=\sum_{k=1}^{n}\left|w_{ik}-w_{jk}\right| \tag{2.11}$$

此时称为绝对值距离。

当 q=2 时，海明距离公式变形为：

$$d_{ij}=\sqrt{\sum_{k=1}^{n}\left|w_{ik}-w_{jk}\right|^{2}} \tag{2.12}$$

此时称为欧氏距离。

当 q=∞ 时，海明距离公式变形为：

$$d_{ij}=\max(\left|w_{ik}-w_{jk}\right|) \tag{2.13}$$

此时称为切比雪夫距离。

2.2 中文预处理

聚类算法难以处理无结构的文本对象，因此在进行文本聚类之前，必须进行预处理操作，将其表示成聚类算法可以处理的数据结构。在上一节已经提到，文本预处理包括文本表示、中文分词等操作，下面将逐一进行详细介绍。

2.2.1 文本表示

文本表示是指采用特征词集合来表示文本信息。当前常用的文本模型[29][30]主要包括布尔模型（boolean model）、向量空间模型（vector space model,VSM）以及概率模型（probability model）等。根据文本表示模型的不同，计算特征权重与加权、训练学习和相似度计算等方法也不同。其中向量模型表示的效果较好，近年来得到广泛采用。

向量空间模型（VSM）[3][4]是 Salton 等人在 20 世纪 60 年代提出的，并成功应用在 Smart 系统中。在向量模型中，一个文本 d_i 当作由一组特征项 $(t_1,t_2,...t_n)$ 组成的 n 维向量，文本 d_i 被表示成一条以特征项的权重为分量的向量模式

$(w_{i1},w_{i2},...w_{in})$,其中权重 w_{ij} 表示特征项 t_j $(j=1,2,...n)$ 对文本 d_i 分类的重要性。

VSM 将无结构化的文本流转化为有结构的向量空间，将复杂的文本处理转换为简单的向量计算，大大降低了计算的复杂程度。但是，向量空间模型基于这样一种假设：一个文本的类别仅与某些特征项在文本中出现的频率相关，与特征项在文本中出现的次序无关，以及特征项之间的上下文关系、位置和文本长度等都不加考虑。在很大程度上丢失了上下文的语义关系，以及损失了文本内部的结构信息，所以基于语义的文本表示模型成为学者研究的重要方向。

2.2.2 中文分词

英文文本分词相对简单，词与词之间以空格作为明显的分词标记，词语不需要做处理就可以进行向量化。而中文由于汉语的本身的特殊性，字与字之间不存在空格分割标记，因而存在如何正确分词的问题，即中文分词技术。

中文分词是中文聚类、分类等文本挖掘技术的关键基础步骤，分词准确性高低直接影响了文本处理结果的质量。

目前常用的中文分词方法存在三种：基于词典的方法、基于统计的方法和混合的方法[32][34]。

1、基于词典的方法

该方法实现比较简单，不需要进行词法、句法及语义分析，只依赖一个分词词典和分词规则，比如最大匹配法、最小匹配法以及最佳匹配法等。这些分词评估规则在一般情况下是合理的，但也存在一些错误的切词。

2、基于统计的方法

该方法的基本思想是首先切分出与词典匹配的所有可能的分词结果，然后采用统计模型和决策模型决定最佳的分词结果。这种方法的优点是能够发现所有的存在的切分歧义，但因为消除歧义的方法依赖统计模型和决策算法，所以需要大量的标注语料信息，并且分词的速度也因词典搜索空间增大而降低。

3、基于规则与统计的方法

首先依据词典以及最大匹配规则进行初步切分，然后对切分的边界处进行歧义探测，运用规则和统计相结合的方法来获取最优的切分，采用相应的规则解决人名、地名、机构名的识别，运用词法结构规则来生成复合词和衍生词。

中文分词目前面临的难点主要包括：词典的建立、未登录词的识别和词的歧义性[31~33]。

1）词典的建立

由于语言表达的多样性和随意性，使得难以建立一个规范的词典库。一个词语存在多种表示方式，比如，“跺脚”在平时口语中说成“跺跺脚”、“跺了下脚”等等。其中哪些词被加入词典库，还是所有的词都加入词典库，至今为止，仍没有一个统一的权威的依据。汉语中还存在后缀词和重叠词，比如“高效性”、“高高兴兴”等，因此词典如何处理处理这些词是一个难点。

2）未登录词的识别

在当前文本中存在这样一类词，它们不出现在词典中，但出现的频度非常高，比如人名、地名、外来词还有一些衍生词等。如果这些词不加以识别和处理的话，会干扰计算机对自然语言的理解，给词频统计带来很大的误差。未登录词的识别需要针对不同的情景采用不同的法则，采用相应的识别策略。识别未登录词可以很大程度上解决词语的歧义性，能够提高分词的精度。

3）词的歧义性

中文分词系统需要解决的关键性难点是消除词语的歧义性。汉语中存在的最基本的词语歧义性有以下两种类型。一类是交集型歧义，该歧义占 85%，形如：AB 和 BC 同时出现在词典中，比如：网球/ 场 vs 网/球场等；另一类是组合型歧义，形如：AB 和 A、B 同时出现在词典中，比如：我/ 个人 vs 三/个/人。提高处理歧义的能力是提高分词质量的关键所在。

2.2.3 停用词过滤

本文采用中科院分词系统进行了分词之后，去掉了连词、介词、虚词等没有实际意义的词以及标点符号，但仍然存在一类词，在文档集的每个文档中都过于频繁的出现，甚至有些概率超过了 80%。一般认为这样的词，如果作为特征项的话，对分类没有较大的作用，被称为停用词，也叫无特征词。常见的无特征词有：数字、命名实体、形式词和离合词等。人名、地名、机构名、专业术语等统称为命名实体，“看看”、“看一看”、“打听打听”、“高高兴兴”、“乐呵呵”是形式词，“游了会儿游”、“担什么心”是离合词。这些词在文本中大量出现，如果将其作为特征项的话，即使是内容完全不同的文本也因为这些特征难以区分出来。所以，将这些无特征词从原始文本中过滤掉是非常必要的，称这一过程为停用词过滤。停用词的过滤不仅可以增加区分文本特征的能力，而且可以减少特征词的数目，降低存储空间，节省后续处理的时间和内存资源。

停用词过滤目前有两种方法[35~37]：一种是基于预先建立的停词表，该表包含了所有应该去除的无特征词。该过程是一个简单的词语检索过程，对文本中的没一词条，遍历停词表，删除在停词表中出现过的词条。另一种是基于统计的方法，统计每个词条在文档集中的文档频次，将超过总数某个百分比的词条作为无特征词，并进行过滤。

2.2.4 同义词合并

同义词是指与给定一个词意思相同或相近，但表达形式不同的词语集合。一个词可能存在多个同义词，比如：轿车，在汉语中也可以表达为小汽车、小轿车、房车等等，如果将其概念化，则可以为汽车、车等等。同义词的出现表达多样化，但使文本表示更加分散，同一个概念占据了多个特征项，导致了特征数据的稀疏性。考虑将同一主题下同义词进行合并替换，不仅可以大量减小特征空间，提高处理的效率，而且能够提高聚类系统的准确性。本文采用了同义词合并替换，进行特征空间降维[41]，具体算法见第 4 章节，下面对《同义词词林》作简要介绍。

《同义词词林》目前存在两个版本。旧版本由梅家驹等人于 1983 年编纂而成，词典结构如下[38]：

表 2.1　词典表结构

Ae07 农民 牧民 渔民
农民 农夫 农人 农 庄稼人 庄稼汉 田父 泥腿子 农家 耕夫 老乡
小农 个体农民
佃农 佃户
上中农 富裕中农
** 菜农 棉农 茶农 烟农 蔗农 花农 药农 林农
雇农 贫农 下中农 中农 上中农 富农
自耕农 半自耕农 集体农民 人民公社社员

新版《同义词词林》[39,40]由哈尔滨工业大学信息检索实验室完成，取名《同义词词林扩展版》，在旧版的基础上，剔除了 14,706 个罕见词，目前总词汇量为 77,343 条。新旧《同义词词林》特征对比如下：

表 2.2　同义词扩展前后比较

词典特征	扩展前	扩展后
词条总数	53,895 个	77,343 个
大类数	12 个	12 个
中类数	94 个	97 个
大类数	1438 个	1400 个
小类数	3 层	5 层
编码长度	4	8

旧版《同义词词林》采用长度为 4 的字符串以 3 级编码的方式表示，第 1 位大写英文字母表示大类，第 2 位小写英文字母表示种类，第 3 和 4 位用两位阿拉伯数字表示小类。新版《同义词词林》由长度为 8 的字符串表示，能够唯一的表示词典中的词语。在旧版基础上新增加了第 4 级和第 5 级编码。第 4 级由大写英文字母表示，第 5 级由两位阿拉伯数字表示。为了对第 5 级分类结果进行特别说明，新增第 8 位字符作为标记。第 8 位字符标记有“=”、“#”、“@”，“=”代表“同义”，“#”代表“同类”，“@”代表它在词典中没有同义词和同类词。具体编码表示见下表：

表 2.3　同义词扩展前后比较

编码位	1	2	3	4	5	6	7	8
符号举例	D	a	1	5	B	0	2	=\#\@
符号性质	大类	中类	小类		词群	原子词群		
级别	第一级	第一级	第一级		第一级	第一级		

2.2.5 权重计算

区分文本关键在于找出文本中具有区分能力的重要的特征项，而越是重要的特征项应该权重越大，但也要避免淹没了其他特征，造成信息的过分放大。目前存在两种权重赋值的方法[42]，分别是人工赋予权重、结合文本本身进行权重计算和与所有其他文本进行比较而得出自身的权重。第一种方法是根据专家经验或领域知识或用户需求，进行人工授予权重。该方法适合某些特定的领域，但因为主观性强、效率低下，不能普遍采用。第二种方法根据文章本身的特点计算特征项的权重，该方法的优点是计算简单，处理时间少，缺点是难以代表

其在分类中的重要程度。第三种方法，在第二种方法的基础上，与其他文本的情况进行对比，进而计算出自身的权重，该方法相对能够更准确地代表特征项的区分文本的重要程度。

本文采用的权重计算方法属于第三种，也是目前广泛采用的 TF-IDF 权重计算公式：

$$W(t,\vec{d}) = tf(t,\vec{d}) * \log(N/n_t + 0.01) \tag{2.14}$$

其中，$W(t,\vec{d})$ 表示文本 $\vec{d}$ 中词 t 所占的权重，$tf(t,\vec{d})$ 表示文本 $\vec{d}$ 中词 t 的词频，N 表示语料库中文本的总数，n_t 表示语料库中出现词条 t 的文本数。

2.2.6 特征选择

文本在进行中文分词后，得到词条数目成千上万，如果将分词的结果作为特征项，则必定因特征向量维数过大而导致时间空间复杂度增大。将该向量作为聚类分析输入数据，必定至少导致以下三个问题[43]：

1）高维的数据进行运算，聚类运算效率低下；

2）这些词条存在不少无特征词，会降低聚类质量；

3）由于存在大量干扰特征项，难以区分文本之间的差异。

因此，为了降低特征空间的维数，提高聚类的质量，在进行中文分词，有必要对特征词进行选择。本文在保证特征选择有效的前提下，进行了三次特征选择[44]。1）对分词的结果进行进一步清洗，比如删除纯数值、符号、全角/半角转换等；2）同义词合并替换，提供不同级别的同义词合并，用一个词表示同义、同类的词条；3）权重限值，过滤掉不满足权重条件的词条。大量相关研究和实验证明，有效的特征选择不仅不会降低聚类质量和效率，反而会提高聚类准确度。

2.3 聚类分析

聚类分析[45]是在没有任何监督、没有任何先验知识的情况下，以聚类样本间的相似度作为区分的规则，按照相应的要求和规律进行样本集相互区分的过程。从实质上来讲，聚类是一种无监督的分类，一种特殊化了的分类，其目的都是为了将相似的个体放入相同的群体，并且区分不同个体的过程。聚类分析使用越来越广泛，常应用于模式识别、各种类型的数据挖掘中。目前比较流行

的聚类分析方法有如下几类：

2.3.1 基于划分的算法

基于划分的算法是一种采用迭代优化思想的启发式算法，通过设定一个评估函数迭代地把样本集划分成若干模块。随着样本集合和聚类特征项的增加，在计算上不能完全搜索所有的可能情况，所以该类算法，是以评估函数为依据，反复调整聚类结果，并重新计算每类的聚类中心，直到满足评估函数的要求为止。

典型的划分聚类算法有：K-means、K-medoid、PAM、CLALA 和 CLARANS。下面以 K-means[46,47]为例，介绍基于划分算法聚类分析过程。

K-means 算法基本过程如下：

1）从样本集中随机选择 k 个样本作为聚类中心;

2）计算每个样本与聚类中心的距离，将其划分到距离最小的类中;

3）重新计算聚类中心，若本次划分相对于上次没有任何改变，则聚类过程结束，否则返回上一步。

K-means 算法的优点：1）聚类过程简单；2）聚类速度快；3）当训练样本区分度比较好时，聚类效果好。

K-means 算法的缺点：1）要求用户预先输入聚类数目 k；2）对初始化结果比较敏感，难以达到全局最优解；3）受噪声和异常点的影响非常大；4）难以处理大小不同类和具有凹形的类；5）由于以样本的平均值作为评估函数，所以不适合运用于非数值的聚类分析。

针对 K-means 的缺点，很多学者提出相应的改进办法，比如动态的 K-means 聚类、修改评估函数、结合其他的方法等。

2.3.2 基于层次的算法

基于层次的算法，是考虑类别是有层次的、有级别的，不同的分类粒度可以产生不同粗细程度的聚类结果。该方法将样本集对象组成一颗聚类的树。根据层次分解的方向为自底向上还是自顶向下，可以将层次聚类继续划分为凝聚和分裂层次聚类[13][45]。

凝聚的层次聚类基本描述如下：首先将每一个对象作为一个类，然后合并相似的原子类为更大的类，知道所有的样本对象被合并在同一个类中，或者满

足特定的结束条件。它采用的是自底向上的策略，典型的算法代表是 AGENES（Agglomerative NESting）。

分裂的层次聚类的基本描述如下：首先将所有的样本对象放入一个类中，然后根据样本之间的距离大小，将样本对象细分为越来越小的类，直到每一个对象划分为一个类，或者满足特定的结束条件。它采用的是自顶向上的策略，典型的算法代表是 DIANA（Divisive ANAlysis）。

2.3.3 基于密度的算法

上述提到的基于划分和基于层次的算法都是基于距离的聚类算法，根据评估函数的特点，只能发现特定形状的聚类结果。针对此问题，学者提出了基于密度的聚类算法。该方法把密度高的对象区域作为类，并把低密度区域作为区分类别的类边界。

典型的基于密度的聚类算法有[45][48]：DBSCAN、OPTICS 和 DENCLUE，下面以 DBSCAN 算法为例，对基于密度的聚类算法进行阐述。

DBSCAN 的基本思想是：类的每个对象，以其为中心半径为 R 的区域所包含的样本对象的数目大于等于给定的最小阀值数。即每个对象的在限定空间的对象密度必须大于指定的阀值，并且能够被其他样本对象所确定。

2.3.4 基于模型的算法

基本模型的聚类方法是基于这样一种假设[49]：数据是按照某种特定的概率分布的，它试图找出一种能够与给定数据相适应的数学模型。基于模型的方法根据学习方法的不同，有分为统计学方法和神经网络方法。

概念聚类是以统计学为基础的聚类算法，对于一组为标记的对象，相应的输出一个分类模式。与传统的聚类方法不同之处在于：概念聚类不仅确定对象的分类，而且给定每类对象的一个特征描述，即概念。因此概念聚类分为两个过程：聚类和特征描述。典型的概念聚类包括：COBWEB 和 CLASSIT 等。

神经网络方法首先给出一个模型作为聚类的原型，一个类就是相应的一个原型。新的样本对象根据与模型的距离大小，被分配给与其最相似的类中，并且模型因新样本的加入而得到调整加权。比较著名的神经网络聚类方法有竞争学习和自组织特征映射（SOM），这两种方法都是涉及竞争神经元的方法，SOM 是本文采用的方法，将在下一章节做详细介绍。

2.3.5 基于网格的算法

基于网格的聚类方法是采用空间量化的方法，将样本对象空间化为有限的数目单元，犹如一个分辨率多样化的网络数据结构，并且聚类操作都在网格上进行。

聚类过程与样本对象的数目无关，仅依赖于量化后的维度上的单元数目，因此处理时间短。具有代表性的基于网格的聚类算法有[45][50]：STING、WaveCluster 和 CLIQUE。其中 STING 是基于统计的信息网格聚类算法，采用存储在网格单元的统计信息。STING 具有如下优点：1）基于网格的计算与查询相互独立的；2）可以进行并行处理和增量聚类；3）时间复杂度低。STING 的聚类的精度取决于网格结构最底层的粒度，粒度越细，聚类的代价越高[45]；但粒度越粗，聚类质量越差。STING 的所有类是矩形的，尽管计算速度快，但可能以降低了聚类的精确性作为代价。

2.4 聚类效果评估

文本聚类效果评估[52]是文本聚类过程必不可少的过程，它是验证文本聚类系统是否有效的唯一途径，是反应文本聚类质量高低的关键性措施。目前评价文本聚类有效性指标有 3 个：①查准率（Precision）、②查全率（Recall）和③F-测量值（F-Measure），下面分别介绍之。

假设现有n篇文本 $D_1, D_2, ...D_n$ 分属于m个类别 $C_1, C_2, ..., C_m$。现对这n篇文本分别采用人工和自动聚类程序进行分类，人工分类近似地表示实际文本分类情况。由属性列（真正属于该类的文档数，真正不属于该类的文档数）表示；自动聚类程序分类表示测试文本分类情况，由由属性列（判断为属于该类的文档数，判断为不属于该类的文档数）表示。将人工分类与自动聚类程序组合在一起，则可以表示2*2种文本分类结果，表2.2为组合的文本分类问题的列联表：

表 2.2　文本分类问题的列联表

	真正属于该类的文档数	真正不属于该类的文档数
判断为属于该类的文档数	a	b
判断为不属于该类的文档数	c	d

其中，a 表示聚类程序把文本判断为该类，且真正属于该类的文档数；b 表示聚类程序把文本判断为该类，而真正不属于该类的文档数；c 表示聚类程序不

把文本判断为该类，而真正属于该类的文档数；d 表示聚类程序不把文本判断为该类，且真正不属于该类的文档数。

由以上描述可知，对于类别 C_i,其查准率 P_i、查全率 R_i 与 F-测量值公式总结如下：

1）查准率（Precision）

查准率 P 是判断与实际类别相符文本在所有判断的文本所占的比率，它反映的是聚类算法的准确程度。其数学表达式为：

$$P_i = \frac{\text{检索出的相关记录篇数}}{\text{所有检索出的文档篇数}} = \frac{a}{a+b} \tag{2.15}$$

2）查全率（Recall）

查全率 R 是判断与实际类别相符文本在所有实际的文本所占的比率，它反映的是聚类算法的完全程度。其数学公式为：

$$R_i = \frac{\text{检索出的相关记录篇数}}{\text{所有相关记录篇数}} = \frac{a}{a+c} \tag{2.16}$$

3）F-测量值（F-Measure）

查准率和查全率之间是一种相互制约、此消彼长的关系,准确率的提高往往伴随着查全率的降低，反之，查全率的提高往往伴随着查准率的降低，它们从两个不同的角度反映了分类结果的两个不同的方面。因此为了更好地评价分类结果的优劣，提出一种综合考虑查准率和查全率的评测方法—F-测量值。F-测量值也称综合分类率，数学公式为：

$$F\text{-}Measure_i = \frac{(1+\beta^2)\times P_i \times R_i}{\beta^2 \times P_i + R_i} \tag{2.17}$$

其中 β 是调节查准率和查全率的比例因子，且满足 $\beta \in [0,\infty)$。当 β 趋近于 0 时，F 测量值趋向于与查准率相同；当 β 趋近于 ∞ 时，F 测量值趋向于与查全率相同；当 β =1 时，$F\text{-}Measure_i = \frac{2\times P_i \times R_i}{P_i + R_i}$，它表示查准率和查全率具有同等的重要性，通常为评价分类效果的常用公式。

4）宏观平均是通过计算所有类的 R_i、P_i 的平均值，对分类器进行整体性能评价，它反映的是宏观上分类器的质量。

（1）宏查准率：

$$MP = \frac{1}{n}\sum_{i=1} P_i \tag{2.18}$$

（2）宏查全率：

$$MR = \frac{1}{n}\sum_{i=1} R_i \tag{2.19}$$

（3）宏 F-测量值：

$$MF = \frac{2 \times \mathrm{MR} \times \mathrm{MP}}{\mathrm{MR} + \mathrm{MP}} \tag{2.20}$$

5）微平均是考虑每个文档对分类器整体性能的影响程度，它反映的是微观上分类器的质量。

（1）微查准率：

$$\mathrm{mP} = \frac{\sum a_i}{\sum a_i + \sum b_i} \tag{2.21}$$

（2）微查全率：

$$\mathrm{mR} = \frac{\sum a_i}{\sum a_i + \sum c_i} \tag{2.22}$$

（3）微 F 测量值：

$$mF = \frac{2 \times \mathrm{mR} \times \mathrm{mP}}{\mathrm{mR} + \mathrm{mP}} \tag{2.23}$$

2.5 本章小结

文本聚类主要包括中文预处理、聚类分析以及聚类效果评估。本章介绍了文本聚类的相关概念、一般过程、文本类的定义以及文本相似度计算方法；重点介绍了文本表示、中文分词、停用词过滤、同义词合并、权重计算以及特征选择等中文预处理技术，然后阐述了几种常用的聚类分析算法和聚类效果评估方法。

第 3 章　自组织映射神经网络的研究与改进

3.1 自组织映射神经网络

自组织特征映射（Self-Organizing Maps）网络，由芬兰赫尔辛基大学信息科学系教授 Teuvo Kohonen 在 1981 年提出的[53]，所以又称为 Kohonen 网络，它使用迭代算法优化目标函数来实现对样本集的聚类。SOM 采用一种无监督的学习方法，由环境自行给出输入信号模式[34]。SOM 通过竞争学习，训练权系数，自动得出各聚类的中心，避免了主观因素；支持较大数目的聚类结果，有效地发现异常数据类，网络训练收敛速度较快。因此自组织特征映射在模式识别、图像分割、模式控制等领域取得举足轻重的地位。

3.1.1 SOM 拓扑结构

经典的 SOM 网络拓扑结构由输入层和输出层两层神经元双向连接而成。输入层模拟视网膜感知受外界刺激的信号，并通过连接得方式将信号传送到大脑皮层，而输出层则模拟大脑皮层作出正确的响应。根据输出层神经元排列的不同，SOM 网络拓扑结构可以分为一维网格和二维网格结构。二维网格 SOM 拓扑结构如下所示：

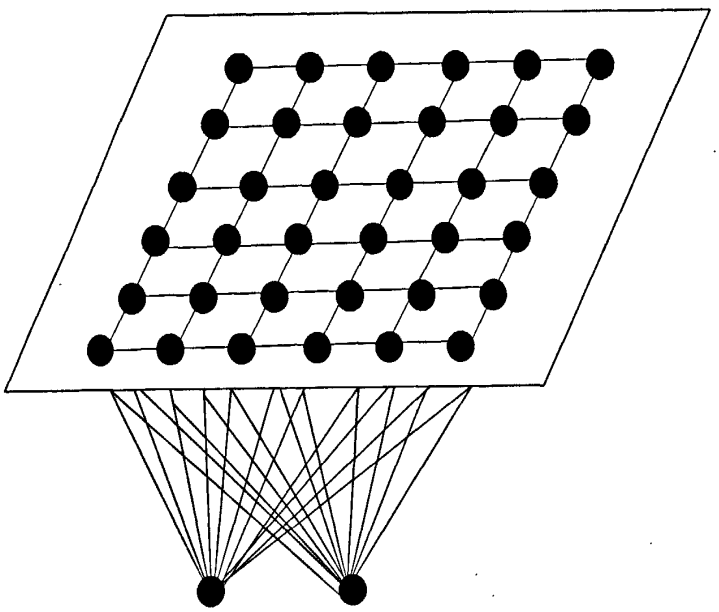

图 3.1　SOM 拓扑结构图

其中输入层由 n 个神经元组成，用矢量 x（$x_1,x_2,...x_n$）表示；输出层由 m 个神经元构成一个二维平面结构；输入层的 n 个神经元与输出层 m 个神经元实现全连接，得到一个 $n*m$ 的二维连接权重矩阵 W，w_{ij} 表示输入层第 i 个神经元与输出层第 j 个之间的连接权值。输入层把输入向量 x 经过连接传送到输出层，输出层根据与输入向量的相似度或者距离来实现竞争，获胜的神经元及其邻域神经元有机会得到学习，对应的连接权向量从而得到调整。输出神经元与其他邻域神经元之间是相互激励的，训练之后的各个神经元就代表了相应的类模式。

3.1.2 SOM 算法描述

SOM 算法流程描述如下：

1）随机初始化每个神经元的权连接矢量 W，设置最大迭代次数 T、初始学习率 $\alpha(0)$、领域半径 $N(0)$；

2）顺序读取输入模式矢量 x；

3）计算输出神经元与输入矢量 x 的距离，寻找获胜神经元 *winner*；

$$winner = \arg\min\left\{\left\|x(t) - w_i(t)\right\|\right\} \tag{3.1}$$

其中 t 代表迭代器次数、x 代表输入矢量、w_i 代表输出层第 i 个神经元的权值；

4）更新获胜神经元 *winner* 及其邻域神经元的权值；

$$\omega_i(t+1) = \omega_i(t) + \alpha(t) * (x_j - \omega_i(t)) \tag{3.2}$$

5）t 自增，调整学习率和邻域半径。

$$\alpha(t) = \alpha(0) * (1 - \frac{t}{T}) \tag{3.3}$$

$$N(t) = INT(N(0) * (1 - \frac{t}{T}) + 1) \tag{3.4}$$

6）重复步骤 *2* 直至 t 达到最大迭代次数 T 为止。

3.1.3 SOM 网络特性

SOM 输出层相邻神经元之间的连线，组成了一层一维或二维网格。网格的维数和神经元的个数，分别远低于输入矢量的维数和个数，但能够准确地描述输入矢量的特征和分布情况。SOM 网络在训练过程中，输出层的神经元并不做

物理上的位置移动，仅仅是调整它们对应的连接权重矢量，因此对输入矢量的学习全部反应在权重矢量的调整过程中。在基于神经网络的聚类中，一个神经元的连接权重矢量可以看作是一个类的“原型点”。可以想象将训练结束后原型点，放回到原始的样本空间中去，然后用连线将相邻的“原型点”联结起来，最后得到的连接图即是映射图。输出神经元在映射图上的逻辑位置由连接权重矢量表示，因而映射具有了拓扑有序性。

自组织特征映射神经网络的特性主要表现在以下几个方面：

1）对输入模式的聚类作用。原始的输入模式由聚类中心来表示，起到了数据压缩的作用，这种压缩能够很好的表示原始输入的特征。

2）保持原始的拓扑有序性。在原始空间中，位置相近的点映射在输出层的神经元是相邻的，即网络在具有相似特征的输入模式的刺激激励下，经过竞争合作的过程后在网络空间中是拓扑有序的。

3）保持原始的概率分布性。输出神经元在空间的分布特性依赖于输入模式在原始空间中的概率分布情况，即样本密度大区域对应的神经元也相对集中，样本密度小的区域对应的神经元相对分散。

4）输出的排序性。相对其他神经元而言，网络中神经元与相邻神经元的之间的距离更小，即越近的神经元的权重向量间的距离越小，越远的神经元的权重向量间的距离越大。

5）容错性。SOM 网络的聚类是通过多个神经元同时作用的，具有较强的容错性。

6）具有自联想性。随着网络的不断训练，神经元能够不断识别周围相似的输入模式，并逐渐吸收他们的特征，因此神经元具有自联想性。

3.1.4 SOM 存在的缺陷

SOM 算法作为无监督的学习方法，具有鲁棒性强、组织性良好等特性，在多个邻域得到广泛的应用和研究。但也存在一些缺陷如下：

1）聚类数目和初始网络结构固定，需要用户预先指定聚类数目和初始的权值矩阵。

2）在训练过程中，有些神经元因无法获胜而永远失去学习的机会，成为“死神经元”。

3）输出结果依赖于样本数据输入顺序，不同的输入序列具有不同的聚类结

果。

4）初始化结果会影响网络的收敛速度，甚至难以达到收敛状态。

针对经典 SOM 存在的上述问题，大量学者针对性地提出了改进方法，详细情况见本文研究现状。

3.2 自组织映射网络的改进

3.2.1 问题的提出

经典自组织映射网络（SOM）的聚类数目必须用户预先设定，并且在训练的过程中不会发生变化，网络初始化在聚类数目确定的基础上，进行随机初始化权值向量，或者随机从样本模式集中随机抽取聚类数目个模式作为初始聚类原型。然而在实际的应用领域中，待训练语料库的类别数目是事先无法预知的，并且随机初始化的聚类中心结点通常会偏离实际的网络结构，在网络的训练过程中，很容易出现“死神经元”和“神经元过度利用”的状况，从而导致系统聚类准确性低和收敛速度缓慢。

针对经典 SOM 算法聚类数目需预先输入、网络结构固定、随机初始化效果差的问题以及聚类结果依赖样本输入顺序，提出一种改进的自增长 SOM 聚类算法模型。采用生长阀值确定聚类数目和网络结构的增长以及聚类中心的初始化，形成初始化聚类中心与实际样本空间分布一致；在网络结构形成之后，采用并行学习策略，并行输入所有样本，对网络并行调整，实现调整结果与样本输入顺序无关。

3.2.2 网络拓扑结构

改进的 SOM 网络拓扑结构由输入层和输出层两层神经元双向连接而成。与经典的 SOM 不同之处在于，输出层是一种树形结构[26]。初始网络只有 1 个神经元，如左图所示。随着网络训练的进行，网络结构初步形成，输出层具有多个神经元，构成 1 棵多神经元树，如右图所示，其中输出层节点与输入层的全互相连，以 *root* 作为根节点，构成一个二维结构的树形网络。

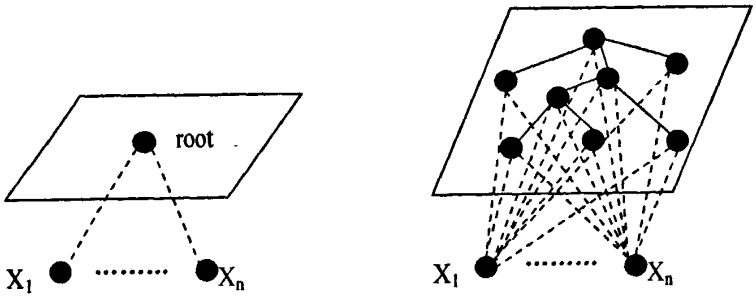

图 3.2　改进 SOM 网络结构拓扑图

3.2.3 关键因子设定

改进 SOM 算法的第一个优点是不需预先输入准确的聚类数目。在网络结构形成最初，网络中只有 1 个神经元节点，即聚类数目为 1；然后随着样本的不断输入，网络对通过对样本的学习，实现自身的生长或调整，直到网络结构的形成。第二个优点是聚类结果不依赖样本输入顺序。在网络结构形成后，进入网络的调整阶段，经典 SOM 采用传统的逐个样本学习策略，当样本规模较小时，调整结果依赖输入序列；改进的 SOM 采用并行学习策略，对网络进行并行调整，实现调整结果与输入序列无关。改进 SOM 的网络形成和网络调整优于经典 SOM，分别由本文的生长阀值和学习率两关键个因子决定。

1）生长阈值

设输入的样本矢量为 x，标记 x 与当前各聚类中心结点相似度最大者为获胜神经元，x 与获胜神经元的距离记为 DT，即

$$DT(x) = \min\{1 - sim(x,o)\}, o \in P \tag{3.5}$$

其中，P 为聚类中心结点集。

改进 SOM 算法根据生长阈值因子 GT 与 $DT\ (x)$ 的大小关系，判断 x 是否成为网络中的新结点。x 为新结点当且仅当 $DT\ (x) > GT$，否则网络对 x 学习，进行权向量调整。GT 值设定的大小直接关系着网络的生长速度和规模。当 GT 值较大时，新输入矢量成为新节点的条件严格，此时网络包含的节点数较少，实现粗粒度的聚类；反之，当 GT 值较小时，新输入矢量成为新节点的条件宽松，此时网络包含的节点数较多，可以实现细粒度的聚类。为兼顾网络生长速度与聚类的粒度需求，本文采用动态的生长阀值 GT，数学公式如下所示：

$$GT = \left[1 - (n(t)/\max)^2\right] / \left[1 + 1/n(t)\right] \tag{3.6}$$

其中 max 表示预先输入的最大聚类数目，用来限制当前聚类数目的增长范

围，取值应尽量接近实际的聚类数目；n（t）表示第 t 时刻当前聚类数目，当 n（t）较小时，生长阈值 GT 值较小，节点增加的机会较大；当 n（t）较大时，GT 值较大，节点增长的机会减小。

2）学习率

在经典 SOM 算法中，学习率依赖于初始值和迭代次数，与聚类中心矢量与输入样本矢量的相似度无关，即一旦当前输入样本的获胜节点与邻域节点确定后，这些节点以同样的速度进行调整；网络调整随样本逐个输入，每次只对当前输入的样本进行学习，不同的样本输入顺序，产生不同的聚类中心矢量。

模糊 C-均值（FCM）是目前常用的聚类分析方法，收敛于代价函数最小值，具有聚类准确性高，学习并行进行的优点。本文针对经典 SOM 学习过程中存在的问题，结合 SOM 自组织和 FCM 并行学习的优点，提出一种并行的改进的 SOM 学习方法。

FCM 代价函数一般定义为聚类集中每一个样本点到该类中心的距离平方和，数学公式如下所示：

$$J_m(U,P)=\sum_{i=1}^{n}\sum_{j=1}^{C}\mu_{ij}{}^{m}\left\|x_i-P_j\right\|^2 \tag{3.7}$$

其中 x_i 表示第 i 个样本矢量（$1\le i\le n$，n 为样本数），P_j 为第 j 类的聚类中心矢量（$1\le j\le C$，C 为聚类总数），$m>1$ 为模糊加权指数。μ_{ij} 为 x_i 在 j 类别中的隶属度。U 和 P 分别为$\{\mu_{ij}\}$和$\{P_j\}$构成的矩阵。FCM 就是要求出合适的 U 和 P，使 Jm（U，P）值最小。

模糊加权指数 m_t 计算公式如下：

$$m_t=m_0-t*(m_0-1)/t_{\max} \tag{3.8}$$

其中，表示迭代次数 t；m_0 是模糊加权指数初始值；$t_{\max}$ 表示最大迭代次数。

$J_m(U,P)$ 取最小值当且仅当：

$$\begin{cases}\dfrac{\partial J_m(U,P)}{\partial U}=0\\[2ex]\dfrac{\partial J_m(U,P)}{\partial P}=0\end{cases} \tag{3.9}$$

求解式（3.9），可以得到 μ_{ij} 定义如下所示：

$$\mu_{ij}=\begin{cases}\left[\sum_{k=1}^{C}\left(\dfrac{\|x_i-p_j\|}{\|x_i-p_k\|}\right)^{2/(m_t-1)}\right]^{-1} & p_k\neq x_i\\ 1 & p_j=x_i\\ 0 & p_j=x_q,q\neq i\end{cases} \tag{3.10}$$

学习率$\eta_{ij}(t)$，其涵义为迭代次数为t，聚类中心j对样本i的学习率，数学公式如下所示：

$$\eta_{ij}(t)=\eta(t)(\mu_{ij})^{m_t} \tag{3.11}$$

其中$\eta(t)=(1-t/t_{\max})$，$\eta_{ij}(t)$的收敛速度要快于一般的学习率$\eta(t)$的速度，在文献[27]中已经证明；

改进SOM聚类矢量调整公式如下所示：

$$P_j(t+1)=P_j(t)+\eta_{ij}(t)*(x_i-P_j(t)) \tag{3.12}$$

其中，$P_j(t)$和$P_j(t+1)$表示迭代次数为t，t+1时，第j个聚类中心矢量（j=1，2 ...C）。

3.2.4 改进算法描述

改进SOM算法主体上分为网络初始化和网络调整两个阶段。网络初始化主要完成两个任务：（1）关键因子的初始化，以及初始化根神经元节点的权重向量；（2）根据输入向量x与聚类中心最小距离DT（x）与生长阀值的大小比值，决定x是否为新聚类中心节点，直到所有样本输入完毕。网络调整是网络聚类中心矢量调整的过程，根据输入样本隶属聚类中心各节点的程度不同，实现输入样本对网络聚类中心矢量并行地不同程度的调整。

算法描述如下：

输入：训练样本、最大聚类数目和初始学习率$\eta(0)$

输出：聚类数目C和聚类中心矩阵P。

（1）网络初始化

①设定模糊参数m_0、迭代次数t_{max}和终止迭代误差ε；

②将训练样本集X中各矢量标准化至[0,1]区间；

③从训练样本集 X 中随机抽取 1 样本作为根节点 $root$。

④从训练样本集 X 中顺序读取训练样本矢量 x_i；

⑤根据相似度公式找到与 x_i 相似度最大的神经元作为获胜神经元；

⑥按公式（3.5）计算 x_i 与获胜神经元的距离差 DT：若 $DT > GT$ ，转步骤⑦；否则转步骤⑧；

⑦将 x_i 添加到当前聚类中心集 P 中；

⑧返回步骤④直到样本集 X 中所有样本输入完毕；

（2）网络调整

①按照公式（3.10），计算样本隶属度矩阵 $\mu_{ij}(t)$

②按照公式（3.11），计算样本学习率 $\eta_{ij}(t)$

③按照公式（3.12）调整 C 个聚类中心矢量的权值

④返回①直到计算误差小于终止迭代误差 ε 或者迭代次数达到 t_{max}.

⑤存储聚类数目 C 和聚类中心矩阵 P。

（3）分类

①测试样本集 V 中各矢量标准化至[0,1]区间。

②从 V 中顺序读取测试样本向量 v。

③计算 v 与聚类中心矢量的相似度 sim（v,j） $j=1，2...C$;计算公式如下：

$$\mathrm{sim}(v,j)=\sum_{i=1}^{m} v_i \times p_{ji} \Big/ \sqrt{\sum_{i=1}^{m}(v_i)2\sum_{i=1}^{m}(p_{ji})2} \qquad (3.13)$$

④ 求 sim（v,j）的最大值，输出最大值对应的类标识符 j。

⑤ 返回②直到所有的数据测试完毕。

3.3 算法分析

本节以示例样本空间为例，根据网络初始化效果是否理想以及聚类结果是否依赖输入序列两方面，进行本文算法与经典 SOM 算法的分析与对比。

假设样本空间 $D=\{I1,I2,...I9\}$，划分为 A、B、C 三类，其中 $I1,I2,I3$ 属于类 A；$I4,I5,I6$ 属于类 B；$I7,I8,I9$ 属于类 C。样本空间分布情况如下所示：

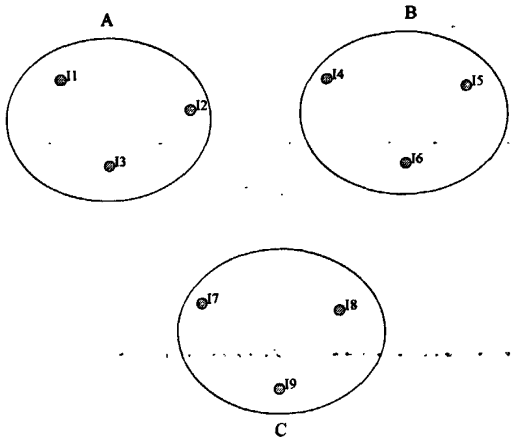

图 3.3　实际样本空间分布图

（1）网络初始化效果是否理想

经典 SOM 采用随机从样本空间 D 中选取 3 个样本进行网络初始化，这种方法优点是聚类中心必定落在样本空间范围内，但缺点是可能会出现某类空间初始化了多个聚类中心，而某类空间没有聚类中心的情况，如下图所示。

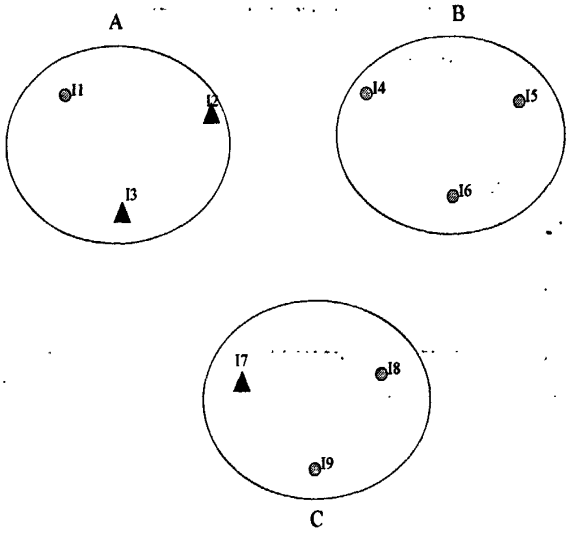

图 3.4　经典 SOM 初始化结果

改进 SOM 算法的网络初始化是一个动态的过程，随网络结构的生长而同步进行的。刚开始网络只初始化了 1 个节点，然后随着网络的训练，通过计算输入的样本节点与获胜神经元节点的距离是否大于生长阀值，判定它是否满足新节点的条件，所以在初始化完成后网络中的节点是分布比较均匀的，基本与样本空间分布一致。下图是改进 SOM 网络初始化结果示意图：

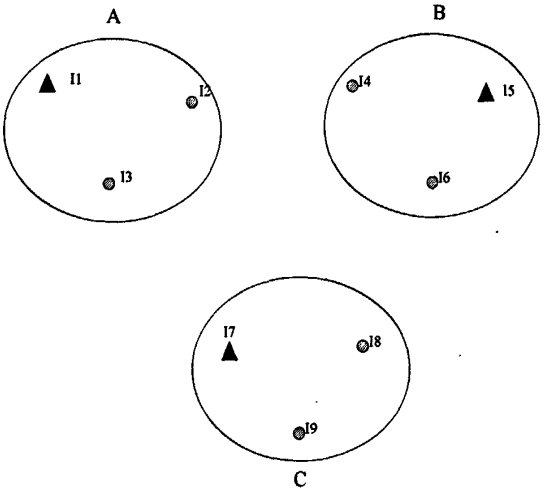

图 3.5　改进 SOM 初始化结果

由图 3.4 和图 3.5 网络初始化结果对比可知，改进的 SOM 网络初始化效果优于经典 SOM。

（2）聚类结果是否依赖样本输入序列

假设某时刻聚类中心在样本空间中的映射如下图所示。其中 c1 和 c2 分别是类空间 A 和 B 的聚类中心节点，*I2* 和 *I4* 是网络待学习的两个样本，现通过改变 *I2* 和 *I4* 输入顺序，来判定聚类结果是否依赖样本输入序列。

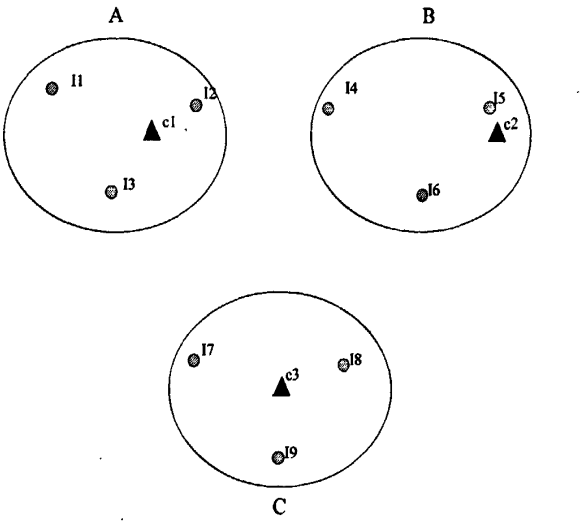

图 3.6　时刻聚类中心在样本空间中的映射

考虑经典 SOM 算法：若先输入 *I2*，则 *A* 聚类中心 c1 得到调整，并且向 *I2* 的方向进行偏靠，然后在输入 *I4*，此时 *I4* 与 c1 的距离反而比 *I4* 与 c2 的距离更

近，导致 *I4* 被归为 *A* 类，c1 再次得到调整；相反若先输入 *I4*，则 *B* 聚类中心 c2 得到调整，并向 *I4* 的方向靠拢，然后输入 *I2*，此时 *I1* 与 c2 的距离反而比 *I2* 与 c1 的距离更近，导致 *I2* 被归为 *B* 类，c2 再次得到调整。因此，经典 SOM 算法，聚类结果可能依赖样本输入的序列。

至于改进的 SOM 算法，首先计算 *I2*、*I4* 以及其它所有样本隶属当前所有聚类中心的程度，然后根据隶属度确定每个聚类中心对每个输入矢量的独立的权向量调整公式，因此调整的结果与 *I2* 与 *I4* 及所有其它样本输入的顺序无关。

综上述对比分析，可以得出如下结论：经典 SOM 算法，聚类结果可能依赖样本输入的顺序；改进的 SOM 算法聚类结果与样本输入的顺序无关。

3.4 本章小结

本章介绍了经典 SOM 的相关理论知识，包括网络拓扑结构、算法描述、网络特性以及 SOM 存在的缺陷。然后针对 SOM 聚类数目需预先输入、网络结构固定、随机初始化效果差的问题以及聚类结果依赖样本输入顺序，本章提出一种改进的自增长 SOM 聚类算法。采用生长阈值确定聚类数目和网络结构的增长以及聚类中心的初始化，形成初始化聚类中心与实际样本空间分布一致；在网络结构形成之后，采用并行学习策略，并行输入所有样本，对网络并行调整，实现调整结果与样本输入顺序无关。最后对改进 SOM 算法前后进行了算法对比分析，在网络形成和网络调整阶段，分别分析论证了改进 SOM 算法的优越之处。

第 4 章　文本聚类系统的设计与实现

为验证改进 SOM 算法的有效性和实用性，本文基于改进的 SOM 算法设计实现了《中文文本聚类系统》平台。

4.1 系统流程图

根据模块化设计思想，系统划分为三个主要功能模块：预处理模块、聚类分析模块和输出模块。预处理模块是聚类分析的数据准备阶段，主要实现文本的中文分词和特征降维；聚类分析模块是本系统的核心模块，采用改进的 SOM 算法，实现样本集的自动聚类；结果输出模块实现聚类结果输出以及持久化。系统流程图如下所示：

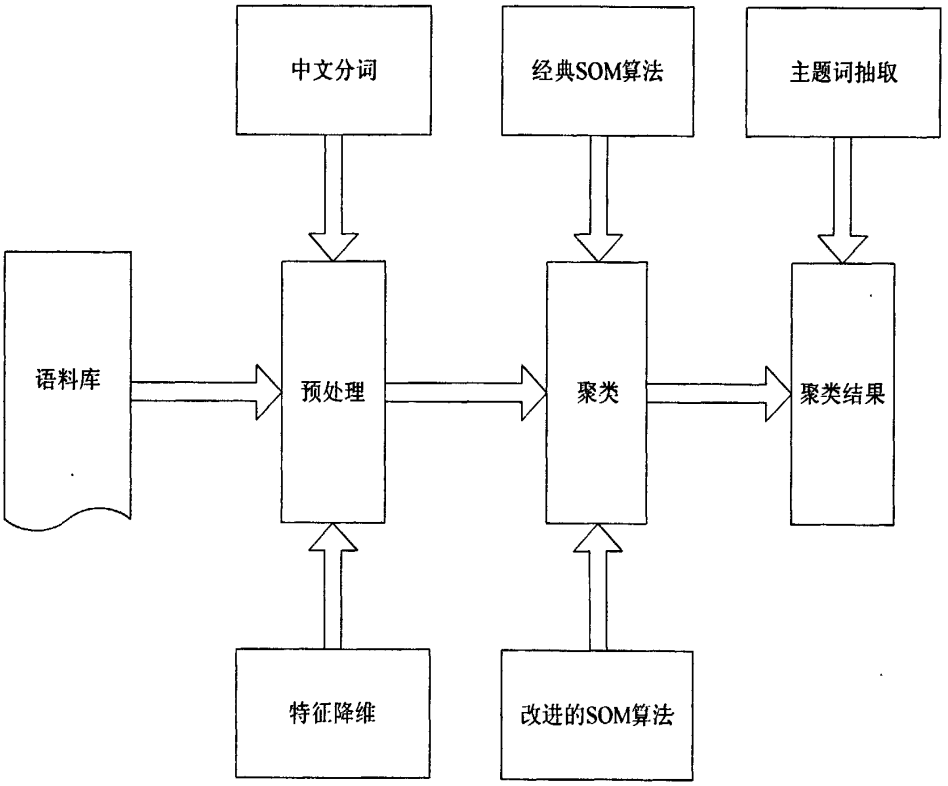

图 4.1　系统流程图

4.2 功能模块

系统主要是包括中文分词、特征降维、聚类分析和结果输出四个功能模块。

4.2.1 中文分词

系统采用了中科院计算机所研发的中文分词系统 ICTCLAS 的 C#版本，该版本是由河北理工大学经管学院吕震宇在 FREE 版本基础上改编而来。ICTCLAS 提供了中文分词、未登录词识别、新词识别和词性标注，同时可以支持用户自定义字典的功能。

类 ChineseSpliter 实现了 ICTCLAS 中文分词过程的调用，其静态结构如下所示：

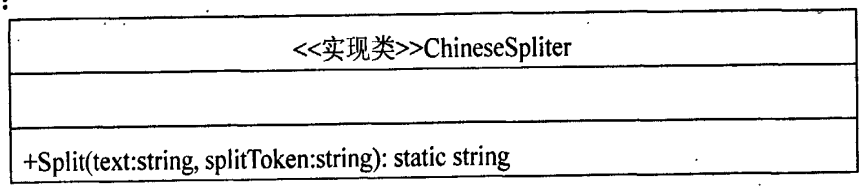

图 4.2　ChineseSpliter 静态类图

其中，静态方法 Split 封装实现了 ICTCLAS 实现分词的具体过程。text 参数表示待切分的文本；splitToken 表示分词的结果分隔符。

4.2.2 特征降维

本系统采用了停用词过滤、同义词合并和权重过滤三种方式进行特征降维。

（1）停用词过滤

停用词即是存在于文本中没有区分文本的特征的词，该类词频繁地出现在原始文本中，严重影响文本特征表示的质量，因此我们务必把这些词过滤掉。停用词过滤的流程如下所示：

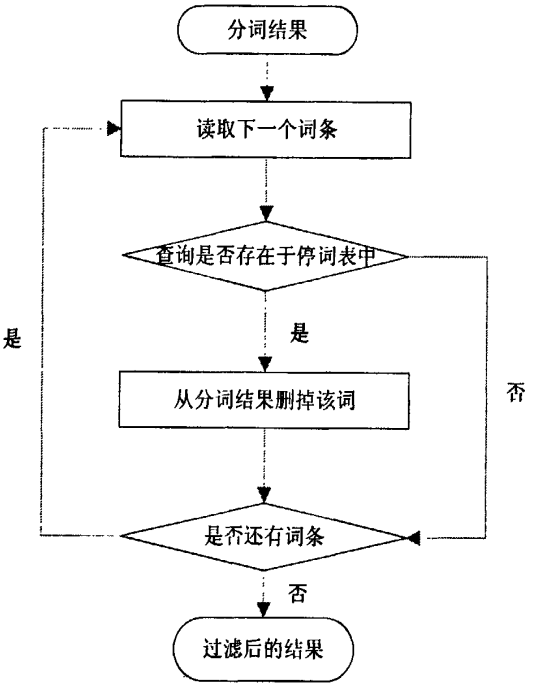

图 4.3　停用词过滤的流程图

类 StopWordsHandler 实现了本系统的停用词过滤功能，其静态类图如下所示：

<<实现类>>StopWordsHandler
-NoisePath:static string = "\\data\\stopwords.txt" -stopwords:HashSet<string> = null
+StopWordsHandler( ): static StopWordsHandler -IsStopWords(word:string):static bool +Parse(words:string[]):static string[]

图 4.4　StopWordsHandler 静态类图

其中，NoisePath 存储停用词表文件的路径；stopwords 是 HashSet 数据类型，存储加载停用词表文件后的具体数据；StopWordsHandler 静态构造方法，实现了读取停用词文件，将数据存储在 stopwords 内存对象中；IsStopWords 实现判断指定单词 word 是否是停用词；Parse 实现对指定的词集合停用词过滤，返回过滤后的结果。

（2）同义词合并

文本中存在很多具有意思相近或属于同类的词，这类词以多种表达形式存

在于文本中。如果把它们都作为特征项，需要多个特征项表示同一个意思的词条，造成存储空间浪费；特征项维数过大，导致计算负载过大、特征空间过于稀疏而使聚类的质量降低。同义词合并的流程如下所示：

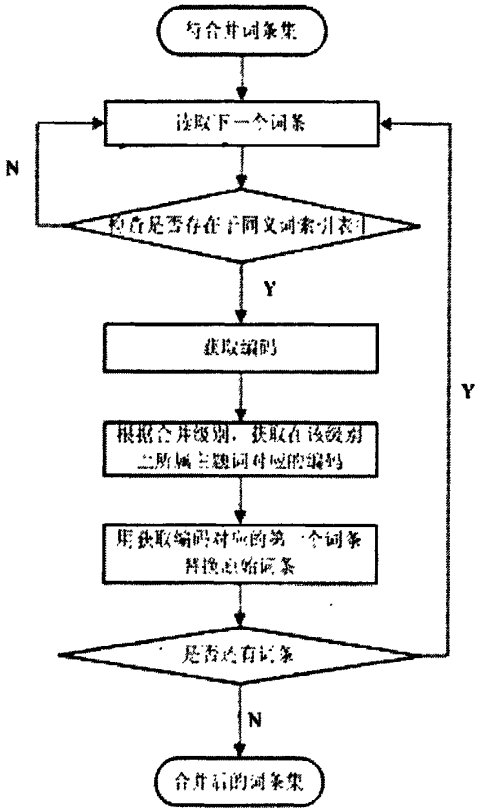

图 4.5　同义词替换的流程图

合并过程的《同义词词林》包括同义词索引文件（improvedThesaurus.index）和同义词数据文件（improvedThesaurus.data）。类 SynonymsHandler 实现了同义词合并的功能，其类静态结构如下所示：

<<实现类>>SynonymsHandler
-SynonymsIndex:static string = "\\SynonymsDic\\improvedThesaurus.index" -SynonymsData:static string = "\\SynonymsDic\\improvedThesaurus.data" -synonymsIndex:HashTable = null -synonymsData:HashTable = null
+SynonymsHandler( ): static SynonymsHandler +Parse(terms:string[],level:int):static string[]

图 4.6　SynonymsHandler 静态类图

其中，SynonymsIndex 存储同义词索引文件的路径；SynonymsData 存储同义词数据文件路径；synonymsIndex 是同义词索引的内存对象表示；synonymsData 是同义词数据的内存对象表示；SynonymsHandler 是静态构造方法，实现自动加载索引文件和数据文件到内存对象中，以便提高处理的速度；Parse 方法实现了对词集合，在 level 级别上的同义词合并替换。

（3）权重计算和特征选择

权重计算是计算词条在文本中的 TF*IDF 权重，权重计算流程如下图所示：

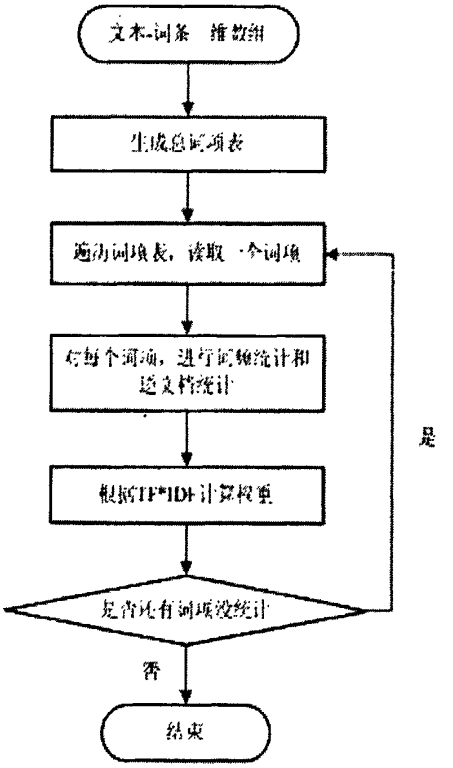

图 4.7　权重计算的流程图

类 TFIDFMeasure 实现了权重计算的功能，其静态结构如下所示：

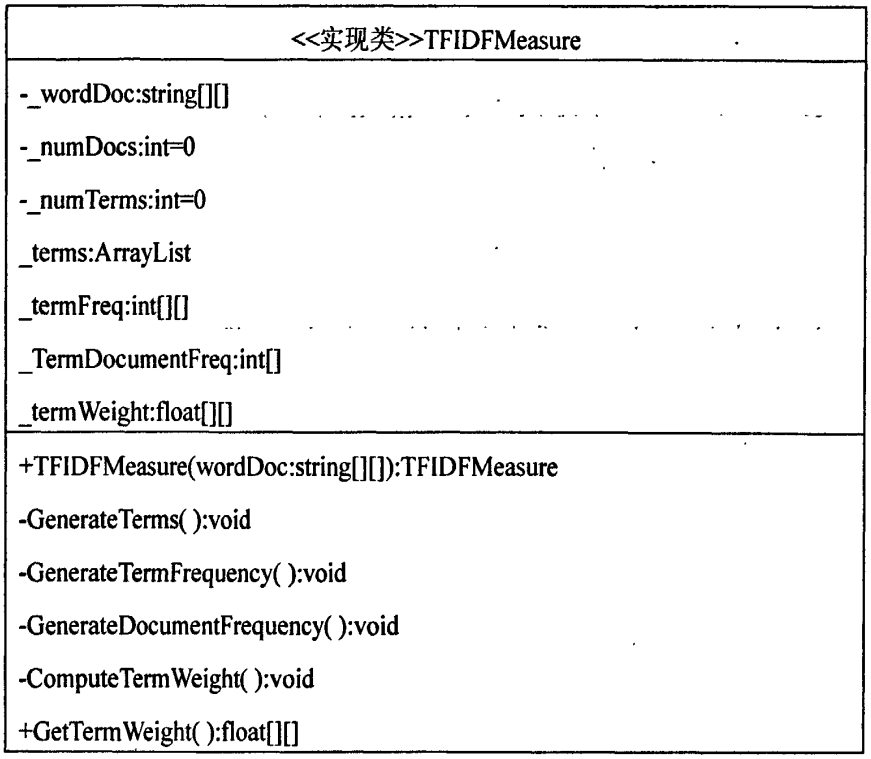

图 4.8　TFIDFMeasure 静态类图

其中_wordDoc 表示文本-词条二维字符串数组，存储文本集中每个文本进行中文分词、停用词过滤和同义词合并后的词条集；_numDocs 存储文本集的文本数；_numTerms 存储文本集中词条总数；_terms 存储文本集中的词条集；_termFreq 存储词条集的词条在各个文本中的词频；_termWeight 存储词条集中词的 TF*IDF 权重；_TermDocumentFreq 存储词条集的词的文档频；TFIDFMeasure 是构造方法，负责类的初始化和成员方法的调用；GenerateTerms 方法实现了合并文本-词条二维字符串数组中的词条，生成覆盖了所有文本词条并且唯一的词条集合；GenerateTermFrequency 方法实现了统一每个词条在每个文本中的词频；GenerateDocumentFrequency 方法实现了统计每个词条的文档频；ComputeTermWeight 方法实现了计算每个词条的权重；GetTermWeight 方法实现了返回文本-词条权重矩阵。

权重计算的输出是二维的文本-词条权重矩阵，作为特征选择的数据输入，特征选择流程如下图所示：

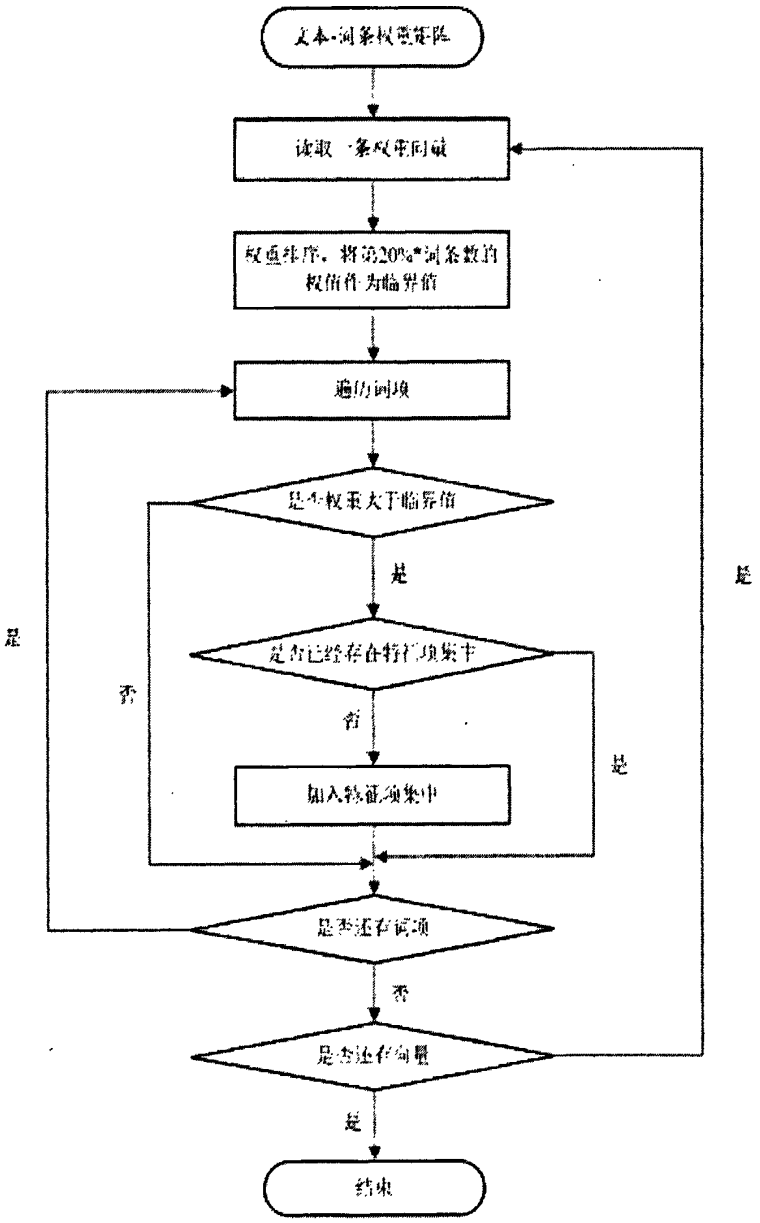

图 4.9　特征选择的流程图

类 FeatureSelector 实现了权重过滤方式的特征选择功能，其静态类结构如下所示：

<<实现类>>FeatureSelector
-_numFeatureTerms:int=0 -_ featureTerms:ArrayList -_data:float[][] -_wordsIndex:Hashtable
+FeatureSelector(tf:TFIDFMeasure,rate:float):FeatureSelector -GenerateFeatureTerms( ):void +GetTrainingData( ):float[][] +GetTrainingData2( ):double[][]

图 4.10　FeatureSelector 静态类图

其中_numFeatureTerms 存储特征词的数目；_featureTerms 存储特征词内容；_data 存储特征词的权值；_wordsIndex 存储特征词对应的索引编号；FeatureSelector 构造方法实现类的初始化，确定特征选择的数据来源和特征选择的比例；GenerateFeatureTerms 方法实现生成特征项；GetTrainingData 方法和 GetTrainingData2 方法分别实现以 float 和 double 的精度返回特征选择后的特征矩阵。

4.2.3 聚类分析

改进 SOM 算法的训练主要包括网络初始化阶段、调整阶段和分类阶段。

（1） 网络初始化

初始化阶段实现聚类数目和网络结构的确定，即初始化网络的聚类中心。初始化阶段的算法流程如下所示：

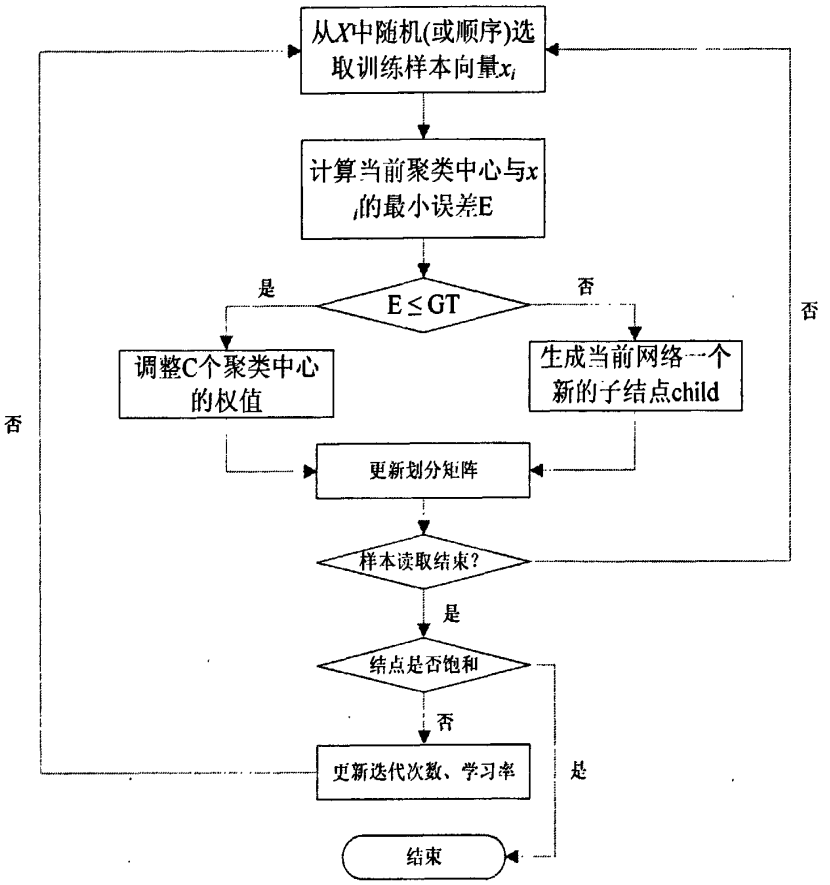

图 4.11　改进 SOM 初始化阶段的流程图

（2） 网络调整

网络调整阶段是网络对输入样本的学习，自动调整的阶段。在该阶段聚类中心进行不断调整，直到网络达到稳定状态。算法流程如下图所示：

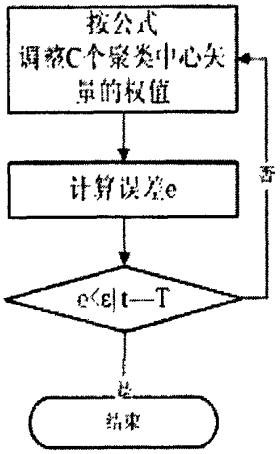

图 4.12　改进 SOM 调整阶段的流程图

类 DSOM 实现了系统的聚类分析功能，其静态结构如下所示：

<<实现类>>DSOM
-_coordCount:int=0 -_ coordinates:double[][] -_k:int -_numClusters:int -_clusters:Cluster[] -_nearestCluster:int[] -_learningRate:float -_learningRadius:float
+DSOM(data:double[][],k:int, learningRate:float):SOM -InitRandom( ):void +Start( ):void +GetDistance(i:double[],center:double[]):double -nearestCluster(ndx:int):int

图 4.13　DSOM 静态类图

其中，_numClusters 存储实际聚类数；_coordCount 存储样本的数目；_k 存储最大聚类数；_ coordinates 存储样本数据；_k 存储聚类的数目；_clusters 存储聚类中心；_nearestCluster 存储距离最近的节点编号；_learningRate 存储学习率；_learningRadius 存储学习半径；DSOM 构造方法实现数据的初始化，以及成员方法的调用；InitRandom 方法实现网络的初始化；Start 实现聚类分析过程；GetDistance 方法实现计算两个节点之间的距离；nearestCluster 方法实现获取指定节点的最近节点。

类 Cluster 实现了对聚类结果的存储功能，其静态类结构如下所示：

<<实现类>>Cluster
-Mean:double[] -CurrentMembership:List<int>
+Cluster(length:int):Cluster +Cluster(dataindex:int,data:double[]):Cluster

图 4.14　Cluster 静态类图

其中 Mean 是 double 类型的数组，存储聚类中心；CurrentMembership 存储数据成员的索引。

4.2.4 输出结果

系统将输出作为单独的模块，与预处理和聚类分析分离开来，使其具有更好的重用性和松耦合度。

为了能够重复利用中间结果以及用户可以随时查看处理的结果，本文以文件的形式存储中间结果和聚类结果。中文分词需要进行频繁的字符串处理，耗费大量的时间，因此有必要保存中文分词的中间结果，作为特征选择的数据输入。同样其他预处理模块的中间结果也有必要保存，尤其是特征选择后的特征词集和权重矩阵。

XML（Extensible Markup Language）即可扩展标记语言，具有语法严格、操作简单等优点，广泛用于数据存储、数据交换和处理半结构化信息。本文将采用 xml 文件用来保存分词、特征选择和聚类的结果。

输出模块由 OutputHandler 和 XMLHandler 两个类来完成。OutputHandler 实现中间结果和聚类结果的输出；XMLHandler 实现 xml 文件的读写处理。OutputHandler 类和 XMLHandler 类的静态类图如下所示：

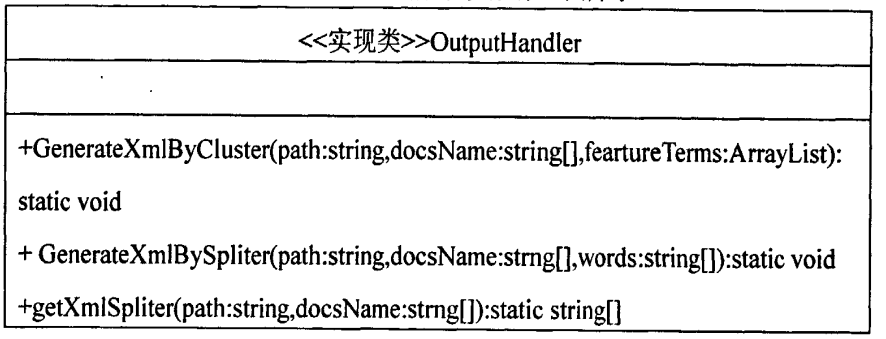

图 4.15　OutputHandler 静态类图

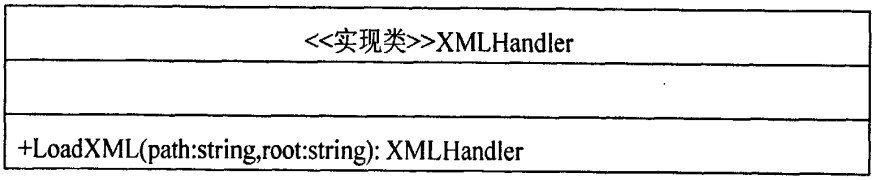

图 4.16　XMLHandler 静态类图

4.3 运行情况

系统运行步骤主要包括：选择训练语料库、中文分词、特征选择和聚类分析四个过程。

（1）选择训练语料库。选择准备的语料库，并加载到《中文文本聚类系统》平台。

（2）中文分词。加载语料库成功后，执行分词操作。

图 4.17 和图 4.18 分别是交通类 4150.txt 分词前后的内容。中文分词操作实现了将原始文本划分为以‘|’作为分隔符的词条集合；并且同时进行了段落合并、空格删除和全角/半角转换的相关工作。图 4.19 是分词结果的 xml 文件存储形式。

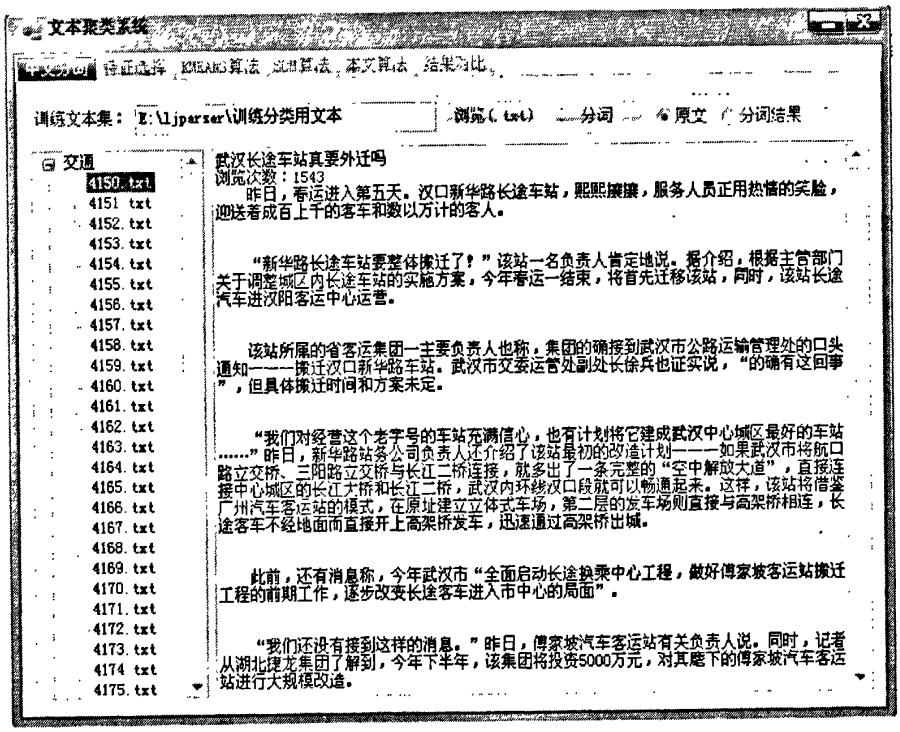

图 4.17　文本原文

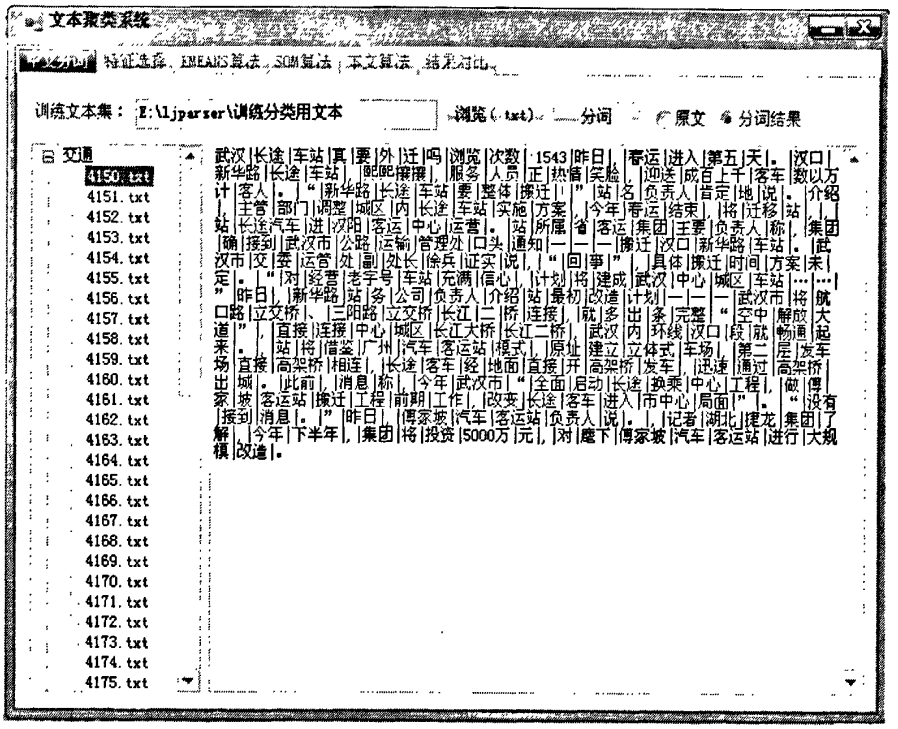

图 4.18　文本分词结果

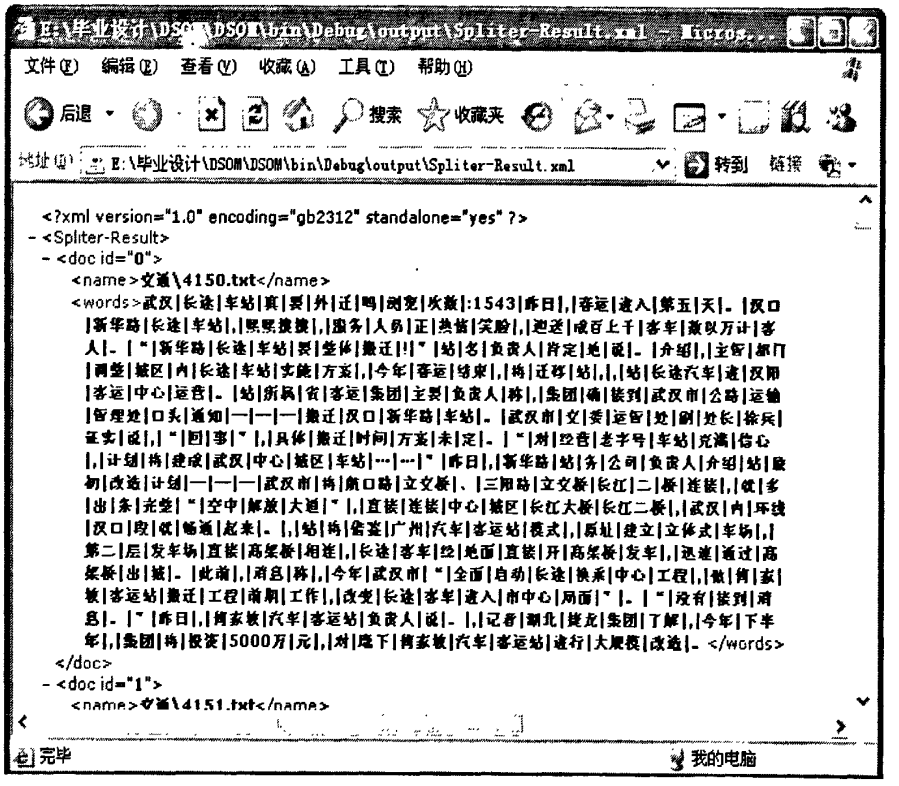

图 4.19　分词结果的 xml 存储

（3）特征选择

特征选择包括停用词过滤、TF*IDF 权重计算与过滤和同义词合并 3 种方式，并且可以组合的方式进行特征选择。图 4.20 显示了采用停用词过滤、TTF*IDF 权重计算与过滤和第 4 级同义词合并的组合方式的特征选择结果，总耗费时间 25.859375 秒，生成的特征词数是 1758。

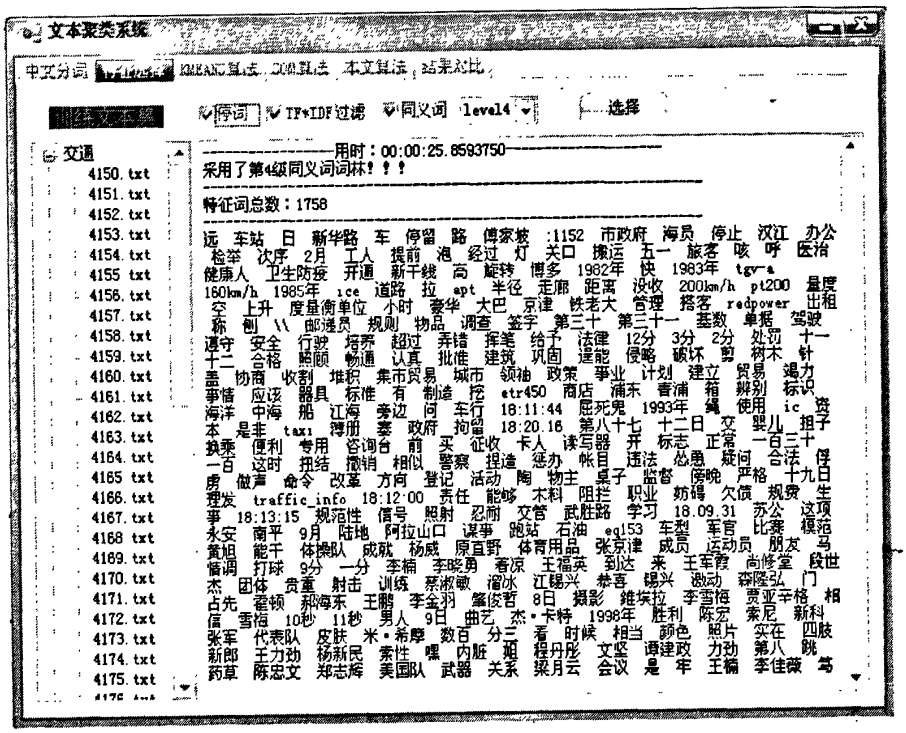

图 4.20　特征选择结果

（4）聚类分析

聚类分析在特征选择结束后进行，实现样本数据集的自动分类。图 4.21 和图 4.22 分别显示了改进 SOM 算法的聚类结果及其 xml 存储形式。输出内容主要包括文本聚类的数目、类编号、类的特征表示以及每个类包含的文本对象。

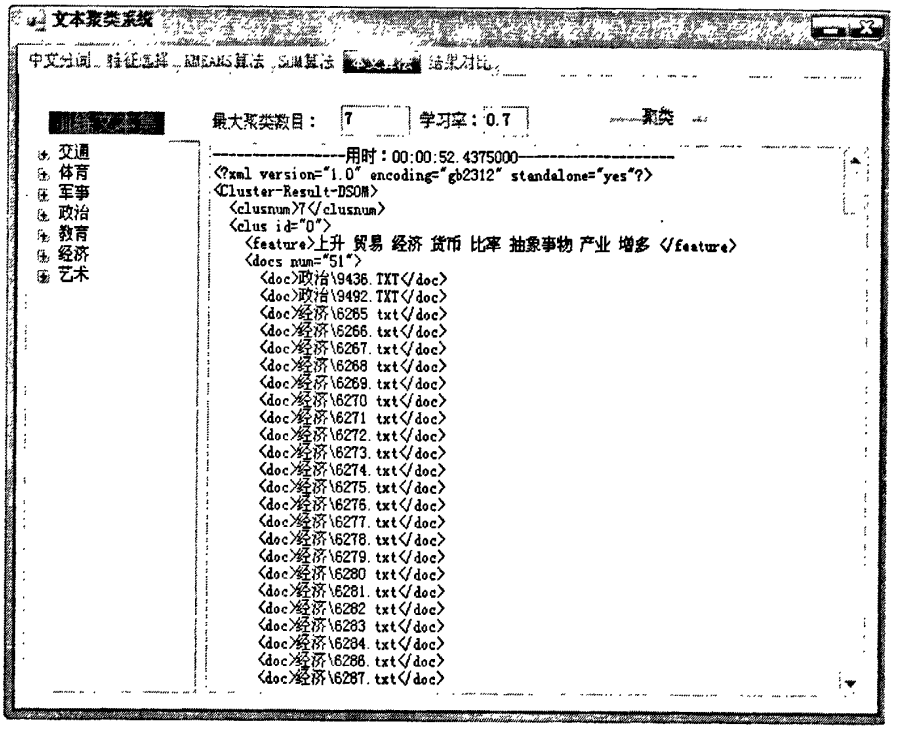

图 4.21 本文算法聚类分析结果

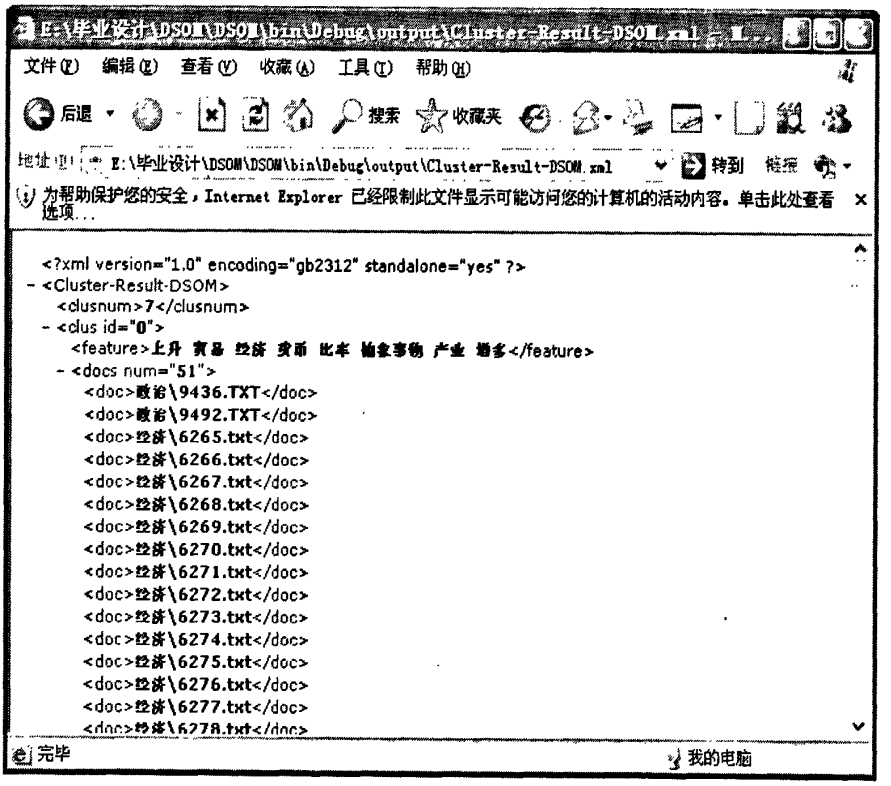

图 4.22 本文算法聚类分析的 xml 存储形式

4.4 本章小结

本章在改进 SOM 算法的基础上，设计实现了《中文文本聚类系统》平台。根据模块化设计思想，系统划分为预处理模块、聚类分析模块和输出模块。预处理模块是数据准备模块，包括中文分词和特征降维；聚类分析模块是系统的核心模块，实现样本数据集的自动聚类；输出模块实现中间结果、聚类结果的输出。本章最后介绍了系统的运行步骤，以及展示了运行结果。

第 5 章　实验结果与分析

5.1 实验环境

本文实现的文本聚类系统的实验环境如下：

CPU：Intel（R） Core Duo CPU T6600@2.20GHz；

内存：2G DDRIII；

操作系统：Windows XP SP3；

开发环境：Microsoft Visual Studio.Net 2008.

5.2 语料来源

实验所使用的语料库由“灵玖”中科软件（北京）有限公司为《灵玖 LJParser 自然语言理解中间件》提供的训练测试语料（下载网页地址是：http://www.lingjoin.com/product/ljparser.html），该语料文本由 7 类文本组成，类别分布情况如下所示：

表 5.1　文本分布情况

文本类别	训练样本数
交通	50
体育	50
艺术	50
教育	50
经济	50
政治	50
军事	50
合 计	350

5.3 测试评估

在《中文文本聚类系统》平台设计实现后，我们需要对平台性能、相关引用技术和改进方案进行准确的评估，确定平台性能是否良好，引用的技术是否可以改善系统的性能或者有何关联，改进的算法是否达到预期效果，从而对该

聚类模型进行推行或进一步的研究。

本章将分两个部分进行实验结果评估：一、同义词合并技术对中文文本聚类系统的性能改善评估；二、基于改进 SOM 前后的中文文本聚类系统的性能对比评估。

（1）同义词合并技术对中文文本聚类系统的性能改善评估

同义词合并技术是本文引用的特征选择技术，主要实现将文本中词义相近或者同类词进行合并替换，采用 1 个词来表示多个同义词。本实验将停用词过滤和 TF*IDF 权重过滤作为特征选择的必选项，同义词合并作为可选项（特征选择的顺序：停用词过滤—>同义词合并—>TF*IDF 权重过滤），记录同义词原始级别（即不采用同义词合并）、Level5（最低级别合并）、Level4、Level3（最高级别合并）运行情况。表 5.2 是四种级别同义词合并，特征选择的效果对比情况：

表 5.2　不同级别同义词合并特征选择效果对比表

同义词级别	特征词数目	耗费时间（S）	降维度（%）
原始	4151	36.4843750	-
Level5	3055	26.7812500	26.40327632
Level4	1758	22.5625000	57.64875934
Level3	1048	22.9375000	74.75307155

从表 5.2 中可知，采用同义词原始级别进行特征选择，特征空间维数是 4151，耗费时间大约 36.5 秒；采用 Level5 的同义词合并进行特征选择，特征空间维数降到 3055，特征降维度达到 26.4%，耗费时间约减少了 9.7 秒；采用 Level4 的同义词合并，相对于原始级别，特征降维度达到 57.6%，耗费时间将近减少 14 秒；采用 Level3 的同义词合并，相对于原始级别，特征降维度达到 74.7%，但耗费时间相对 Level4 反而增加了近 0.4 秒，主要原因是 Level3 相对于 Level4 的同义词合并替换操作的增加时间大于降维后在 TF*IDF 权重过滤节约的时间。总之，同义词合并技术可以实现特征空间的降维，越高级别降维度越大；Level4 及低级别同义词合并，级别越高特征选择的耗费时间越少，但 Level3 同义词合并操作相对 Level4 特征选择时间反而增加。

（2）基于改进 SOM 前后中文文本聚类系统的性能对比评估

本文重点是针对经典 SOM 存在聚类数目须预先输入、网络结构固定、初始化效果不理想以及聚类结果依赖样本输入顺序，导致 SOM 收敛速度缓慢，聚类质量差的问题，提出一种改进的 SOM 算法。因此基于改进 SOM 前后的中文文

本聚类系统的性能对比评估是是本章的核心，也是本文的目标有效性验证的关键步骤。由于聚类分析受中文预处理步骤的影响，所以本文同义词原始级别（即不采用同义词合并）、Level5（最低级别合并）、Level4、Level3（最高级别合并）特征选择条件下，分别进行 SOM 和改进 SOM 聚类实验分析。表 5.3、表 5.4、表 5.5 和表 5.6 分别是该四种特征选择条件下，进行聚类结果评估对比情况。表 5.7 反映的是四种同义词级别下，SOM 与改进 SOM 聚类耗费的时间对比情况。

表 5.3　原始级别改进 SOM 前后聚类结果对比表

文本类别		SOM			本文算法		
		查准率%	查全率%	F-测量%	查准率%	查全率%	F-测量%
交通		86.36364	76	80.85106	100	100	100
体育		95.91837	94	94.94949	98.03922	100	99.0099
艺术		96.15385	100	98.03922	100	100	100
教育		98	98	98	98	98	98
经济		87.5	98	92.45283	100	98	98.9899
政治		80.43478	74	77.08333	94.23077	98	96.07843
军事		86.79245	92	89.32039	97.91667	94	95.91837
总体评价	宏平均	90.16616	90.28571	90.2259	98.31238	98.28571	98.29904
	微平均	90.28571	90.28571	90.28571	98.28571	98.28571	98.28571

表 5.4 Level5 级别改进 SOM 前后聚类结果对比表

文本类别		SOM			本文算法		
		查准率%	查全率%	F-测量%	查准率%	查全率%	F-测量%
交通		100	96	97.95918	100	100	100
体育		94.33962	100	97.08738	98.03922	100	99.0099
艺术		100	100	100	100	100	100
教育		98	98	98	98	98	98
经济		96.07843	98	97.0297	98	98	98
政治		97.91667	94	95.91837	97.95918	96	96.9697
军事		98	98	98	98	98	98
总体评价	宏平均	97.7621	97.71429	97.73819	98.5712	98.57143	98.57131
	微平均	97.71429	97.71429	97.71429	98.57143	98.57143	98.57143

表 5.5　Level4 级别改进 SOM 前后聚类结果对比表

文本类别		SOM			本文算法		
		查准率%	查全率%	F-测量%	查准率%	查全率%	F-测量%
交通		100	100	100	100	100	100
体育		98	98	98	98	98	98
艺术		100	98	98.9899	100	98	98.9899
教育		94.23077	98	96.07843	94.23077	98	96.07843
经济		96.07843	98	97.0297	96.07843	98	97.0297
政治		95.74468	90	92.78351	95.74468	90	92.78351
军事		96.07843	98	97.0297	96.07843	98	97.0297
总体评价	宏平均	97.16176	97.14286	97.15231	97.16176	97.14286	97.15231
	微平均	97.14286	97.14286	97.14286	97.14286	97.14286	97.14286

表 5.6 Level3 级别改进 SOM 前后聚类结果对比表

文本类别		SOM			本文算法		
		查准率%	查全率%	F-测量%	查准率%	查全率%	F-测量%
交通		97.4359	76	85.39326	83.92857	94	88.67925
体育		79.24528	84	81.5534	86	86	86
艺术		83.92857	94	88.67925	90	90	90
教育		74.19355	92	82.14286	82.14286	92	86.79245
经济		84.21053	96	89.71963	94.73684	72	81.81818
政治		97.56098	80	87.91209	90.47619	76	82.6087
军事		88.09524	74	80.43478	79.31034	92	85.18519
总体评价	宏平均	86.38143	85.14286	85.75767	86.6564	86	86.32695
	微平均	85.14286	85.14286	85.14286	86	86	86

表 5.7 SOM 与本文算法聚类耗费时间对比表

同义词级别	聚类运行时间（s）	
	SOM	本文算法
原始	140.5	121.1
Level5	100.7	84.7
Level4	61.5	49.3
Level3	34.8	29.1

从表 5.3、表 5.4、表 5.5、表 5.6 和表 5.7 中可知，1）同义词合并可以实现特征空间降维，从而降低聚类分析的特征空间，提高聚类分析的速度；在 level5 和 level4 级别的同义词合并可以提高聚类的准确性，但当同义词级别为 level3 时，由于同义词过度合并，特征词过于减少导致文本之间的特征区分能力下降，文本聚类的质量反而下降。2）在四种级别同义词合并的条件下，改进 SOM 算法在聚类结果质量和收敛速度两方面都要优于经典 SOM 算法。

5.4 本章小结

本章介绍了本文的实验环境和语料来源；重点对实验结果进行对比分析，在此基础上实现对系统聚类结果进行评估。结果评估分为两部分：一、同义词合并技术对中文文本聚类系统的性能改善评估，评估结果表示同义词合并可以降低特征空间维数，低级别的同义词合并可以改善聚类的性能；二、基于改进 SOM 前后的中文文本聚类系统的性能对比评估，相对于经典 SOM，改进的 SOM 算法收敛速度更快，聚类质量更优，因此证明改进 SOM 算法有效。

第6章 结论与展望

6.1 结论

本文针对当前中文文本聚类领域的研究成果较少，聚类效果普遍不太理想，对文本聚类的理论和技术做了一些探讨，通过实验测试评估可以得到结论如下所示：

（1）本文将同义词合并技术引入文本聚类中。同义词合并可以实现特征空间降维，从而降低聚类分析的空间，提高聚类分析的速度；低级别的同义词合并可以提高聚类的准确性，但高级别同义词因过度合并，特征词过于减少导致文本之间的特征区分能力下降，文本聚类的质量反而下降。

（2）经典 SOM 存在聚类数目须预先输入、网络结构固定、初始化效果不理想以及聚类结果依赖样本的输入顺序等缺陷。本文针对此状况，提出改进的 SOM 算法。在网络形成阶段，采用聚类数目和网络结构动态增长、生长阀值约束聚类中心节点的生长策略，形成与实际样本分布一致的初始化网络，利于网络的快速收敛，聚类质量提高；在网络调整阶段，将模糊数学引入网络学习中，采用并行学习策略，实现所有训练样本对网络并行调整，使得调整结果与样本的输入顺序无关。

（3）本文在经典 SOM 研究及改进基础上，结合中文分词、停用词过滤、同义词合并等中文处理技术，设计实现了《中文文本聚类系统》平台，实践表明，改进的 SOM 算法可以改善中文文本聚类系统的性能。

6.2 展望

本文针对经典 SOM 算法存在的缺陷，提出了改进的 SOM 算法，将其应用于中文文本聚类中。虽然本文工作达到了预期的目标，但仍然存在许多需要进一步研究和改进之处，现将围绕这些不足之处展开进一步的工作：

（1）初始化网络依赖于生长阀值的设定，生长阀值稳定性和通用性没有理论和实践支撑，因此需要研究各种领域数据分布特点，研究出具有通用性或领域代表性和使网络生长稳定的生长阀值因子。

（2）最大聚类数目通过影响生长阀值来控制文本聚类的粒度，不能与实际

聚类数目相差太大。虽然它代替了预先输入聚类数目，但还是需要用户对实际的样本分布有所了解才能确定最大聚类数目，且其值的设定带有人为因素。所以如何将最大聚类数目控制聚类粒度或者采用扩展因子代替最大聚类数目是本文下一步的研究工作。

（3）学习率中模糊参数 m 的选取都是基于试验或者经验性的，没有确切的理论依据，因此研究模糊参数对学习率的影响也是下一步工作重点。

（4）停用词过滤和权重过滤完全是基于统计的，难以解决文本中的语义问题，难以实现深入而准确的特征降维，因此需要考虑结合统计和语义理解进行特征降维。

（5）同义词合并可以在一定程度上实现语义上的特征降维，但高级别同义词合并会因过度合并降低特征词区分文本的能力，导致聚类质量下降。目前还没一个如何控制同义词合并的度，使得聚类效果最佳，因此同义词的深入研究成为特征降维的一个重要方向。

致 谢

本文是在导师段隆振教授的悉心指导下完成的。导师严谨求实的治学态度、渊博厚实的学识和正直坦然的为人都是论文完成的重要保证，将使作者终身难忘。同时，导师给我提供的宽松的研究氛围，使作者能够在广阔的科研天地中充分发挥想象力和创造力。至此论文完成之际，谨向导师表示衷心的感谢和崇高的敬意！

在攻读硕士学位期间，得到了同门师兄弟在生活和学习上的帮助，在此表示对他们的由衷感谢！

另外，本课题还得到了一些朋友的无私帮助，他们提供的资料使得该课题能够顺利如期完成，在此深表感谢！同时还要感谢同窗苦读的其它37位同学，这段美好的学习生涯将在我的脑海中永存！

最后，我要感谢一直以来默默支持和鼓励我的父母，是你们的爱和无私奉献使我能够顺利的完成学业！衷心的感谢你们对儿子一直以来的关怀、支持和无私的爱！

刘飞荣

2010年10月

参考文献

[1] http://baike.soso.com/v88322.htm?ch=ch.bk.innerlink.

[2] 段明秀. 结合 SOFM 的改进 CLARA 聚类算法[J]. 计算机工程与应用 2010,46(22): 210~212.

[3] Salton G. Automatic Text Processing. [J] The Transformation Analysis and Retrieval of Information by Computer. Addison-Wesley, Reading, Massachusetts, 1989.

[4] Gahagan, Sean M. Adding value to value stream mapping: A simulation model template for VSM [J] IIE Annual Conference and Expo 2007: 712~717.

[5] Mhamdi, Faouzi. Textmining feature selection and datamining for proteins classification [J] From Theory to Applications, ICTTA 2004: 457~458.

[6] Konrad, Karsten. Model generation for natural language interpretation and analysis [J] Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science), 2004: 29~53.

[7] Ghandeharizadeh, S. Alternative approaches to distribute an E-commerce document management system [J] Proceedings of the International Workshop on Research Issues in Data Engineering - Distributed Object Management -RIDE-DOM, 2001:59-65.

[8] Yoshioka, Masaharu. Analyzing multiple news sites by contrasting articles [J] SITIS 2008 - Proceedings of the 4th International Conference on Signal Image Technology and Internet Based Systems, 2008: 45~51.

[9] http://baike.baidu.com/view/1133919.htm?func=retitle.

[10] 张启宇, 朱玲, 张雅萍. 中文分词算法研究综述[J]. 情报探索 2008,133(11):53~56.

[11] Gao Xin-bo and Xie Wei-xin, Advances in theory and applications of fuzzy clustering, Chinese Science Bulletin, 2000.11(45): 961-970.

[12] K.Alsabti, S.Ranka, and V.Singh. An Efficient K-Means Clustering Algorithm, http://www.cise.ufZ.edu/ranker/,1997.

[13] P. Willett, Recent Trends in Hierarchic Document Clustering: a critical review [J], Information Processing and Management, 1988.24 (5): 577~597.

[14] Wang Cui-ru and Duo Chun-hong, An Improved Density-based DBSCAN Clustering Algorithm, Journal of Guangxi Normal University Natural Science, 2007.25(4): 104-107.

[15] Li Cun-hua, Sun Zhi-hui, A Mean Approximation Approach to a Class of Grid-Based Clustering Algorithms, Journal of software, 2003.14(7):1267~1274.

[16] K.Y.Yeung, C.Fraley and A.Murua, Model-based Clustering and Data Transformations for Gene Expression Data, Bioinformation 2001.10(17): 50~52.

[17] H.Jin. Expanding Self-Organizing Map for data visualization and cluster analysis [J], Intormanon Sciences 2004 163:157~173.

[18] 杨占华, 杨燕. SOM 神经网络算法的研究与进展[J].计算机工程,2006,32(16):201~203

[19] 邵超, 黄厚宽. 一种新的基于 SOM 的数据可视化算法[J].计算机研究与发展 2006,43 (3):429~435.

[20] 王飞, 钱玉文, 王执铨. 基于混合 AIS/SOM 的入侵检测模型[J]. 计算机工程 2010, 36(12): 164~166.

[21] 刘远超, 王晓龙, 徐志明, 关毅. 文档聚类综述[J].中文信息学报,2006,20(3):55~62.

[22] Bezdek J C, Eric Chen-Kuo TsaoPal, NR. Fuzzy Kohonen Clustering Networks[C].IEEE International Conference on Fuzzy Systems,1992: 1035~1043.

[23] 曹均阔, 叶水生, 丁香乾. 模糊 kohonen 网络在烟叶分类中的应用[J]. 计算机工程 2005,31(2):222~224.

[24] Wang Xizhao, Wang Yadong, Wang Lijuan. Imp roving fuzzy c-means clustering based on feature2weight learning. Pattern Recognition Letters 25, 2004: 1123~1132.

[25] 孙晓霞, 刘晓霞, 谢倩茹. 模糊 C-均值(FCM)聚类算法的实现[J].计算机应用与软件. 2008, 25(3):48~50.

[26] 王莉, 王正欧. TGSOM: 一种用于数据聚类的动态自组织映射神经网络[J]. 电子与信息学报, 2003,25(3):313~319.

[27] 耿新青, 王正欧. DFKCN: 一种动态模糊自组织神经网络及其应用[J]. 计算机工程 2006,32(20):22~25.

[28] 胡可云, 田凤占, 黄厚宽. 数据挖掘理论和应用[M]北京 清华大学出版社 2008:139.

[29] 曾致远, 张莉. 基于向量空间模型的网页文本表示改进算法[J].计算机工程 2006,32(3): 134~136.

[30] 张爱华, 荆继武, 向继. 中文文本分类中的文本表示因素比较[J].2009, 26(3):400~407.

[31] 翟伟斌, 周振柳, 蒋卓明, 许榕生.汉语分词词典设计[J].计算机工程与应用 2007,43(1):1~3.

[32] 周宏宇, 张政. 中文分词技术综述[J].安阳师范学院学报 2010(2):55~57.

[33] 裘江南, 罗志成, 王延章. 基于中文语义词典的语义相关度方法比较研究[J].情报理论与实践 2008,31(5):715~719.

[34] 吴红艳. 基于自组织特征映射网络的聚类算法研究[D]. 重庆大学, 2009.

[35] 顾益军, 樊孝忠, 王建华等. 中文停用词表的自动选取[J].北京理工大学学报 2005, 25(4): 337~340.

[36] 化柏林. 知识抽取中的停用词处理技术[J].现代图书情报技术 2007,154(8):48~51.

[37] 熊文新, 宋柔. 信息检索用户查询语句的停用词过滤[J].计算机工程 2007,33(6): 195~197.

[38] 梅家驹, 竺一鸣, 高蕴琦等编. 《同义词词林》 [M].上海: 上海辞书出版社, 1983.

[39] 罗志成, 叶奋飞. 哈工大《同义词词林》共享版的若干改进[J] .

[40]《同义词词林》扩展版. http://www.ir-lab.org/.

[41] 吕震宇, 朱卫东等. 基于同义词词林的文本特征选择与加权研究[J].情报杂志 2008(5) .

[42] 王之鹏.web 文本分类系统中文本预处理技术的研究与实现[D]. 南京理工大学, 2009.

[43] Mao Yong, Zhou Xiao-bo, Pi Dao-ying, Sun You-xian, Wong Stephen T.C. Parameters selection in gene selection using Gaussian kernel support vector machines by genetic algorithm [J]. Journal of Zhejiang University Science B, 2005, 10(6).

[44] 毛勇, 周晓波等. 特征选择算法研究综述[J].模式识别与人工智能 2007,20(2):211~218

[45] Jiawei Han ,Micheline Kamber 著. 范明, 孟晓峰等译. 数据挖掘概念与技术(第 2 版)[M]. 机械工业出版社, 2007-3.

[46] Wang Yan, Applications A Modified K-means Clustering Algorithm, Computer and Software, 2004.21(10): 122~123.

[47] Liu Yuan-chao and Wang Xiao-long, An Adapted Algorithm of Choosing Initial Values for K-means Document Clustering, Chinese High Technology Letters, 2006.16(1):11~1 5.

[48] M Ester and H Kriegel, Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications, Data Mining and knowledge Discovery, 1998.2(2): 169~194.

[49] K.Y.Yeung, C.Fraley and A.Murua, Model-based Clustering and Data Transformations for Gene Expression Data, Bioinformation 2001.10(17): 50-52.

[50] Li Cun-hua, Sun Zhi-hui, A Mean Approximation Approach to a Class of Grid-Based Clustering Algorithms, Journal of software, 2003.14(7):1267~1274.

[51] 邱保志, 郑智杰. 基于局部密度和动态生成网格聚类算法[J]. 计算机工程与设计 2010,31 (2):385~388.

[52] 夏侯振宇. 基于粗糙集和支持向量机的文本分类方法研究[D].南昌大学, 2008.

[53] T. Kohonen. Self-Organizing Maps [M].Heidelberg: [s.n.], 1995(Second Extended Edition).

攻读学位期间的研究成果

已发表论文：

[1] 刘飞荣，段隆振等．一种基于动态模糊Kohonen网络的聚类模型及应用[J]. 南昌大学学报（理科版），2010,12(6).

[2] Duan Long-zhen, Liu Fei-Rong. A collaborative filtering algorithm based on the candidate set of recommended items [J]. 2010 2nd International Conference on Future Computer and Communication, 2010:287~291. (EI/ISTP检索)

参加的科研项目：

1、江西省社会劳动保障部门 “江西省小额贷款系统”软件研发项目

2、江西省社会劳动保障部门 “江西省养老保险系统”数据迁移项目