第3章 基于R-Grams的文本聚类方法

文本聚类是文本数据挖掘中的一个重要研究方向，网络热点发现是基于文本聚类的典型应用，且对聚类的要求较为特殊，即对准确率的要求较高，对召回率的要求次之。传统的文本聚类方法普遍同时追求准确率和召回率，计算资源耗费较多。针对传统文本聚类中存在着聚类准确率和召回率难以平衡等问题，本章提出一种基于 R-Grams 文本相似度计算方法的文本聚类方法，该方法首先将待聚类文档降序排列，其次采用 R-Grams 文本相似度算法计算文本之间的相似度，根据相似度实现各聚类标志文档的确定并完成初始聚类，最后通过对初始聚类结果并基于此对其进行聚类合并形成最终聚类。实验结果表明：基于 R-Grams 的文本聚类方法有效提高了聚类速度，聚类结果可以通过聚类阈值灵活调整以适应不同的需求，最佳聚类阈值为 15。随着聚类阈值的增大，各聚类准确率增大，召回率呈现先增后降的趋势。该聚类方法避免了大量的分词、特征提取等烦琐处理，实现简单，在文本处理领域具有良好的应用前景。

3.1 引言

在文本挖掘领域，文本聚类是一类常见而又重要的数据挖掘手段，同时也是很多其他挖掘操作的前置工作。随着互联网的高速发展，文本聚类在 Web 数据处理中应用尤其广泛，例如，搜索引擎、用户兴趣模式挖掘、网络舆情等。其中，网络舆情热点发现，网络舆情演化传播等研究都离不开对聚类的依赖。

文本聚类方法众多，暂无统一的划分方法，而且很多聚类方法并非完全

独立。传统的文本聚类方法，可以分为平面划分聚类（典型的划分聚类方法如 k-means[19]）、层次聚类（典型的如 BIRCH[20]）、基于密度的聚类[21-23]、基于模型的聚类（如基于神经网络[24]、蚁群算法[25]、遗传算法[26]等）、基于语义的聚类[27-29]、基于本体的聚类[30]、模糊聚类[31-32]）、谱聚类[33]、后缀树聚类[34]等。同时也有不少针对这些方法的改进或者多方法融合及分阶段应用的聚类方法[35-37]。在传统的聚类方法中，有诸多方法都需要词或特征项支撑，并常以这些为基础构建向量，进而实现聚类。如文献[38]中利用提取的特征实现聚类。对中文文档而言，词的获取需要借助中文分词技术来完成，而分词的准确率和速度则往往是一对矛盾，即使给予充分的时间也无法确保分词的准确性。分词完毕往往还需要统计词频、去停用词等操作，此外还需要计算词的权重，并以此来构建文档的向量表达。对于采用特征项的方法，则往往需要进行特征项的降维，文献[37]中基于前期降维方法的不足，提出一种**三阶段**（three-stages）降维方法。当采用适当的方式挑选合适的聚类中心后，通过计算各点与聚类中心的距离来实现聚类。上述过程中涉及过多烦琐且正确性难以保证的操作，这一方面降低了聚类的速度，另外也可能影响最终的聚类结果。

一般来说，准确率和召回率是聚类所追求的两个重要指标。但在很多实际的应用中，对聚类的准确性和速度要求较高，而对召回率的要求则并不严格。例如，互联网实时话题检测或者舆情热点发现是一个被广泛研究的方向，网络热点话题的检测往往正是建立在聚类基础之上。在该过程中，由于所处理文本文档数量庞大，计算速度尤其重要；并且由于最终目的往往只需要计算得到某话题相关文档集的频繁项集即可，这意味着，在庞大的数据集中，只需将某个话题相关的适量文档准确地识别并聚集起来即可，亦即该应用场景下的聚类对准确率的要求应尽量高，以减少频繁项集分析中所受的干扰；至于是否识别的足够全面则不再重要，即对该聚类的召回率并无太高要求。在该情况下，只需要将文档相似度计算或者距离计算过程中的阈值设置得尽量大，保证某些主要聚类的高准确率，同时保证其召回率不要过低即可，此时的聚类速度完全依赖于相似度或者距离计算方法。

在网络热点发现研究过程中，为了分析得到当前网络热点，面临着大量的网页数据聚类，即使这种聚类分析处理仅仅只是针对增量采集的数据而言，也面临着聚类时间长、聚类后各个聚类的准确率和召回率难以平衡等问题，并且在聚类之前，在分词、特征提取及计算等环节也需要消耗较多的计算资源。另外，由于热点发现具有其特殊性，故我们在关注聚类的高准确率

时无须太过关注其召回率。通俗地讲，虽然爬取的数据数量巨大，然而要识别发现其中的热点，只需要保证相关热点的聚类中数据足够纯即可，至于是否足够全则无关紧要。举例来说：假设 Web 页数据数目为 10000，并假设其中包含 3 个热点，各个热点的 Web 页数目分别为 3000、2000、1000。隶属于同一热点的 Web 数据虽多，然而要实现识别出该热点，并不需要将所有隶属于该热点的数据都聚到一个类中。例如，若将相似度阈值设置为一个较大的值，针对 3 个热点所获得的典型聚类大小分别为 300、200、100。虽然其召回率较低，然而由于阈值较大从而保证了聚类的准确率（纯度），仍然可以正确地分析获取当前热点。

文献[39]提出一种基于随机 n-Grams 的文本相似度计算方法，简称 R-Grams。基于 R-Grams 的文本相似度计算方法，可以省却烦琐的文本特征项提取过程，从而大幅提高计算速度，并且该方法还具有速度与精度易于调控、具备语言无关性的优点。基于前期关于文本相似度研究所提出的 R-Grams 文本相似度计算方法，提出本章的聚类方法，充分利用了 R-Grams 在计算相似度时无须执行分词等特征提取相关操作，而且阈值和聚类速度极容易调控。虽然在阈值较大时，本章聚类方法容易将隶属于一个热点的数据聚到多个聚类中，然而只需要挑选出其中不至过小的聚类即可进行热点分析。不过本章方法并不适宜应用到其他较为传统的聚类场合，利用该算法实现文档相似度计算，并最终完成文档聚类，且聚类结果在确保一定的召回率的同时达到了高准确率。为解决这一问题，故本章提出在聚类的第二阶段利用传统方法对第一阶段聚类结果进行聚类合并的解决方案。

本章创新点及方法优势如下：

（1）提出了基于 R-Grams 文本聚类方法，适用于高准确率、较低召回率的场合；

（2）该方法聚类过程中无须繁杂的特征提取操作；

（3）该方法基于阈值实现聚类，但对阈值的要求极宽松，无须在准确率和召回率之间权衡。

3.2　方法及原理

本章以 R-Grams 文本相似度计算方法为基础实现文本聚类，为后文表述方便，称其为 R-Grams 聚类，其主要过程如图 3-1 所示。

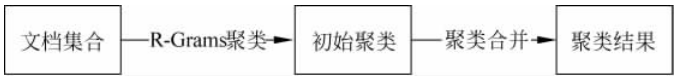

图 3-1　聚类的主要过程

3.2.1　R-Grams 聚类

R-Grams 文本相似度计算的核心算法公式可表述为：设有两个文档 D_i 与 D_j，则其相似度评价函数定义为

$$S(D_i,D_j)=\frac{\sum_{k=1}^{n}F(e_k)W(e_k)}{\sum_{k=1}^{n}W(e_k)} \tag{3-1}$$

其中 $F(e_k)=\begin{cases}1(e_k\in D_i\bigcap D_j)\\0(e_k\notin D_i\bigcap D_j)\end{cases}$，即若元素 e_k 不是同时存在于 D_i、D_j 中，则该元素对相似度无贡献。$W(e_k)$是元素 e_k 的权重评价函数。

聚类时，需要解决两个关键问题，其一是聚类中心（标志文档）的选择确立问题，其二是距离或者相似度计算问题。将 R-Grams 相似度算法应用于聚类时，第二个问题自然迎刃而解；对于第一个问题，本章采用先按文档长度降序排序然后通过逐个选择各类的聚类标志文档并完成该类文档的识别，从而完成聚类。降序排序的原因在于长文档往往比短文档含有更多的信息，故而优先用作聚类标志文档。

设原始子文档集为 k 个，每个子文档集对应一个主题，记为 $D_1=\{d_1^1,d_2^1,\cdots,d_{n_1}^1\}$，$D_2=\{d_1^2,d_2^2,\cdots,d_{n_2}^2\}$，$\cdots$，$D_k=\{d_1^k,d_2^k,\cdots,d_{n_k}^k\}$。实验文档集为上述文档并集，即 $D'=\{d_1^1,d_2^1,\cdots,d_{n_1}^1,d_1^2,d_2^2,\cdots,d_{n_2}^2,\cdots,d_1^k,d_2^k,\cdots,d_{n_k}^k\}$，在不必区分或者无法区分文档的归属时，可将文档集记为：$D'=\{d'_1,d'_2,\cdots,d'_n\}$，其中 $n=\sum_{i=1}^{k}n_i$，为文档集中文档数。由于聚类前需要对该文档排序，故将排序后的文档记为：$D=\{d_1,d_2,\cdots,d_n\}$，下文的各项表述或操作均是对文档集 D 进行。聚类过程中的相似度阈值为 T，即若文档相似度值不低于该值，则将这些文档归属到一个类中。聚类中文档数阈值为 C，即若某个初始聚类中的文档数不低于该值，则认定该初始聚类为一个有效聚类，否则舍弃。

聚类的核心过程如下：

输入：文档集合 $D=\{d_1,d_2,\cdots,d_n\}$

对文档集合 D 聚类，其核心伪代码如图 3-2 所示。

```
flag=1                          //既用于记录聚类数，同时也用作各聚类的
                                //序号
for(i=1; i<n; i++)
{
if di 未标记类别                 //di 未被标记所属类别
    for(j=i+1;j<n+1;j++)
    {
      if dj 未标记类别           //确保每个文档不会被归属到多个聚类中
      S=Sim(di,dj)              //利用 R-Grams 计算 di、dj 的相似度
      if S≥T
        if di 未标记类别
          标记 di、dj 类别为 flag
          flag++
        else
          标记 dj 类别为 flag
    }
}
```

图 3-2　R-Grams 文本聚类核心伪代码

聚类完毕，根据 flag 值即可知所获得的聚类个数，且每个聚类中最少元素个数为 2。可根据实际需要，设定合适的文档数阈值 C，过滤那些元素数过少的聚类。

经由上述 R-Grams 聚类后，所得聚类结果可以直接用于类似网络热点识别之类的应用场景。只需将所获得聚类中的较大聚类作分词等处理即可获知网络热点，并不需要将隶属于某个类的文档都准确地聚类出来才可分析出相关热点。由于这属于具体的应用范畴，并非本章重点，此处不再赘述。

倘若需将 R-Grams 聚类应用到其他更为广泛聚类场合，则需要对上述初始聚类结果作合并处理。

3.2.2　聚类合并

聚类合并可通过对上述聚类结果的二次聚类完成。二次聚类的对象并不再是整个文档集 D，只需要取各个类的标志文档即可。若两个类的标志文档被认定为一个类，则合并这两个标志文档所在的类。二次聚类采用常

规的聚类方法即可完成，本章采用先分词，然后利用文献[40，41]中频繁项集的方法进行二次聚类实现聚类合并。

3.2.3 聚类覆盖率

设与原始文档集对应的各个聚类中元素数为 $n_1^p, n_2^p, \cdots, n_k^p$，正确的元素数分别为 $n_1^q, n_2^q, \cdots, n_k^q$，则聚类的整体覆盖率定义为 $C_a = \frac{\sum_{i=1}^{k} n_i^p}{n}$，即所有聚类中文档数之和与总文档数的比值；正确覆盖率定义为 $C_r = \frac{\sum_{i=1}^{k} n_i^q}{n}$，即所有聚类中正确的文档数之和与总文档数的比值。

3.3 实验设计及结果分析

3.3.1 实验方案与目的

由于目前已有的几个大型互联网 Web 相关语料库中某个类的数据极为“稀疏”，且其中包含的类数量极其庞大，同时也少有公认的人工标记聚类可供使用，在无法获知数据集详细的人工标记的情况下，对聚类结果的解读和评价也是难以进行的。并且，由于本节聚类算法的应用场景为高准确率和较低召回率，即要求较高的阈值，故语料库的大小并不会影响聚类结果，只会影响聚类的速度。基于上述原因，本章并未使用公开的 Web 语料库，而是采用爬虫程序从互联网抓取数据并提取其中的正文文本，然后从中选取如下五个主题的文档共计 875 篇，其中与“企业跑路”相关的为 180 篇（记为“企业”），与“超女 XX”相关的为 165 篇（记为“超女”），与“转基因”相关的为 165 篇（记为“转基因”），与“以房养老”相关的为 165 篇（记为“养老”），与“染色馒头”相关的为 200 篇（记为“馒头”）。

实验中，R-Grams 相似度计算相关参数：所取元素长度为 2、3、4、5、6，每种长度均取 20 个。聚类设置相关参数：文档数阈值 $C=3$，文档相似度阈值 T 取 3，5，7，9，11，13，15，17，19，21，23。

实验方案与目的主要是对上述所有文档，执行如 3.2.1 节所述的聚类，记录相关聚类数据，然后对这些聚类执行合并，并计算各个聚类的准确率、

召回率等，探求聚类阈值对 R-Grams 聚类结果的影响。

3.3.2　实验结果与分析

1. 聚类阈值对聚类的影响

在利用上述方法进行聚类的过程中，阈值的设置对聚类结果起着决定性作用。阈值越小，各文档的类别归属出错的可能性越大；反之，阈值越大，各文档类别归属出错的可能性越小，但同时，也可能导致本属于同类的文档被归属到不同的类别中。聚类准确率结果如图 3-3 所示。

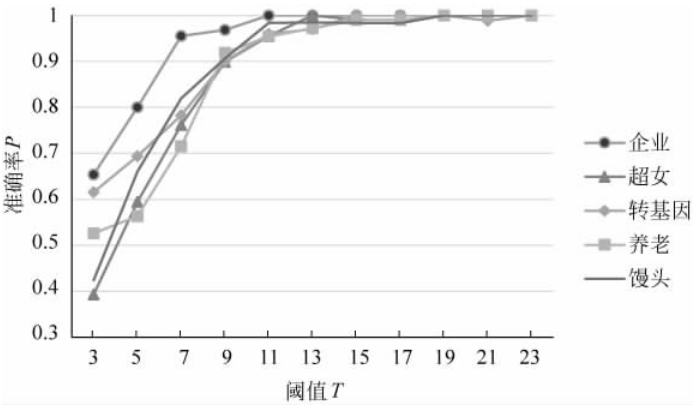

图 3-3　聚类阈值与准确率的关系

由图 3-3 可见：①随着阈值的增加，聚类准确率迅速提高，且各聚类准确率差值逐渐减小，直至最终基本相等。其主要原因在于：在聚类过程中，阈值充当着对各文档类别归属的把关作用，阈值越大，意味着聚类标准越严格，各文档被划分至错误类别的可能性减小，从而导致准确率的逐渐提高，直至最终均提高到接近 1；同时，各聚类准确率的最大差值也由 0.25（即 $T=3$）减小至 0（即 $T=19$）。②在阈值较小时，“企业”的准确率显著高于其他文档集的准确率；而“超女”“馒头”的准确率则稍低于其他文档集。经过深入细致的分析得知，造成该现象的主要原因在于：聚类过程中，各聚类标志文档的筛选是基于排序后的结果进行的，而标志文档的出现时机则在一定程度上会影响该聚类的准确率。在文档排序队列中，所处位置越靠前，则相应文档也将更容易成为聚类标志文档，同时，若某文档成为某聚类的标志文档，则处于列表前端的标志文档也将获得更多的与其他候选文档进行相

似度匹配计算的机会，而处于后端的标志文档则仅能获取较少的与其他文档进行相似度匹配的机会。这也就意味着将会出现两种现象：其一，先被确立为聚类标志的文档，将会纳入更多的文档至其所在的聚类，这一方面使得本该隶属于该聚类的文档不会被归属到其他聚类中，同时也可能使得很多本不该归属到该聚类的文档被错误的归属到该聚类中。其二，由于已有大量文档被归属到其应有的聚类，故后被确立为聚类标志的文档更多的只是将本应隶属于本类的文档纳入到本类中，而错误地将其他类别文档归属到本类的可能性大为减少，故会拥有较高的准确率。“企业”文档集中文档在排序队列中并未处于极其靠前的位置，这正是“企业”文档集聚类准确率普遍高于其他聚类的主要原因。而“超女”“馒头”文档集中的聚类标志文档则处于靠前位置，从而在一定程度上影响了其准确率，但这并非唯一的决定性原因，还与文档分布情况及算法本身因素等相关。聚类与长度的相关性，也是该算法在后续研究中应予以优化或解决的环节。③从图 3-3 可见，从聚类准确率角度来看，在利用 R-Grams 相似度算法实现聚类时，相似度阈值范围可初步确定在区间[11，19]，其中以[15，17]为优。

2．聚类阈值与召回率的关系

在评价聚类时，召回率也是一个重要指标，本实验聚类召回率如图 3-4 所示。

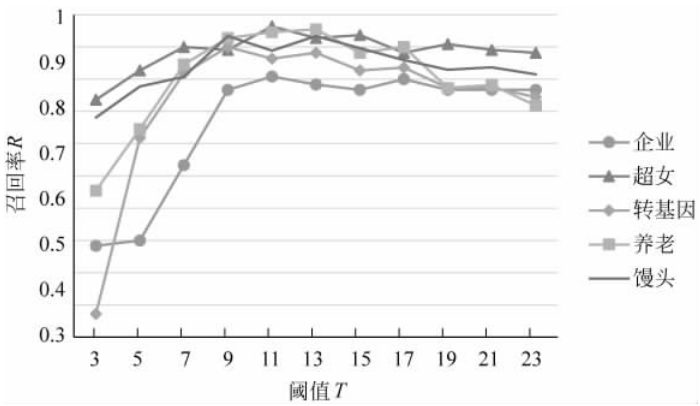

图 3-4　聚类阈值与召回率的关系

由图 3-4 可见：①随着阈值的增加，召回率普遍呈现为先增后降的趋势。其主要原因在于：在起始阶段（$T \leqslant 9$），由于阈值过小，聚类时文档归属错误的可能性极大，即很多本该隶属于某类的文档却被归属到其他类中，而

很多不应归属于某类的文档却被归属到本类中，这就导致了较低的召回率；随着阈值的增大，文档归属错误情况逐渐缓解，于是召回率逐渐提升。因此可以把低阈值时的阈值作用归结为“类间纠错”。然而在后期阶段（$T>11$），由于阈值已经足够正确区分绝大多数文档的正确归属，因此其作用不再是“类间纠错”，而是“类内细分”，即由于过高的阈值，将在较低阈值时归属于一个较大类中的文档分割为多个较小的聚类，虽然这些较小的类中相当一部分最终仍然被合并起来，不过仍将有一部分被分割为独立文档或者极小的聚类，这些极小的聚类由于文档数太少而被舍弃，从而导致了后期阶段召回率的下降。另外，由于本实验中将有效聚类元素个数确定为 3，因此意味着只会舍弃那些元素小于 3 的聚类，被舍弃的文档量极其有限，这就是后期召回率下降缓慢的原因；倘若将有效聚类的元素数提高，则召回率的下降趋势将逐渐更为明显。②在阈值较小时，“超女”等的召回率处于高位，而“企业”的召回率则明显低于其他文档集。召回率在整体上与准确率呈现为相反的顺序，即在文档排序队列中，所处位置越靠前，其准确率往往更低，而召回率则往往更高，但并非与此严格吻合。其原因前文已述及，即所处位置靠前，将会获得更多的相应属于本类的文档纳入到本类中，从而呈现较高的召回率。③从图 3-4 易知，从聚类召回率角度来看，在利用 R-Grams 相似度算法实现聚类时，相似度阈值范围可初步确定在区间[9,17]，其中以[13,15]为优。

3. 聚类阈值与 F-score 的关系

在评价聚类时，由于准确率和召回率都只是从某一个方面来评价聚类，在实际中往往会采取综合指标，即 F-score。相关结果如图 3-5 所示。

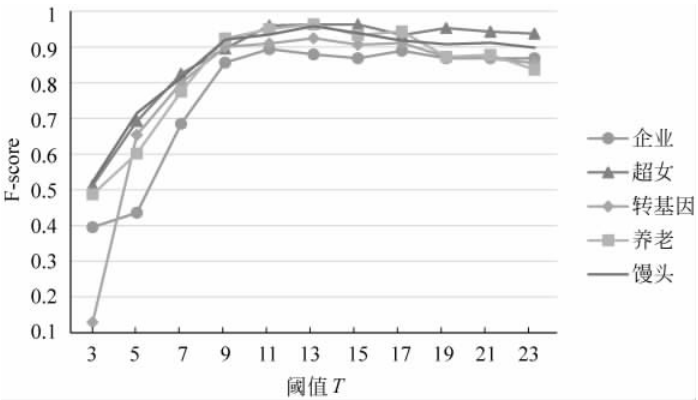

图 3-5　聚类阈值与 F-score 的关系

由图 3-5 可见，在整体上，F-score 曲线的升降趋势与召回率一致，即先升后降。此外，从图 3-5 也可看出，阈值的优选区间为[11，15]。

4．聚类阈值对初始聚类数的影响

初始聚类数是利用 R-Grams 进行聚类后直接聚类的结果，实验结果如图 3-6 所示。

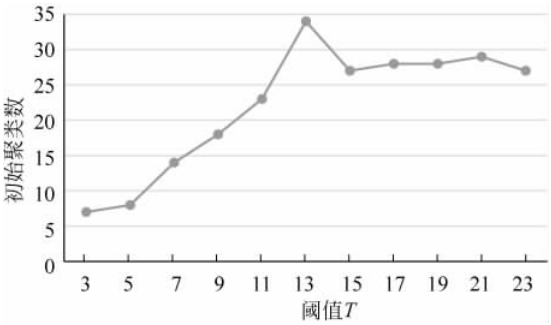

图 3-6　初始聚类数与阈值的关系

从图 3-6 可见，当阈值稍大时，利用该方法所获得的初始聚类数较多，因而各聚类召回率必然偏低，但是各聚类准确率都很高。这恰好符合前文所述的类似网络热点识别之类的应用场景。虽然一些较好的聚类算法能取得较好的召回率和准确率，但在聚类过程中需消耗过多的计算资源。本章方法在实现网络热点发现的前提下，同时可大幅降低资源消耗。以实验数据为例，若采用基于分词的聚类方法，则需要对 875 个文档作分词或特征提取及后续诸多烦琐操作，采用本方法后，则完全无须任何分词及后续的操作。即使在其他一般性文本聚类场合，本章所述方法也仅在聚类合并阶段才需要执行分词之类的操作，由上述结果可见，在此阶段需要执行这一系列烦琐操作的文档数大约为 30 个，仅占总文档数 3%左右，最终聚类效果与常规的文本聚类方法相当。

5．聚类阈值对聚类文档覆盖率的影响

整体覆盖率 C_a 和正确覆盖率 C_r 实验结果如图 3-7 所示。

由图 3-7 可见，整体文档覆盖率随着聚类阈值的增加呈现单调递减趋势，正确文档覆盖率则呈现先升后降的趋势。显然，随着聚类阈值的增大，

文档将更难以聚到一起，或者难以聚成较大的类。由于各个聚类对纳入该类的文档的限制更为严格，这将导致越来越多的文档成为独立于任何聚类的个体文档，或者由于所含文档过少而无法被认定为有效聚类，在宏观上即呈现为整体文档覆盖率的持续下降。对正确元素覆盖率而言，则与上述情形有所不同。在阈值较小时，虽然绝大多数的文档都被归属到相关聚类中，但是正如前文所述，低阈值时的归属错误率极高，这一问题随着阈值的增大将逐渐缓解（即低阈值时阈值呈现为“类间纠错”功能），这正是正确文档覆盖率在开始阶段呈现增长趋势的原因。在阈值较大时，由于阈值的“类内细分”作用，诸多的大类被分割为多个细小的聚类甚至一些独立的文档，在该过程中，越来越多的独立文档和极其细小的聚类被排除到有效聚类之外，宏观上即呈现为正确文档覆盖率的缓慢下降。这在另外一个侧面再次印证了前文所论述的阈值的两种典型作用。当阈值增大到一定程度时，阈值已具备充分的辨识能力，可确保被归属到同一个类中的文档在实际上也的确是同类文档，此即当阈值较大时，两条曲线基本重合的原因。

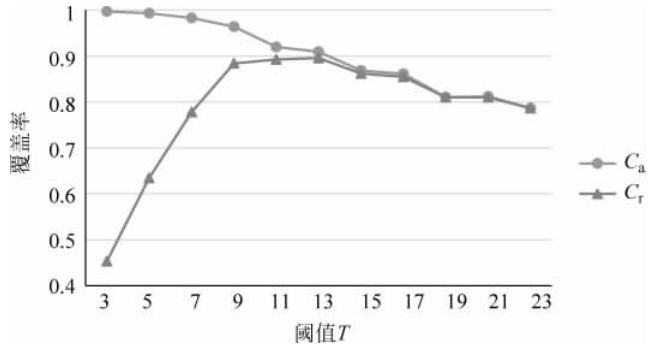

图 3-7　聚类阈值与覆盖率的关系

6. R-Grams 聚类方法特性及不足之处

R-Grams 聚类方法的特性可总结为：聚类多，准确率高，召回率低，聚类精度和速度易于调控。该方法可通过调整相似度计算中 n-Grams 的数目及各项阈值来实现聚类精度和速度的调控，故决定了其可用于实时性较高的场合，也可用于精度要求较高的场合，但并不能用于召回率较高的场合。另外由于该方法可以获取多个准确率高的聚类，通过其中的较大的聚类即可完成类似网络热点发现之类的应用需求。这主要是由于在实际情况下，网

络热点一旦产生，虽然围绕着一个热点话题的数据往往涉及多个方面，但其中往往存在着大量由于转载或其他原因而导致有较大重复率的文档。只要能把这些重复率较高的文档识别出来，就足以分析出相关热点，而并不需要识别出该热点所有相关数据，这正是本章聚类方法具有实用价值的客观支撑条件。从本实验的初步聚类结果来看（即在不进行聚类合并条件下的聚类结果），虽然聚类数较多，但其中较大的聚类却并不多，在实际进行网络热点分析时，只需利用其中的几个较大聚类即可实现。另外，由于实现海量网络数据中热点的识别只需要能够取得其中一个较大的且准确率高的聚类即可，至于该类中元素是多一些还是少一些，都不会影响热点分析结果，这就决定了虽然本章方法仍然是基于阈值进行聚类的，但是对阈值要求却很低，只需要阈值较大，例如在 15 以上，但不要高于 19 即可。以 $T=15$ 时的各聚类数分布如图 3-8 所示，直方图中分组间距为 15，第 1 组为聚类中文档数为 3 的聚类数，第 2 组为文档数为 4～18 的聚类数，以此类推。

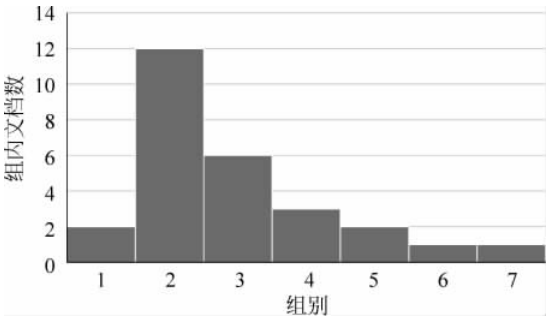

图 3-8　阈值 $T=15$ 时的各聚类数分布直方图

另外，虽然采用本章方法时，取较小的阈值能够获得较少的聚类，不过由于此时各聚类中包含了一定数量的归属错误的文档，这些对热点分析不利，故低阈值并不适合进行热点分析。R-Grams 不足之处在于：①需要在聚类前进行排序，即存在一定的文档长度相关性。②在低阈值时，排序结果将对聚类结果产生一定影响。不过高阈值时不再存在该问题。③在低阈值且计算量极少时，R-Grams 文本相似度计算方法存在一定的随机性。不过该问题在高阈值时基本不存在，或者也可以通过提高循环计算次数以降低其在低阈值情况下的随机性影响。

3.4　结论

本章提出的基于随机 n-Grams 的文本聚类方法，避免了很多传统聚类算法在文本聚类过程中不可避免的一些预处理，例如，分词、特征提取与筛选等，聚类计算速度得到了极大提高；另外本方法也可轻松地通过调整阈值实现对聚类速度、聚类精细程度等的调控，以适应不同的应用场景。本方法可直接应用于网络话题检测或者网络热点识别等场合，在结合聚类合并操作或其他聚类方法的基础上，可广泛地应用于其他文本聚类场合，具有良好的应用价值。本章所提方法聚类结果的文档长度相关性则有待后续进一步研究，对低阈值时存在的问题的解决方案细节则有待进一步优化。