基于语义特征抽取的文本聚类研究

殷 硕,王卫亚,柳有权
(长安大学 信息工程学院,陕西 西安 710064)

摘 要:基于向量空间模型(VSM)的文本聚类会出现向量维度过高以及缺乏语义信息的问题,导致聚类效果出现偏差。为解决以上问题,引入《知网》作为语义词典,并改进词语相似度算法的不足。利用改进的词语语义相似度算法对文本特征进行语义压缩,使所有特征词都是主题相关的,利用调整后的 TF-IDF 算法对特征项进行加权,完成文本特征抽取,降低文本表示模型的维度。在聚类中,将同一类的文本划分为同一个簇,利用簇中所有文本的特征词完成簇的语义特征抽取,簇的表示模型和文本的表示模型有着相同的形式。通过计算簇之间的语义相似度,将相似度大于阈值的簇合并,更新簇的特征,直到算法结束。通过实验验证,与基于 K-Means 和 VSM 的聚类算法相比,文中算法大幅降低了向量维度,聚类效果也有明显提升。

关键词:文本聚类;语义特征抽取;特征降维;文本相似度;知网
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2020)03-0046-05
doi:10.3969/j.issn.1673-629X.2020.03.009

Research on Text Clustering Based on Semantic Feature Extraction

YIN Shuo,WANG Wei-ya,LIU You-quan
(School of Information Engineering,Chang'an University,Xi'an 710064,China)

Abstract:Text clustering based on vector space model (VSM) has the problems of too high vector dimension and lack of semantic information,which results in the deviation of clustering effect. In order to solve the above problems,we introduce HowNet as semantic dictionary and improve the word similarity algorithm. The improved word semantic similarity algorithm is used to compress the text features semantically so that all feature words are subject-related. The adjusted TF-IDF algorithm is used to weigh the feature items to complete the text feature extraction and reduce the dimension of the text representation model. In clustering,the text of the same class is divided into the same cluster,and the semantic features of the cluster are extracted by using the feature words of all the text in the cluster. The representation model of the cluster has the same form as the representation model of the text. By calculating the semantic similarity between the clusters,the clusters with similarity greater than the threshold are merged and the features of clusters are updated until the end of the algorithm. Experiment shows that compared with K-Means and VSM-based clustering algorithm,the proposed algorithm greatly reduces the vector dimension and improves the clustering effect significantly.

Key words:text clustering;semantic feature extraction;feature dimension reduction;text similarity;HowNet

0 引 言

文本聚类是将大规模文本按照某种表示模型划分为多个簇,使得同一个簇中的文本之间相似度尽可能大,不同簇中的文本之间相似度尽可能小[1]。文本聚类中最重要的两个步骤是:特征选取和利用特征进行相似度判断[2]。常见的文本聚类有基于向量空间模型的文本聚类和基于潜在语义索引的文本聚类[3]等。其中以向量空间模型[4](vector space model,VSM)作为文本表示模型,并使用 TF-IDF(term frequency-inverse document frequency)作为模型中元素的权重的文本聚类方法应用最为广泛,比如文献[5]提出了一种基于 K-Means 和 VSM 的聚类算法,利用 VSM 模型计算文本相似度,从而实现文本聚类算法。但是使用 VSM 作为文本表示模型会产生两个问题:一是表示文本的向量维度过高,导致算法复杂度过高;二是 VSM 模型缺乏词语的语义信息。VSM 向量维度过高的问题通

收稿日期:2019-05-15 **修回日期**:2019-09-17 **网络出版时间**:2019-12-05
基金项目:中央高校基本科研业务费专项资金(310824173401)
作者简介:殷 硕(1995-),男,硕士研究生,研究方向为计算机网络及移动互联技术、数据挖掘;王卫亚,博士,教授,研究方向为计算机网络及应用;柳有权,博士,教授,研究方向为计算机图形学。
网络出版地址:http://kns.cnki.net/kcms/detail/61.1450.TP.20191205.1146.078.html

常采用降维策略,对文本进行特征抽取[6-8]或者挖掘频繁项集作为特征信息[9-10]的方法降低数据的维度。

文中将《知网》[11]作为语义词典引入到文本聚类中,提出一种既能降低向量维度,又能弥补 VSM 所缺少的语义信息的聚类方法。该方法首先改进词语语义相似度算法,其次在词语语义相似度的基础上对文本进行语义特征抽取,降低文本表示模型的维度,以及完成对簇的语义特征抽取,最后通过计算抽取的特征集合之间的相似度,完成文本聚类。

1　词语语义相似度算法改进

1.1　义原相似度算法

《知网》将义原分为了几个大类,类与类之间不存在交集。通过义原之间的上下位关系,为每一个类构建出一棵义原层次树,不同义原层次树之间不存在可达路径。在知网中义原层次树部分示意图见图 1。

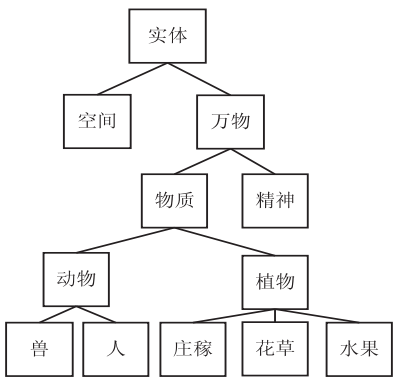

图 1　义原层次树示意图

朱新华[12]提出了综合义原层次树的深度以及密度因素计算义原相似度的公式,在一定程度上提高了词语语义相似度的准确性,具体公式为:

$$\mathrm{sim}(p_1,p_2)=\frac{1}{2}\times\frac{\alpha}{\alpha+\sum_{i=1}^{N}\mathrm{weight}(\mathrm{level}(i))}+\frac{1}{2}\times\frac{2\times\log f(\mathrm{LCN})}{\log f(p_1)+\log(p_2)}\tag{1}$$

其中,p_1 和 p_2 为两个义原,α 为可调节参数,N 为可达路径长度,$\mathrm{level}(i)$ 为可达路径上的边在义原层次树中的层次,LCN 为两个义原在层次树中的最小公共父节点,$f(\cdot)$ 为当前节点的密度信息,其值为所有的兄弟节点的个数(含自身)除以义原层次树的总节点个数,$\mathrm{weight}(\cdot)$ 函数为每一条边的权重,定义为:

$$\mathrm{weight}(i)=\frac{\mathrm{depth}-1-i}{\mathrm{depth}-1}\times\left(1+\sin\left(\theta*i*\frac{\pi}{180}\right)\right)\tag{2}$$

其中,depth 为义原层次树的高度,θ 为调节参数,与树高 depth 成反比,经过实验验证取 $\theta=4$,i 为当前所在的层次。

1.2　义项相似度计算

义项是使用知识表示语言进行描述的,通过对《知网》知识描述语言进行分析,刘群[13]按照描述形式的不同将描述义项的义原分为 4 个集合:

义项表达式
- 第一基本义原
- 其他基本义原
- 关系义原
- 关系符号义原

通过计算相同类型集合的相似度,再对其进行加权求和得到两个义项之间的相似度。具体公式为:

$$\mathrm{sim}(S_1,S_2)=\sum_{i=1}^{4}\beta_i\prod_{j=1}^{i}\mathrm{sim}_j(S_1,S_2)\tag{3}$$

其中,S_1 和 S_2 为两个义项,$\mathrm{sim}_j(S_1,S_2)$ 为第 j 类集合的相似度,β_i 为对集合相似度的加权,且满足 $\beta_1+\beta_2+\beta_3+\beta_4=1$,$\beta_1>\beta_2>\beta_3>\beta_4$。

1.3　词语相似度计算

假设现有两个词语 W_1 和 W_2,词语 W_1 有 n 个义项,即 $s_{11},s_{12},\cdots,s_{1n}$;词语 W_2 有 m 个义项,即 $s_{21},s_{22},\cdots,s_{2m}$,在计算词语之间的相似度时,首先需要进行消歧,具体消歧算法后面进行讨论。在经过消歧之后,得到两个词语唯一的义项 S_1 和 S_2,W_1 和 W_2 之间的相似度就是 S_1 和 S_2 之间的相似度。

2　基于语义特征抽取的文本聚类算法研究

2.1　词语语义相似度改进

虽然文献[12]在计算词语相似度时使用了义原层次树的密度信息,但是却没有考虑到可达路径上所有节点的密度对相似度的影响。所有子节点是对父节点所表达的概念的进一步细分,比如“植物”的子节点有“水果”、“花草”、“树”等,所以密度越大代表细分的程度越大。可达路径上的所有节点都比正在计算相似度的节点在树中的层次高,即在可达路径上的所有节点都是这两个节点中某一个的父节点,父节点的密度越大,在一定程度上也影响着子节点的分类细致程度。所以,文中将结合可达路径上的所以节点的密度,并对其进行加权再求和,得义原相似度计算时的密度部分:

$$\mathrm{sim}_d(p_1,p_2)=\frac{2\times\sum_{i=1}^{N}\varepsilon_i\log f(p_i)}{\log f(p_1)+\log f(p_2)}\tag{4}$$

$$\varepsilon_i=\frac{\mathrm{level}(i)}{\sum_{j=1}^{N}\mathrm{level}(j)}\tag{5}$$

其中,$\mathrm{sim}_d(p_1,p_2)$ 表示两个义原相似度计算时的密度部分,N 为可达路径长度,ε_i 为可达路径上的节

点 p_i 的密度的加权值，level(i) 为节点所在的层次，$\sum_{j=1}^{N}$ level(j) 为归一化参数，表示可达路径上所有节点的层次的和。

通过上述处理，得到新的义原相似度计算函数：

$$\mathrm{sim}(p_1,p_2)=c_1\times\frac{\alpha}{\alpha+\sum_{i=1}^{N}\mathrm{weight}(\mathrm{level}(i))}+c_2\times\frac{2\times\sum_{i=1}^{N}\varepsilon_i\times\log f(p_i)}{\log f(p_1)+\log f(p_2)} \tag{6}$$

其中，c_1 和 c_2 是平衡深度和密度对相似度影响的权重因子，经过实验，文中取 $c_1=0.7$，$c_2=0.3$。

2.2 文本预处理

2.2.1 文本内容分词

对于一篇文本，并不是所有的词语都是有实际意义的。中文包含许多停用词、虚词等，所以需要对文本进行分词、去停用词、去虚词等操作。文中使用 NLPIR-ICTCLAS[14] 分词系统进行分词，首先对 NLPIR-ICTCLAS 提供的二次开发接口进行编程对文本进行分词，再利用停用词表、虚词表对分词结果进行过滤，得到分词过后的词集。

2.2.2 基于语义相似度的词语消歧算法

中文包含多义词，多义词在《知网》中具有多个义项，所以需要对多义词进行消歧，确定词语唯一的义项。笔者认为，多义词在一个句子中的义项应该是唯一的，在多义词的所有义项中，需要确定的义项与其他已经确定了义项的词语之间的相似度是最大的。具体的消歧算法如下：

(1)获得多义词 W 的所有义项 ($s_1,s_2,\cdots,s_m$)，以及句子中已经确定了义项的词语集合 ($W_1,W_2,\cdots,W_n$)；

(2)令 W 的所有义项的初始权重都为 0；

(3)依次计算 W_i 的义项和 ($s_1,s_2,\cdots,s_m$) 之间的相似度，如果 W_i 和 s_j 之间的相似度最大，则对 s_j 的权重加 1，其中 $1\leqslant i\leqslant n$，$1\leqslant j\leqslant m$；

(4)比较 ($s_1,s_2,\cdots,s_m$) 的权重，选择权重最大的义项为 W 的唯一义项。

通过上述算法，确定多义词在一个句子中的唯一义项。但是在一篇正文中，多义词可能会出现在多个句子中，而且所有句子中的义项不一定相同。针对这种情况，文中采取如下做法：

(1)计算每个义项在正文中所出现的次数；

(2)选取出现次数最多的义项作为多义词在正文中的唯一义项。

2.3 文本语义特征抽取

如果直接使用 2.2 中得到的文本词集作为文本表示模型会出现两个问题：一是由于模型维度过高而导致算法复杂度过高，二是词集中含有大量与文本主题无关的词语，会降低聚类的精准度。所以需要对预处理后的文本词集进行语义特征抽取，在获得文本主题相关的特征项的同时，也可以降低模型维度。

2.3.1 语义特征压缩

文本的主题是通过一系列主题词进行描述的，而主题词之间则具有较大相似度，通过词语之间的语义相似度，可以获取到文本的主题词集合 d，具体算法为：

(1)对集合 $D=\bigcup_{i=1}^{n}W_i$，计算集合中所有词语的两两相似度，其中 W_i 表示集合中的词语；

(2)将步骤(1)中结果记为 $S=\bigcup_{i\neq j}S_{ij}$，将 S 按照相似度的降序排序；

(3)在 S 中，将相似度 $S_{ij}\geqslant\mu$ 的词 W_i 和 W_j 所在的集合合并，其中 μ 表示语义相似度阈值，相似度大于 μ 的两个词语归为同一集合；

(4)最后选取元素最多的一个集合作为文本主题词集合 d。

2.3.2 文本特征抽取

在获取到文本的主题词集合 d 之后，需要根据主题词的权重抽取出文本的特征集。由于进行了语义压缩，笔者认为语义因素比词语的频数因素更加重要，所以对 TF-IDF 进行调整之后提出如下公式计算主题词的权重：

$$f(W_i)=1-\frac{N_i}{N} \tag{7}$$

其中，N_i 为包含词 W_i 的文本个数，N 为文本总数。

在计算出所有主题词的权重之后，选取权重降序排序的前 15 个词作为文本的特征词集，主题词的权重仅仅作为特征选择的依据，并不参与文本相似度计算。通过特征词集建立文本表示模型 $D_i=\{W_{i1},W_{i2},\cdots,W_{in}\}$，其中 D_i 为文本集中的第 i 个文本，W_{ik} 为 D_i 的第 k 个特征项。由于特征词都是经过语义压缩以及主题词权重排序抽取得到的，所以文中所有特征词具有相同的语义权重。

2.4 文本语义相似度计算

假设有两个文本表示模型 $D_i=\{W_{i1},W_{i2},\cdots,W_{in}\}$ 和 $D_j=\{W_{j1},W_{j1},\cdots,W_{jm}\}$，且 $m\geqslant n$，语义相似度算法为：

(1)采用完备二部图的构造方法，将两个模型的特征集的元素作为二部图中的两个顶点集合，建立连接，D_i 和 D_j 所构成的二部图如图 2 所示。

(a)计算 D_i 部每个顶点和 D_j 部每个顶点的相似

度,把它作为两个顶点的边的权值,所有边的权值集合记为 S;

(b)从 S 中选取权值最大的边 $\{W_{ip},W_{jq}\}$ 加入集合 L,并从顶点集合中删除顶点 W_{ip} 和 W_{jq} 以及从 S 中删除所有与之相关的边;

(c)重复(b),直到 D_j 部中的顶点为空。

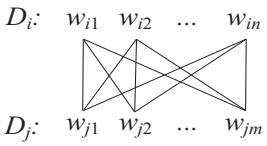

图 2　两个文本模型构成的二部图

(2)由集合 L 中的边的权值得出文本表示模型的相似度计算方法:

$$\mathrm{sim}(D_i,D_j)=\frac{1}{m+n}\Big(\sum_{L}\mathrm{sim}(W_{ip},W_{jq})+0.1*(m-n)\Big) \tag{8}$$

其中,$0.1*(m-n)$ 是当 $m>n$ 的情况出现时,W_i 中元素与空对应,赋予一较小常数。

2.5　簇的语义特征抽取

设簇 $C=\bigcup_{p=1}^{k}D_p$ 包含 K 个文本 $\{D_1,D_2,\cdots,D_k\}$,通过如下算法抽取簇的特征:

(1)将 C 中所有文本的特征抽取出来,组成向量 $\boldsymbol{D}'=\{(W_1,F_1),(W_2,F_2),\cdots,(W_n,F_n)\}$,其中 F_i 为所有文本中 W_i 出现的频数;

(2)类似于文本特征抽取算法,计算 $\boldsymbol{D}'$ 中所有词语的两两相似度,找到相似度大于阈值 μ 的最大集合 d';

(3)选取 d' 中频数降序排序的前 30 个词作为簇的特征集。

与 2.2 类似,这里的频数仅仅作为簇的特征抽取的依据,并不参与簇的相似度计算,簇中的特征项具有相同的语义权重。获取到簇的特征集之后,将簇的表示模型定义为 $C=\{W_1,W_2,\cdots,W_n\}$,与文本表示模型形式相同,所以簇之间的相似度计算类似于文本相似度计算,以下不再描述。

2.6　文本聚类算法设计

假设现有文本数量为 N,需要将这 N 篇文本进行聚类,使之被分在不同的集合中,不同的集合代表不同的簇。首先利用文中提出的文本语义特征抽取算法抽取每个文本的特征集,初始情况下,将这 N 个文本视为 N 个集合,即 N 个簇,每个簇的特征集为对应文本的特征集。计算所有簇两两之间的相似度 $\mathrm{sim}(C_i,C_j)$,如果相似度大于阈值,则将两个簇进行合并,并重新抽取新簇的特征。如果两次迭代之后簇的个数不变,则终止该算法。具体描述为:

(1)抽取每个文本的特征集;

(2)将 N 个文本初始化为 N 个簇,每个簇的特征集为对应的文本的特征集;

(3)计算簇之间的两两相似度,如果两个簇的相似度大于阈值 α,则将两个簇合并;

(4)根据簇的语义特征抽取算法更新所有簇的特征集;

(5)重复步骤(3)和步骤(4),直到两次迭代之后簇的个数不变。

3　实验与分析

3.1　实验数据获取

使用爬虫程序在新浪新闻网站中爬取财经、旅游、教育、文化、军事 5 个类别各 400 篇网页,共 2 000 篇作为实验数据。

3.2　聚类实验

为了检验所提出的聚类算法的优劣性,使用准确率(Precision,P)、召回率(Recall,R)和 F_1 度量值作为评价指标,具体公式如下:

$$P=\frac{a}{a+b} \tag{9}$$

$$R=\frac{a}{a+c} \tag{10}$$

$$F_1=\frac{2\times P\times R}{P+R} \tag{11}$$

其中,a、b、c 所表示的含义如表 1 所示。

表 1　评价指标参数

	真正属于该簇的文档数	真正不属于该簇的文档数
判断为属于该簇的文档数	a	b
判断为不属于簇的文档数	c	\

实验之前,首先需要确定文本特征抽取和簇特征抽取过程中所使用的阈值 μ,以及聚类算法中不同簇之间的相似度阈值 α。文中参考刘怀亮[15]所使用的词语相似度阈值,令 $\mu=0.8$。然后需要确定阈值 α 的最佳值,图 3 显示了不同阈值 α 下对聚类结果的影响。

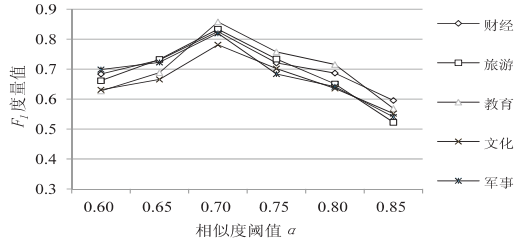

图 3　不同阈值对聚类的影响

当 $0.6\leqslant\alpha\leqslant0.7$ 时,F_1 度量值随着 α 的增大而增大,表明聚类效果越来越好。主要原因是当阈值 α 变大时,不同簇之间的区分度也越来越大,所以聚类效果也在逐步提升。当 $0.7\leqslant\alpha\leqslant0.85$ 时,F_1 度量值随

着 α 的减小而减小,表明聚类效果反而降低了。主要原因是当阈值 α 变得过大时,原本应当合并为一个新簇的两个簇的相似度却达不到阈值 α ,所以聚类效果逐步降低。

在设定簇相似度阈值 $\alpha = 0.7$ 之后,添加文献[5]基于 K-Means 和 VSM 的聚类算法作为对比,表 2 为两种算法中每个类别文本的所有特征维度比较。

表 2 特征集维度比较

类别	文献[5]算法	文中算法
财经	17 168	4 527
旅游	16 863	4 616
教育	18 082	4 594
文化	15 908	5 237
军事	16 097	4 679

由表 2 可以得出,文中提出的文本表示模型相较于传统的 VSM 文本表示模型在维度方面有着极大的优势,主要因为文中使用语义对特征词进行了抽取,每一个文本的特征词数量都不会超过 15,而 VSM 则将所有词语所组成的向量作为文本表示模型,使向量维度极大。

表 3 为两种算法的准确率、召回率和 F_1 度量值的对比。

表 3 实验结果对比

类别	文献[5]算法			文中算法		
	P	R	F_1 值	P	R	F_1 值
财经	0.743 1	0.730 0	0.706 2	0.834 6	0.816 4	0.825 4
旅游	0.724 3	0.710 2	0.717 2	0.852 6	0.814 6	0.833 1
教育	0.743 7	0.732 5	0.738 1	0.852 4	0.863 1	0.857 7
文化	0.711 2	0.708 4	0.709 7	0.794 8	0.768 3	0.781 3
军事	0.701 9	0.725 1	0.713 3	0.833 9	0.806 0	0.819 7

由表 3 可以得出,文中提出的算法相较于文献[5]的算法在准确率、召回率和 F_1 度量值上都有所提高,其原因主要有两点:一是加入了语义信息,弥补了 VSM 文本模型中语义缺失的问题,使词语相似度更符合人类主观判断的结果,二是通过语义对文本特征进行了抽取,使特征项都是主题相关的,减少了主题无关词语对文本相似度的影响,从而得到了更加准确的文本相似度。

4 结束语

文中提出一种基于语义特征抽取的文本聚类算法,使用词语的语义信息和词语权重对文本的特征项进行了抽取,不仅可以降低文本表示模型的维度,同时所抽取的特征都是主题相关的,彼此之间有着很大的关联。通过计算文本表示模型之间的相似度使同一类的文本聚集到同一个簇中,并更新簇的特征,使簇的特征值可以更好地体现簇中文本主题。通过实验分析,提出的聚类算法不仅能大幅降低文本表示模型的维度,而且聚类效果提升也比较明显。

参考文献:

[1] 王洪佳,邢长征,王 星. 基于相对密度的多耦合文本聚类算法[J]. 计算机应用研究,2016,33(6):1624-1627.

[2] 符保龙,张爱科. 中心聚类和语义特征融合的网页信息文本挖掘方法[J]. 辽宁工程技术大学学报:自然科学版,2016,35(1):85-88.

[3] PARK H,KWON K,KHIATI A Z,et al. Agglomerative hierarchical clustering for information retrieval using latent semantic index[C]//2015 IEEE international conference on smart city/socialcom/sustaincom. Chengdu:IEEE,2016:426-431.

[4] SALTON G. A vector space model for automatic indexing[J]. Communications of the ACM,1975,18(11):613-620.

[5] 翟东海,鱼 江,高 飞,等. 最大距离法选取初始簇中心的 K-means 文本聚类算法的研究[J]. 计算机应用研究,2014,31(3):713-715.

[6] BHARTI K K,SINGH P K. Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering[J]. Expert Systems with Applications,2015,42(6):3105-3114.

[7] ABUALIGAH L M,KHADER A T,ALBETAR M A. Unsupervised feature selection technique based on genetic algorithm for improving the text clustering[C]//Proceedings of the 2016 7th international conference on computer science & information technology. Amman:IEEE,2016:1-6.

[8] HE W,CHENG X,HU R,et al. Feature self-representation based hypergraph unsupervised feature selection via low-rank representation[J]. Neurocomputing,2017,253:127-134.

[9] TRAN T,VO B,LE T T N,et al. Text clustering using frequent weighted utility itemsets[J]. Cybernetics and Systems,2017,48(3):193-209.

[10] 彭 敏,黄佳佳,朱佳晖,等. 基于频繁项集的海量短文本聚类与主题抽取[J]. 计算机研究与发展,2015,52(9):1941-1953.

[11] 董振东,董 强,郝长伶. 知网的理论发现[J]. 中文信息学报,2007,21(4):3-9.

[12] 朱新华,马润聪,孙 柳,等. 基于知网与词林的词语语义相似度计算[J]. 中文信息学报,2016,30(4):29-36.

[13] 刘 群,李素建. 基于《知网》的词汇语义相似度计算[J]. 中文计算语言学,2002,7(2):59-76.

[14] ZHANG H P,YU H K,XIONG D Y,et al. HHMM-based Chinese lexical analyzer ICTCLAS[C]//Proceedings of the second SIGHAN workshop on Chinese language processing. Sapporo,Japan:ACL,2003:184-187.

[15] 刘怀亮,杜 坤,秦春秀. 基于知网语义相似度的中文文本分类研究[J]. 现代图书情报技术,2015(2):39-45.