一般间隙序列模式挖掘的关键词抽取

刘慧婷[1,2],刘志中[1,2],王利利[1,2],吴信东[3,4]
(1. 安徽大学计算智能与信号处理教育部重点实验室,安徽合肥 230601;
2. 安徽大学计算机科学与技术学院,安徽合肥 230601;
3. 合肥工业大学计算机与信息学院,安徽合肥 230601;
4. School of Computing and Informatics, University of Louisiana at Lafayette, Lafayette 70503)

摘 要: 本文提出了有监督的关键词抽取算法——KEING(Keyphrase Extraction using sequentIal patterns with oNe-off and General gaps condition)算法.首先,将每篇文档作为一个序列库,利用SPING(Sequential Patterns mIning with oNe-off and General gaps condition)算法获取词语之间的关系及其多种变化形式,并利用统计模式特征的方式描述候选关键词;然后,通过朴素贝叶斯分类算法对大量带标记的训练数据进行训练,构造分类器;最后利用分类器从测试文档中识别出关键词.通过实验验证了SPING算法的完备性以及KEING算法的有效性.

关键词: 一般间隙; 模式挖掘; 关键词抽取; 机器学习
中图分类号: TP181 **文献标识码:** A **文章编号:** 0372-2112 (2019)05-1121-08
电子学报URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2019.05.020

Keyphrase Extraction Using Sequential Patterns Mining Algorithm with One-Off and General Gaps Condition

LIU Hui-ting[1,2], LIU Zhi-zhong[1,2], WANG Li-li[1,2], WU Xin-dong[3,4]
(1. *Key Laboratory of Intelligent Computing and Signal Processing of the Ministry of Education, Anhui University, Hefei, Anhui* 230601, *China*;
2. *School of Computer Science and Technology, Anhui University, Hefei, Anhui* 230601, *China*;
3. *School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, Anhui* 230601, *China*;
4. *School of Computing and Informatics, University of Louisiana at Lafayette, Lafayette* 70503, *USA*)

Abstract: Keyphrases are used to summarize the document and high-quality keyphrases have great importance in text summarizing, reading and indexing. However, most studies of keyphrase extraction have strict limitation in the form of patterns, and are unable to achieve the semantic relation between words and phrases. The results are failure to autonomously extract keyphrases. Keyphrase extraction using sequential patterns mining with one-off and general gaps condition algorithm (KEING) is proposed in this paper. Taking into account one off condition and general gaps, SPING (Sequential Patterns mIning with oNe-off and General gaps condition) can catch semantic relations between words and phrases more effectively. Therefore, KEING will get effective candidate keyphrases and count their features. Then a supervised machine learning method is used to train features and construct a classification model, we can extract keyphrase with this model. Experimental results demonstrate KEING can effectively extract high quality keyphrases.

Key words: general gap; sequential patterns mining; keyphrase extraction; machine learning

1 引言

随着大数据时代的到来[1],关键词抽取技术已经被广泛的应用到文档的自动摘要[2]、文档的分类聚类、信息检索[3]等重要领域.美国IBM公司的Luhn提出了基于词频统计的文献自动标引方法,标志着关键词抽取研究和应用的开始[4].

目前关键词抽取算法主要分为无监督和有监督的学习算法.无监督的学习算法可归纳为三种:基于统计特征的关键词抽取、基于主题模型的关键词抽取和基

收稿日期:2018-03-05;修回日期:2018-08-03;责任编辑:蓝红杰
基金项目:国家重点研发计划(No. 2016YFB1000901);国家自然科学基金(No. 61202227);安徽高校自然科学研究项目(No. KJ2018A0013)

于图的关键词抽取[5]. 其中 KP-Miner 算法为经典的无监督关键词抽取算法[6]. 有监督的学习算法将关键词提取问题视为分类问题,其中代表性的算法有 KEA 算法[7],Extractor 算法[8]. 有监督的算法通过大量的文档的训练,以及构造不同的分类器,可以达到比较好的关键词提取效果. 典型的关键词抽取算法如表 1 所示.

表 1 典型的关键词抽取算法

方法类型	作者	简单描述
无监督的关键词抽取方法	Barker[9]	基于短语的长度、频率、第一个词频度的抽取方法
	Steier、Belew[10]	基于两个词语间的互信息的抽取方法
	Nedelina Teneva[11]	基于主题建模的关键词抽取方法
	Aytug Onan 等人[12]	基于独立于领域的相关特征的抽取方法
	Qingren Wang[13]	基于带间隙模式挖掘和带熵的模式频率的抽取方法
有监督的关键词抽取方法	Mounia[14]	基于文档短语极大性指数和逻辑回归的抽取方法
	Medelyan、Witten[15]	基于专业词库和决策树的抽取方法
	Sujatha Das Gollap-alli[16]	基于序列标记的关键词抽取方法

基于无监督学习的关键词抽取研究中,Barker 和 Cornacchia[9]首次提出利用名词作为候选关键词,然后分析各个名词或者名词短语的频度,长度,第一次出现的位置构造关键词抽取模型. Nedelina Teneva 等人[11]提出了基于主题模型的关键词抽取算法. Aytug Onan 等人利用独立于领域的特征构造关键词抽取模型[12].

有监督的关键词抽取算法中 KEA 系统采用朴素贝叶斯的分类算法训练词语的离散特征值,随后 Medelyan 等人[15]在 KEA 算法基础上结合专业词库提出了KEA ++ 算法;GenEx 系统则是利用 C4. 5 决策树和遗传算法抽取关键词. Sujatha Das Gollapalli 等人把关键词抽取看作序列标注问题[16].

目前大多数关键词抽取算法中的主要步骤通过频繁模式挖掘的方法找出高质量的候选关键词. 而若只根据词语在文档中出现的次数作为关键词的评判标准,则无法获得词语之间的语义关系,导致关键词抽取质量较低. 为了抽取高质量的关键模式,xie 等人提出了基于带通配符序列模式的关键词抽取算法[17],该方法获取的词语不仅词频较高,而且具有强关联性、灵活性的特点. 但是,一篇文档中意思相似的关键内容会反复的出现,同时为了保证表达的多样性,词语的形式和位置会发生变化,如 firmware update 与 update firmware 都表示固件更新升级,topic model 与 model topic 都有主题模型的含义,而非负间隙的模式挖掘无法对上述这种顺序发生变化的短语的出现次数进行准确的统计. 因此,本文提出了利用一般间隙的序列模式挖掘的方式进行关键词挖掘,允许模式中位置颠倒现象的出现,不仅获得了高词频的词语,而且获取了词语之间位置变化的复杂关系,进一步提高了匹配的灵活性.

综上,目前研究中关键词抽取的质量并不太理想. 因此,本文在一般间隙序列模式匹配,序列模式挖掘的基础上,提出了新的关键词抽取算法. 利用一般间隙序列模式挖掘从文档中获取词语的模式特征,并构造分类模型,进而抽取关键词. 通过大量的实验结果验证,本文提出的算法可以提高抽取的关键词的质量.

2 问题定义

本文提出的关键词抽取算法的基础是序列模式挖掘,序列模式挖掘通过模式匹配算法计算模式在序列以及序列库中出现的次数,然后将模式出现的次数作为关键词抽取的特征值. 下面将具体的介绍序列模式挖掘以及关键词抽取算法中的基本概念.

定义 1 序列串 $S=\{(s_0s_1\cdots s_i\cdots s_{n-1})\mid s_i\in\Sigma,1\leqslant i\leqslant n-1\}$,其中,$n$ 表示序列串 S 的长度. Σ 代表符号集,应用场景不同 Σ 代表的符号也不相同,在关键词抽取方面 Σ 由文档中的基本词语构成. 序列数据库 $SeqDB=\{S_1,\cdots,S_N\}$ 表示 N 条序列的集合.

定义 2 具有间隙约束的模式串 P 可以表示成:$P=\{(p_0\cdots p_j\cdots p_{m-1})\mid p_j\in\Sigma\}$,其中,$m$ 表示模式串的长度,P_j 与 p_{j+1}之间的间隙约束为 $g=[\min,\max]$,$0\leqslant j\leqslant m-1$,且满足 $\min\leqslant\max$. 若模式串 P 中的间隙满足 $\min<0$ 的条件,则模式串 P 称为一般间隙模式串,否则模式串 P 称为非负间隙模式串.

定义 3 若一个序列 S 中的位置索引 $occ=<o_0,o_1,\cdots,o_j,\cdots,o_{m-1}>$,其中,$1\leqslant j\leqslant m-1$,$0\leqslant o_j\leqslant n-1$ 满足以下的条件:

$$S_{o_j}=p_j \tag{1}$$

$$o_{j-1}\neq o_j \tag{2}$$

$$\begin{cases}\min\leqslant o_j-o_{j-1}-1\leqslant\max, & \text{if } o_{j-1}<o_j\\ \min\leqslant o_j-o_{j-1}\leqslant\max, & \text{if } o_{j-1}>o_j\end{cases} \tag{3}$$

则称 occ 是模式串 P 在序列串 S 中满足间隙约束的一个出现. 如果 occ 是模式 P 在 $S_j\in SeqDB$ 的一个出现,则可以表示为$(j,<o_0,o_1,\cdots,o_j,\cdots,o_{m-1}>)$.

定义 4 给定两个出现 $occ=(j,<o_0,o_1,\cdots,o_k,\cdots,o_{m-1}>)$,$occ'=(j',<o_0',o_1',\cdots,o_q',\cdots,o_{m-1}'>)$,如果满足 $j\neq j'$并且 $\forall k,q,o_k^j\neq o_q^{j'}$,$0\leqslant k,q\leqslant m-1$,则称为出现

occ 与 *occ*′满足 one-off 条件.

定义 5　模式 P 在序列集 *SeqDB* 出现的次数称为支持度,表示成 $Sup(P)$,如果模式 P 的支持度不小于给定的支持度阈值 min_*sup*,即 $Sup(P) \geqslant \min\_sup$,则模式 P 称为频繁模式.

定义 6　给定模式串 $P = p_0 \cdots p_j \cdots p_{m-1}$,$Q = q_0 q_1 \cdots q_k \cdots q_{n-1}$ $(n \geqslant m)$,(P,Q 为带有通配符的模式串),如果存在一个位置序列 $0 \leqslant i_1 < i_2 < \cdots < i_m < n$,并且满足 $p_j = q_{i_j}, 0 \leqslant j < m$,则模式 P 称为模式 Q 的子模式,Q 称为 P 的超模式,表示成 $P \subseteq Q$. 如果同时满足 $i_j = i_{j+1} - 1, 0 \leqslant j < m-1$,则模式 P 称为模式 Q 的连续子模式,Q 称为 P 的连续超模式. 当且仅当 $Sup(P) = Sup(Q)$ 时,模式 P 称为闭模式.

3　KEING 算法的分析与设计

3.1　算法概述

本文提出的基于一般间隙序列模式挖掘的有监督的关键词抽取算法,总体上可以分为训练以及测试两个阶段. 其中,训练阶段主要是构造分类器,首先需要对文档进行预处理,即通过停用词表,以及 poster stemmer[18] 处理单词,并且将每一篇文档作为一个序列数据库 *SeqDB*,文档中的每一段作为一条序列 S;然后利用一般间隙序列模式挖掘算法 SPING 统计词频,并获取频繁模式的其他特征值;最后利用朴素贝叶斯分类器进行训练,构造分类器. 测试阶段主要是验证训练阶段构造的分类器的性能,在测试文档中通过比较抽取的关键词与真实关键词计算算法的有效性. 算法模型如图 1 所示.

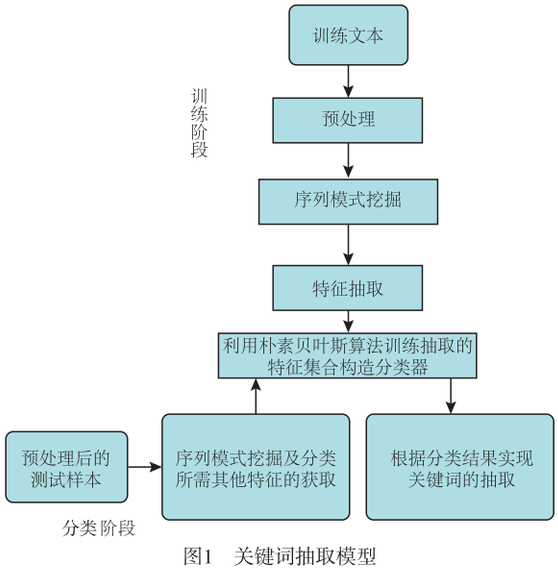

图1　关键词抽取模型

3.2　SPING 算法设计

SPING 算法是在一般间隙与 one-off 条件下进行的模式挖掘,其中利用基于 one-off 条件的一般间隙模式匹配算法 MSAING[19] 计算支持度,由 Apriori 性质可知,当支持度不小于支持度阈值时,则为频繁模式,继续扩展该模式并判断其超模式是否频繁,若该模式的支持度小于支持度阈值则停止扩展,直至遍历完所有的频繁模式.

算法 1 给出了 SPING 算法的总体框架. 首先,将序列集 *SeqDB* 的元素集 Σ 作为初始模式(第 1 ~2 行),第 3 ~4 行对每一个模式,调用算法 2 FPMine,判断是否为频繁模式,如果频繁则加入到频繁项集 *FP* 中.

算法 1　SPING(*SeqDB*,min,max,min_*sup*)算法

输入:序列集 *SeqDB*;最小间隙 min,最大间隙 max;支持度阈值 min_*sup*
输出:频繁模式集 *FP*

```
1.  Σ = {SeqDB 中长度为 1 的模式集}
2.  for each e ∈ Σ do
3.      P = e;
4.      FPMine(SeqDB,P,min_sup);//调用算法 2
5.  end for
```

算法 2 FPMine 首先通过调用算法 3——基于 one-off 条件的一般间隙模式匹配算法 MSAING 算法计算模式 P 在序列集 S 上满足一般间隙与 one-off 条件下的支持度 $Sup(P)$(第 1 行),如果支持度不小于阈值,则模式 P 为频繁模式,加入到频繁模式集 *FP* 中. 然后扩展模式,递归的调用 FPMine 算法,判断新生成的模式 P' 是否为频繁模式,直至所有的频繁模式集都为空(第 2 -8 行). 其中,采用 Apriori 性质作为剪枝策略,即当子模式 P 为非频繁模式时,其超模式 P' 也必为非频繁模式,因此不需要再扩展模式 P.

算法 2　FPMine(*SeqDB*,*P*,min_*sup*)

输入:序列集 *SeqDB*;模式串 P,支持度阈值 min_*sup*
输出:前缀为模式串 P 的频繁模式集

```
1.  Sup(P) = MSAING(SeqDB,P);//调用算法 3
2.  if (Sup(P) ≥ min_sup) then
3.      FP = FP ∪ P
4.      for each e ∈ Σ do
5.          P' = P◇e;
6.          FPMine(SeqDB,P',min_sup);
7.      end for
8.  end if
```

算法 3　MSAING 算法

输入:$P = p_0 p_1 \cdots p_{m-1}$,$g = [\min, \max]$,$S = s_0 s_1 \cdots s_{n-1}$
输出:最大的出现数目 $|OCC_{\max}|$

```
1.  根据序列串,模式串结构分析判断是否转置
2.  for i = 0 to i < n
3.    if(s_i = p_{m-1}) then // Locate phase
4.      建立搜索表 table,如果搜索表 table 建立成功,布尔型变量 flag_table 为 true
5.      if(flag_table)
6.        occ = 采用最左优先策略,利用回溯机制遍历 p 在 table 中的位置
7.      end if
8.    if(occ 存在)
9.      OCC_SPMGOO = OCC_SPMGOO ∪ occ
10.     for j = 0 to m - 1
11.       used[occ_j] = true
12.     end for
13.   end if
14.  end if
15.  end for
16.  return |OCC_SPMGOO|
```

MSAING 算法主要是计算模式的支持度,为了提高匹配的完备性,首先通过对模式串 P 的形式分析,判断是否需要转置(第 1 行). 然后建立搜索表(第 3 ~ 4 行). 如果搜索表建立成功,采用回溯机制遍历模式串在搜索表中出现的位置,其过程采用最左优先策略(第 5 ~ 7 行). 如果存在一次出现,则把出现的结果加到 OCC_{SPMGOO} 集合中,同时根据 one-off 条件标记 $used[occ_j] = true$,直至序列中模式的出现位置全都遍历完,该算法结束,输出该模式的出现次数(第 8 ~ 16 行).

一般间隙与 one-off 条件的模式匹配问题需要考虑内部重复出现的情况,因此本文提出了内部检测机制,通过减少重复检测的次数,提高执行效率. 具体检测过程如算法 4 所示. 本文在 MASING 算法模式匹配的过程中增加了一个布尔型的标志数组 *incheck*,用于标记 p_j 和 p_i 的出现位置是否有重复的可能(其中,$0 \leqslant j < i \leqslant m-1$),*incheck* 的初始值为 false(第 1 行). 由上述分析可知,只有当 min < 0 ≤ max,需要进一步验证(第 2 行),则从 p_{m-3} 开始依次向前查找,检查 p_{m-3} 是否会和 p_{m-1} 发生重复. 如果 $p_{m-3} = p_{m-1}$ 则将 p_{m-3} 的 *incheck* 设置为 true,否则继续向前检测 p_{m-4},以此类推直到检测完 p_0 为止(3 - 11 行).

算法 4 inside_Checking

```
输入:模式串 P,间隔约束 g = [min,max]
输出:标志数组 incheck
1.  初始化 incheck 标志数组全为 false
2.  if(min < 0 ≤ max) then
3.    for i = m - 3 down to 0
4.      for j = i + 2 to m - 1
5.        if(p_i = p_j) then
6.          incheck[i] = true;
7.          break;
8.        end if
9.      end for
10.   end for
11.  end if
12.  return incheck
```

3.3 KEING 算法

本文研究的关键词抽取算法 KEING 属于有监督学习的关键词抽取算法. 因此,特征值的选取对于抽取关键词质量有着重要的影响. 本文选取的特征值如表 2 所示.

表 2 词语的基本特征和模式特征

类别	特征	模式
基本特征	TFIDF	TFIDF
	Pos	词语第一次出现的位置
模式特征	*Sup*	词语所在的最长模式的支持度
	len	词语所在模式的最大长度
	sup * *len*	词语所在最长模式的支持度与长度的乘积
	closedNum	闭合模式的数据

关键词抽取算法 KEING 如算法 5 所示.

算法 5 KEING (*TD*, *d*, min, max, min_*sup*)

```
输入:带有关键词标记的训练集 TD;测试文档 d;间隙约束 g = [min,max];支持度阈值 min_sup.
输出:测试文档 d 中抽取的关键词
1.  //训练阶段
2.  for each document t ∈ TD
3.    SeqDB = 将文档 t 转换成序列集//预处理阶段
4.    FP = SPING(SeqDB, min, max, min_sup)//模式挖掘阶段调用 SPING 算法
5.      for each cw ∈ FP do
6.        提取候选关键词 cw 的模式特征值 PF_cw
7.        V_cw 矢量表示候选模式特征值
8.        VC_cw = VC_cw ∪ V_cw//构成候选模式的矢量集合
9.      end for
10.   end for
11.   KEModel = Naive Bayes(VC_cw)//利用朴素贝叶斯算法训练候选词矢量集合,形成分类器
12.   //分类阶段
13.   SeqDB' = 文档 d 转换成序列集//预处理阶段
14.   FP' = SPING(SeqDB', min, max, min_sup)//模式挖掘阶段,调用 SPING 算法
15.   for eachcw ∈ FP' do
16.     提取候选关键词 cw 的模式特征值 PF'_cw
17.     V'_cw 矢量表示候选模式特征值
18.     P_cw = KEModel(V'_cw)//利用分类器计算候选关键词作为关键词的概率
19.   end for
20.   KW[ ] = 选取 top - k 个高概率的候选关键词
```

KEING 算法分为训练阶段(第 1 ~ 11 行)和分类阶

段(第 12 ~ 20 行). 其中,第 3 行为预处理阶段,将训练集中的文档转化为序列集;第 4 行在转化的序列集上调用 SPING 算法,找出该文档中的频繁模式;然后提取每一个频繁模式的特征值,并使用矢量的形式表示特征值,最后获取所有频繁模式特征值的矢量集合 VC_{cw}(第 5 ~ 9 行). 采用朴素贝叶斯分类器对集合 VC_{cw} 进行训练,获得分类器 KEModel. 分类阶段与训练阶段步骤相似,只是在第 18 行,利用分类器计算候选关键词称为关键词的概率,并且根据概率大小进行排序,选择前 top-k[20] 个高概率的候选词作为关键词.

4　实验结果与分析

4.1　数据集与评价指标

本节采用真实的生物 DNA 序列验证序列挖掘 SPING 算法的有效性,同时采用 SemEval2010 文本数据库验证关键词挖掘 KEING 算法的性能. 其中,DNA 序列可以从美国国家生物计算信息中心的网站下载. SemEval2010 是 ACL SemEval 出版的 244 篇文章构成的文本数据库,其中分为两部分,144 篇文章作为训练集,其中 40 篇用来验证训练过程中关键词抽取的质量;另外 100 篇文章作为测试集,检测 KEING 算法抽取关键词的质量. 实验运行的软、硬件环境为:Intel(R) Core(TM) i3-4170 CPU@ 3. 70GHz,8. 0GB 内存,操作系统为 Window7,64 位操作系统,程序使用 Java 语言编写,并且采用 eclipse 集成开发环境进行编译和运行. 本文使用准确率 P,召回率 R 以及平衡两者关系的 $F1$ 值作为关键词抽取质量的评价标准,具体公式如下所示.

$$P = \frac{\#correct}{\#extracted}$$

$$R = \frac{\#correct}{\#labeled}$$

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

其中,*#correct* 表示抽取正确的关键词的个数;*#extracted* 表示设定抽取的关键词的个数,即选取 top-k 个高概率的候选关键词时的 k 值;*#labeled* 表示带标签的关键词的个数.

4.2　三种序列挖掘算法性能比较

为了验证序列模式挖掘 SPING 算法的有效性,本文分别与 One-off Mining 算法[21],MPP 算法[22] 和 SPMW 算法[17] 进行比较.

在序列 AX829174 上随机的选择不同长度的子序列作为测试序列,同时间隙约束设置为[9,12],支持度阈值设置为 0. 015 * |S|. 表 3 列出了不同模式挖掘算法在序列长度分别为 1000 到 5000 的挖掘出的频繁模式数,图 2 描述了不同序列模式长度的条件下各个挖掘算法运行的时间.

表 3　SPING 与 SPMW、One-off Mining 在不同长度的序列上挖掘频繁模式的个数

|S|	1000	2000	3000	4000	5000
One-off Mining	3413	4422	5642	5325	4953
SPMW	11299	17516	24966	23171	20772
SPING	11351	17659	25311	23409	20993

由表 3 可知,SPING 算法挖掘的频繁模式平均是 One-off Mining 的 4 倍,比 SPMW 高出 1% 左右. 这是由于 One-off Mining 算法只考虑最左优先策略,导致在满足间隙约束的条件时很多位置在下次匹配过程中无法被使用. SPMW 算法是利用水平实例图的结构进行模式匹配,但是采用最左优先的贪心策略,当完成一个匹配后并没有考虑对下一次匹配位置的影响,只能达到局部最优;而且 SPMW 算法没有考虑序列以及模式结构的特点.

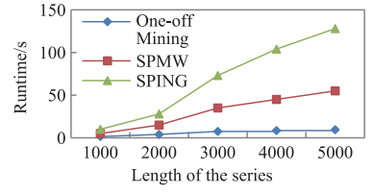

图2　SPING 与SPMW、One-off Mining运行时间比较

从图 2 看出 SPING 算法运行所消耗的时间最多,这是由于该算法考虑了一般间隙的情况,因此需要对模式颠倒的情况进行判断是否匹配,而且在序列模式匹配算法 MSAING 中利用了回溯机制,增加匹配解的完备性,从而导致了时间消耗的增加.

根据序列 AX829170 上包含的 3737 个字符,随机的选取任意长度的序列作为子序列. 间隙约束设为[9,12],MPP 的支持度阈值设为 0. 0003,而为了保持与 MPP 算法在相同的条件进行比较,算法 SPMW 与 SPING 的支持度阈值分别设置为 28、54、79、98. 表 4 列出了不同模式挖掘算法在序列长度分别为 1000 到 3737 挖掘出的频繁模式数,图 3 描述了不同序列模式长度的条件下各个挖掘算法运行消耗的时间. 为了验证 SPING 算法的性质,图 4 显示了在 AX829170 序列上选取不同长度的子序列,在不同间隙约束条件下 SPING 算法挖掘出的频繁模式的个数.

由图 4 可知,频繁模式的个数随着序列长度的增加而增加,而随着阈值的增加而减小. 这是因为当序列越长时,可用来匹配的位置也越多,则相应的模式匹配出现的解也越多,从而频繁模式增加. 然而,当阈值变大时,满足条件的模式则相应的减少.

表4 SPING 与 SPMW、MPP 在不同长度的序列上挖掘频繁模式的个数

$\|S\|$	1000	2000	3000	3737
MPP	2405	2238	2295	2136
SPMW	2534	2374	2304	2213
SPING	2572	2389	2337	2246

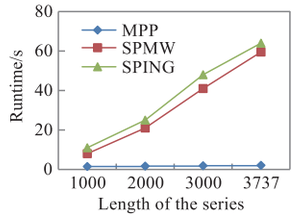

图3 SPING与SPMW、MPP运行时间比较

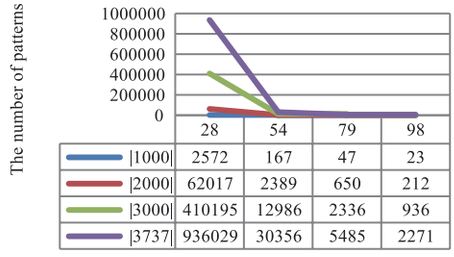

图4 随着序列长度以及阈值变化SPING挖出的频繁模式的个数

4.3 四种关键词抽取算法性能比较

下面从准确度 P,召回率 R 以及 F1 值 3 个指标评价 KEING 算法与 KEA 算法,Extractor 算法,Key_Ex 算法抽取关键词的质量.

由图 5,6 可知,当抽取的关键词数目从 3 个增加到 25 个时,抽取关键词的准确率在不断的减少,而召回率却在不断的增加. 主要是由于随着抽取关键词数目的增加,将抽取更多正确的关键词,因此召回率增加;然而抽取正确的关键词增长速度小于抽取关键词增加的数目,因此导致准确度下降.

图 7 是关于 F1 值的比较,从 F1 值的比较中可知,在抽取关键词数目由 3 个增加到 15 个的过程中,F1 值在不断的增加的,当关键词抽取的数目达到 15 个时 F1 值达到峰值.

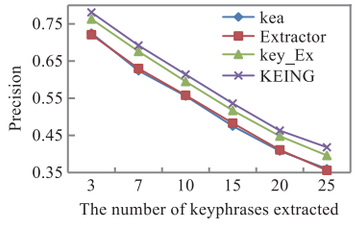

图5 关键词抽取算法准确度比较

同时从图 5 ~ 7 可以发现,KEING 算法相对于算法 KEA,Extractor,Key_Ex 均取得最好的结果. 这是由于

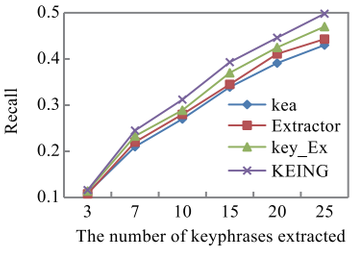

图6 关键词抽取算法召回率比较

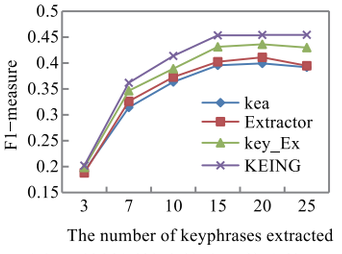

图7 关键词抽取算法F1值比较

KEING 算法不仅采用有监督机器学习算法构造分类器进行分类,而且它还是基于一般间隙模式挖掘的关键词抽取策略. 模式挖掘将单词,词组视为模式中的项,不仅能够获取词语的基本特征,还能够有效的获取模式特征,以及词语之间的关系,因此 KEING 算法提高了关键词抽取的质量.

4.4 KEING 算法提取不同特征值的性能比较

KEING 算法是基于有监督学习的关键词抽取算法,如何选取特征值对关键词抽取的质量有着很大的影响,因此在本实验中,研究特征值选取的不同对关键词抽取算法性能的影响. 本文将对比下面 4 组不同的特征子集对关键词抽取质量的影响.

Ⅰ:$sup + len + pos$

Ⅱ:$closedNum + pos$

Ⅲ:$sup * len + pos$

Ⅳ:$TFIDF + pos$

由图 8 ~ 10 可知当选取基本特征与模式特征相结合的方式抽取关键词时关键词的提取质量较高,其中 $TFIDF + pos$ 的方式实质为 KEA 算法,而 $sup * len + pos$ 的方式关键词抽取的效果最好. 因为,$TFIDF + pos$ 只是基本特征,没有考虑词语之间的关系,因此抽取的质量最低;而 $sup * len + pos$ 不仅考虑了词语在文中出现的位置,出现的频度,还考虑了词语的长度,由于词语的频度越高,长度往往越小,而词语的频度显示了词语的重要程度,长度表明了词语表达的信息量,因此通过 $sup * len$ 将频度与长度相结合提高关键词提取的质量.

通过上述实验表明,基本特征 + 模式特征的提取方式能够有效的提高关键词的抽取质量,而模式特征值受到选取的支持度阈值、最大间隙大小的影响. 下面

在提取 *sup* * *len* + *pos* 特征的条件下,利用控制变量法,改变支持度阈值,最大间隙进行实验比较,结果如图 11~12 所示.

由图 11,12 可知,当支持度阈值为 3 时,F1 值达到较大值;而当最大间隙达到 10 时,F1 值达到峰值,并随着间隙的增大而趋于平缓. 因此,通过调整支持度阈值,以及间隙可以提到关键词的提取质量.

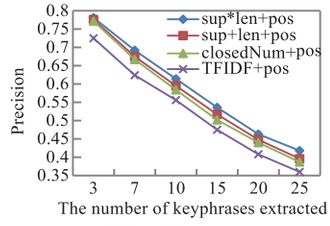

图8　KEING算法提取不同特征值的准确度的比较

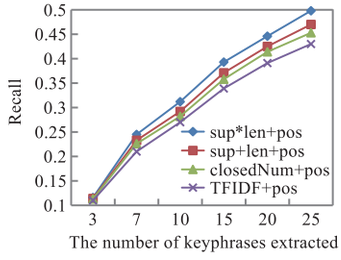

图9　KEING算法提取不同特征值的召回率的比较

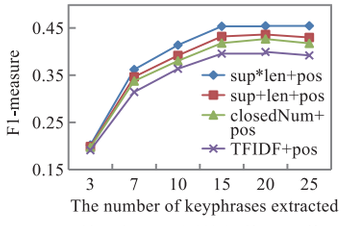

图10　KEING算法提取不同特征值的F1值的比较

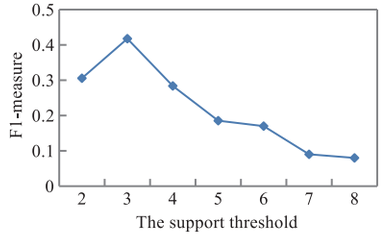

图11　KEING算法F1值与阈值之间关系

4.5　KEING 算法选取不同分类器的性能比较

本节通过实验验证了在关键词抽取方面,使用不同的分类算法训练相同的数据集合得到不同的分类模型后,抽取出的关键词质量的差别. 本文分别选取了朴素贝叶斯、随机森林、C4.5 和 SVM 这 4 种分类器,分类器的代码来自于开源机器学习平台 Weka. 由图 13 可以看出利用朴素贝叶斯模型训练带有关键词标记的文本

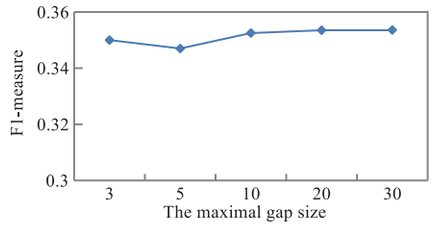

图12　KEING算法F1值与间隙之间关系

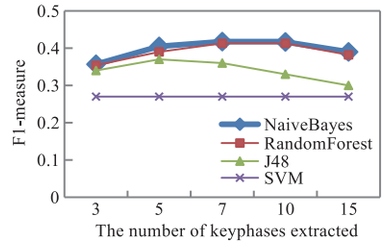

图13　KEING算法F1值与分类器之间的关系

生成的分类模型,抽取的关键词最准确,所以在 KEING 算法中,使用朴素贝叶斯模型来生成分类模型.

5　结束语

本文提出了基于一般间隙序列模式挖掘的关键词抽取算法 KEING,该算法利用一般间隙条件允许用户更加灵活的设定词语间隙,能够更加有效的获取词语间的关系. 本文首次把一般间隙应用到关键词抽取问题的研究中,首先通过一般间隙序列模式匹配算法 MSAING 计算词语出现的频度,然后利用一般间隙序列模式挖掘算法 SPING 获取候选关键词,接着通过朴素贝叶斯分类器在大量训练集中的训练,构造分类器,最后通过分类器获取关键词,并评估关键词抽取的质量. 本文通过序列库以及文本库中的实验验证了 SPING 算法、KEING 算法的有效性.

在本文中关键词出现的最小阈值是人为设定的,进一步研究中可以针对每篇文档动态的设置最小阈值,从而有效的提高关键词抽取的质量.

参考文献

[1] Zhang J, Wang W, Wei X, et al. Climate analytics workflow recommendation as a service-provenance-driven automatic workflow mashup[A]. Proceedings of 2015 IEEE International Conference on Web Services[C]. New York, USA:IEEE Computer Society, 2015. 89-97.

[2] 赵京胜,朱巧明,周国栋,张丽. 自动关键词抽取研究综述[J],软件学报,2017,28(9):2431-2449.
Zhao JS, Zhu QM, Zhou GD, Zhang L. Review of research in automatic keywordextraction[J]. Journal of Software, 2017, 28(9):2431-2449. (in Chinese)

[3] Liu X, Song Y, Liu S, et al. Automatic taxonomy construc-

tion from keywords[A]. Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. Beijing, China: ACM, 2012. 1433 – 1441.

[4] Luhn H P. A statistical approach to mechanized encoding and searching of literary information[J]. IBM Journal of Research and Development, 1957, 1(4): 309 – 317.

[5] 马慧芳,刘芳,夏琴,等. 基于加权超图随机游走的文献关键词提取算法[J]. 电子学报, 2018, 46(6): 1410 – 1414.
MA HF, LIU F, XIA Q, et al. Keywords extraction algorithm based on weighted hypergraph random walk[J]. Acta Electronica Sinica, 2018, 46(6): 1410 – 1414. (in Chinese)

[6] El-Beltagy S R, Rafea A. KP-Miner: A keyphrase extraction system for English and Arabicdocuments[J]. Information Systems, 2009, 34(1): 132 – 144.

[7] Witten I H, Paynter G W, Frank E, et al. KEA: practical automatic keyphraseextraction[A]. Proceedings of ACM Conference on Digital Libraries[C]. Berkeley, USA: ACM, 1999. 254 – 255.

[8] Turney P D. Learning algorithms for keyphraseextraction[J]. Information Retrieval, 2000, 2(4): 303 – 336.

[9] Barker K, Cornacchia N. Using noun phrase heads to extract document keyphrases[A]. Proceedings of Canadian Conference on AI 2000[C]. Montréal, Canada: Springer, 2000. 40 – 52.

[10] Steier A M, Belew R K. Exporting phrases: a statistical analysis of topicallanguage[A]. Second Symposium on Document Analysis & Information Retrieval[C]. Cite Seer, 1993. 179 – 190.

[11] Teneva N, Cheng W. Salience rank: efficient keyphrase extraction with topicmodeling[A]. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics[C]. Vancouver, Canada: ACL, 2017. 530 – 535.

[12] Onan A, Korukoglu S, Bulut H. Ensemble of keyword extraction methods and classifiers in text classification[J]. Expert Systems with Applications, 2016, 57: 232 – 247.

[13] Wang Q, Sheng V S, Wu X. Document-specific keyphrase candidate search and ranking[J]. Expert Systems with Applications, 2018, 97: 163 – 176.

[14] Haddoud M, Abdedda¿m S. Accurate keyphrase extraction by discriminating overlapping phrases[J]. Journal of Information Science, 2014, 40(4): 488 – 500.

[15] Medelyan O, Witten I H. Thesaurus based automatic keyphraseindexing[A]. Proceedings of the ACM/IEEE Joint Conference on Digital Libraries[C]. Chapel Hill, USA: ACM, 2006. 296 – 297.

[16] Gollapalli S D, Li X. Keyphrase Extraction using Sequential Labeling[EB/OL]. arXiv: 1608. 00329[cs. CL], 2016-08-01/2018-03-05.

[17] Xie F, Wu X, Zhu X. Efficient sequential pattern mining with wildcards for keyphrase extraction[J]. Knowledge-Based Systems, 2017, 115: 27 – 39.

[18] Porter M F. An algorithm for suffixstripping[J]. Program Electronic Library & Information Systems, 2013, 14(3): 130 – 137.

[19] 刘慧婷,刘志中,黄厚柱,吴信东. 一般间隙与 One-Off 条件的序列模式匹配[J]. 软件学报, 2018, 29(2): 363 – 382.
Liu H T, Liu Z Z, Huang H Z, Wu X D. Sequential pattern matching with general gap and one-offcondition[J]. Journal of Software, 2018, 29(2): 363 – 382. (in Chinese)

[20] 王新军,闫实,彭朝晖,李庆忠. Extractor: 支持查询重构的高效数据库关键词检索系统[J]. 电子学报, 2014, 42(2): 209 – 216.
WANG X J, YAN S, PENG Z H, LI Q Z. Extractor: a query-reformulation embedded efficient keyword search system over relational databases[J]. Acta Electronica Sinica, 2014, 42(2): 209 – 216. (in Chinese)

[21] Huang Y, Wu X, Hu X, et al. Mining Frequent patterns with gaps and one-off condition[A]. Proceedings of the 12th IEEE International Conference on Computational Science and Engineering[C]. Vancouver, Canada: IEEE Computer Society, 2009. 180 – 186.

[22] Zhang M, Kao B, Cheung D W, et al. Mining periodic patterns with gap requirement fromsequences[J]. ACM Transactions on Knowledge Discovery from Data, 2007, 1(2): 7.

作者简介

刘慧婷 女,1978 年出生,安徽阜阳人,博士,副教授,CCF 专业会员,主要研究领域为数据挖掘、机器学习.
E-mail: htliu@ahu.edu.cn

刘志中 男,1990 年出生,硕士,主要研究领域为数据挖掘.