文章编号：1003-0077(2020)09-0044-09

基于粗糙数据推理的 TextRank 关键词提取算法

周 宁，石雯茜，朱昭昭

（兰州交通大学 电子与信息工程学院，甘肃 兰州 730070）

摘 要：基于图模型的 TextRank 算法是一种有效的关键词提取算法，在提取关键词时可取得较高准确度。但该算法在构造图的关联边时，所采用的共现窗口规则仅考虑了局部词汇间的关联，并具有较大随意性与不确定性。针对这一问题，该文提出了一种基于粗糙数据推理理论的改进 TextRank 关键词提取算法，粗糙数据推理可扩大关联范围，增加关联数据，得到的结果更加全面。结合粗糙数据推理理论中的关联规则，该文提出的算法做了以下改进：依据词义对候选关键词进行划分；再通过粗糙数据推理对不同分类中候选词间的关联关系进行推理。实验结果表明，与传统的 TextRank 算法相比，改进后算法的提取精度有了明显的提高，证明了利用粗糙数据推理的思想能有效地改善算法提取关键词的性能。

关键词：粗糙数据推理；关键词提取；关联规则；TextRank 算法

中图分类号：TP391 **文献标识码**：A

TextRank Keyword Extraction Algorithm Based on Rough Data-Deduction

ZHOU Ning, SHI Wenqian, ZHU Zhaozhao

(School of Electronics and Information Engineering, Lanzhou Jiaotong University, Lanzhou, Gansu 730070, China)

Abstract: TextRank algorithm based on graph model is an effective keyword extraction algorithm with high accuracy. However, when constructing the edges of a graph, the algorithm adopts the co-occurrence window rule that considers only the association between local words, yielding greater randomness and uncertainty. To address the issue, an improved TextRank keyword extraction algorithm based on rough data-deduction is proposed. In this method, candidate keywords are classified according to word meanings, and the association between candidate words in different classes is deduced by rough data-deduction. The experimental results show that the extraction precision of improved algorithm has been significantly improved.

Keywords: rough data-deduction; keyword extraction; association rule; TextRank algorithm

0 引言

互联网的飞速发展使得网络文本数据呈现爆发性的增长。快速发展的互联网在给用户带来便利的同时，也使得用户很难从海量复杂的数据中快速、准确、全面地获取自身所需的信息。关键词被认为是能够描述文本主题的一系列词语，可实现类似摘要的作用，在信息检索以及文本分类方面被广泛应用。对于现实生活中的文本数据，如果能准确地将这些文本内容用几个关键的词语进行描述，就可以使用户更为方便地浏览和获得有效信息，因此关键词的准确提取就显得非常重要。

关键词的提取主要包含有监督和无监督两类方法。有监督提取方法[1]是将关键词的提取转换为一个二分类问题进行分析，判断候选词是否为关键词，同时该方法需要提供已标注的语料；无监督提取方法无须提供标注语料，利用统计性质对候选词排序，取最重要的几个词作为关键词。随着无监督提取方法的不断完善，其提取性能也在逐渐接近有监督的方法[2]，且适应性强，从而被广泛使用。无监督提取算法的主要代表有：基于统计特征的 TF-IDF[3-4]算法、基于主题模型的 LDA 算法[5]以及基于词图模型的 TextRank 算法[6]。其中 TextRank 算法作为

收稿日期：2019-09-16 **定稿日期**：2020-01-14

基金项目：国家自然科学基金(61650207,61841303)；教育部人文社会科学基金(19YJC760012)

基于词图模型的典型代表，具有较好的关键词提取效果，且在一些情况下接近有监督方法的效率。鉴于此，该方法一直以来都得到了研究者们的广泛关注。

近年来，为进一步提高 TextRank 算法的关键词提取效果，部分学者对此算法做出了改进。李鹏等人[7]将社会化标签 Tag 引入算法来估计词图中边的权值，计算词项的重要性，结果表明其提取效果更优。Yijun 等人[8]将 LDA 融合到算法中，考虑到了整体文档集中主题信息的影响力，从而提高了关键词提取的准确度。Zhang 等人[9]提出了一种基于主题权重的改进的 TextRank 算法，充分利用文章自身的丰富内涵，提高了关键词提取的准确率。Wan[10]将时间维度添加到算法中，可以更好地适应不断变化的主题，提高了关键词提取的有效性。肖辛格[11]基于基本层次范畴这一理论，对算法做出了改进。Zuo 等人[12]将 Word2Vec 嵌入到 TextRank 算法中，可以更好地提取关键词。刘竹辰等人[13]将文档中词的位置分布信息融入到算法中，改进了算法对于关键词提取的效果。柳林青等人[14]提出了基于条件概率模型的 TextRank 算法，并将马尔可夫模型引入其中，提升了算法的提取性能。徐馨韬等人[15]将 Doc2Vec 模型与 K-means 算法融入到算法中，提高了关键词提取的质量。传统的 TextRank 算法在构建候选关键词图时会采用共现窗口原则建立节点之间的关联。即在同一窗口内的两节点间可构造一条边，因此利用共现关系可以方便地得出所需词图。但采用共现关系判断节点间的关联性仅考虑了局部关系，较为局限，可能会导致提取结果不够全面、准确。同时由于 TextRank 算法中用到的共现关系对词的顺序位置敏感，在一定程度上会受文本作者写作习惯的影响，因此对于一些文本数据可能会得到一些极端的结果，不够稳定。

为解决这一问题，得到更准确的提取结果，本文引入粗糙数据推理理论对 TextRank 算法做出改进。由于粗糙数据推理具有上近似特性，且推理对象为数据[16]，因此将该理论用于具有潜在关联的问题时，对问题的模型建立与算法模拟都具有重要的应用意义。但目前关于该理论的应用研究较少，仅在图像修复方面有所涉及[17]，还未用于文本语言处理的相关研究，因此基于粗糙数据推理对 TextRank 算法进行改进具有理论和实际意义。本文算法运用粗糙数据推理理论对节点间的关联关系进行推理，判断两节点间是否具有潜在关联，构图进行关键词提取，进而取得更加精确的关键词提取结果。实验结果表明，相较于经典的 TextRank 算法，引入了粗糙数据推理思想的改进算法，得到了更加准确的关键词提取结果。

1　基于粗糙数据推理的改进算法

1.1　TextRank 算法原理

TextRank 关键词提取算法是一种基于词图模型的排序算法。该算法源于谷歌的 PageRank 算法，即先将目标文本分割为若干个有意义的词汇并构建候选词图，再利用投票机制对候选词排序，实现关键词的提取。

关键词提取的任务是从目标文本中提取若干个重要词汇。TextRank 算法是利用词汇间的局部联系（即共现窗口）来确定候选词之间的关联，再对候选关键词进行迭代计算与排序。其主要步骤如下：

（1）分句：将目标文本 T 依据句子的完整性进行分割，即 $T=[S_1,S_2,\cdots,S_m]$。

（2）分词、过滤：对每个句子 $S_i\in T$，执行分词和词性标注，然后过滤掉停用词以及一些不包含在指定词性中的词，即 $S_i=[t_{i,1},t_{i,2},\cdots,t_{i,n}]$，其中 $t_{i,j}$ 是过滤后的候选关键词。

（3）构图：构建候选词图 $G=(V,E)$，G 中包含点集 V 和边集 E。集合 V 由步骤（2）得到的候选词构成；集合 E 是 $V\times V$ 的子集，其利用共现（co-occurrence）窗口构建图中两节点之间的边，只有当两节点所对应的候选词出现在长度为 K 的窗口中时，两节点之间才有边，其中 K 为窗口大小，决定最多可共现的词数。

（4）迭代计算：按照式（1）[6]对各节点的权重进行迭代计算，直到计算结果收敛。

$$\mathrm{WS}(v_i)=(1-d)+d\times\sum_{v_j\in \mathrm{In}(v_i)}\frac{w_{ji}}{\sum\limits_{v_k\in \mathrm{Out}(v_j)}w_{jk}}\mathrm{WS}(v_j)\tag{1}$$

其中，$\mathrm{In}(v_i)$ 代表的是指向节点 v_i 的节点集；$\mathrm{Out}(v_j)$ 代表节点 v_j 所指向的节点构成的集合；w_{ji} 表示由节点 w_j 指向 w_i 边的权重，这里的权重由两词共现次数决定；d 为阻尼系数，表示从图中任一指定节点转移到其他节点的概率，其取值范围是 $[0,1]$，通常取 0.85。若在候选词图中存在某一节点的误差率小于特定的极限值，则认为该节点达到了收敛，通常将此极限值设置为 0.000 1。

(5) 排序：对所得节点权重倒序排序，将前 K 个词作为目标文本的关键词。

1.2 粗糙数据推理思想

1.2.1 粗糙集理论

粗糙集理论最初在文本处理方面的应用是用来进行文本分类，以加快分类的速度，提高分类的准确度[18]。粗糙数据推理思想以粗糙集理论为基础，将上近似概念中的近似信息融入到数据推理过程中。因此对粗糙集理论相关概念知识的介绍，将起到理解粗糙数据推理的作用。下面对粗糙集内的一些相关知识进行简要介绍。

定义 1[19] 设 U 为数据集，R 为 U 上的等价关系，将 U 和 R 所组成的结构记作 $M=(U,R)$，称 $M=(U,R)$ 为近似空间，U 称为论域。

定义 2[19] 设 $M=(U,R)$ 为近似空间，$U/R=\{[a]_R|a\in U\}$ 为 U 相对于 R 的划分，其中 $[a]_R$ 是由 a 确定的 R 等价类。对 U 中的任一子集 $X\subseteq U$，在近似空间 M 内，子集 X 的上近似 $R^*(x)$ 和下近似 $R_*(x)$ 的定义如下：

$$R^*(x)=\bigcup\{[a]_R\mid[a]_R\in U/R\text{ 且}[a]_R\cap X\neq\phi\} \tag{2}$$

$$R_*(x)=\bigcup\{[a]_R\mid[a]_R\in U/R\text{ 且}[a]_R\subseteq X\} \tag{3}$$

即子集 X 的上近似 $R^*(x)$ 等于所有与 X 的交不等于空集的 R 等价类的并；子集 X 的下近似 $R_*(x)$ 等于所有包含在 X 中的 R 等价类的并。

下近似 $R_*(x)$ 是从 X 的内部逼近 X，而上近似的是从 X 的外部逼近。若认为 X 包含的是精确信息，则认为包含在精确信息 X 内部的 $R_*(x)$ 往往比精确信息更加精确，而 $R^*(x)$ 却扩大了精确信息的范围，包含有 X 之外的信息，这样便可以引出粗糙集的概念，即：

当 $R^*(X)\neq R_*(X)$ 时，称 X 是粗糙集；

当 $R^*(X)=R_*(X)$ 时，称 X 是确定集[19]。

由于 $R_*(x)$ 的信息过于精确，$R^*(x)$ 中的信息涵盖了 X，是精确信息的扩展。因此将 $R^*(x)$ 融入粗糙数据推理中，增加推理数据，扩大推理范围，得到的结果也会更加准确。

1.2.2 粗糙推理空间

粗糙推理空间是粗糙数据推理所依赖的结构空间，是对近似空间 $M=(U,R)$ 在内容与结构上的扩充。同时令 $K=\{R_1,R_2,\cdots,R_n\}(n\geqslant 1)$，这里 R_1，R_2，$\cdots$，R_n 指的是 U 上 n 个不同的等价关系；对于 U 上的二元关系 $S\subseteq U\times U$，将其作为推理关系，则称 $W=(U,K,S)$ 为粗糙推理空间[16]。

1.2.3 粗糙数据推理

有别于其他基于逻辑的推理，粗糙数据推理的推理对象是数据。由于现实生活中大多事物与对象均可抽象表示为数据，因此面向数据的推理使用面更加广泛。

对于粗糙数据推理有以下相关定义：设 $W=(U,K,S)$ 是粗糙推理空间，对于 $a\in U$ 及 $R\in K$ 则有：

(1) 设 $b\in U$，如果 $b\in R^*([a-R])$，则称 a 关于 R 直接粗糙推出 b，记作 $a\Rightarrow_R b$。其中 $[a-R]$ 定义为：$[a-R]=\{x\mid x\in U$ 且存在 $z\in[a]_R$，使得 $\langle z,x\rangle\in S\}$ 为 $[a]_R$ 的 S 的后继集；

(2) 设 $b_1,b_2,\cdots,b_n,b\in U$，如果 $a\Rightarrow_R b_1$，$b_1\Rightarrow_R b_2,\cdots,b_{n-1}\Rightarrow_R b_n,b_n\Rightarrow_R b(n\geqslant 0)$，则称 a 关于 R 粗糙推出 b，记作 $a|=_R b$；

(3) 对于 $R\in K$，a 由(1)或(2)推理得到 b 的推理称为 $W=(U,K,S)$ 中有关 R 的粗糙数据推理，简称为粗糙数据推理[16]。

粗糙数据推理可扩大关联的范围，增加关联的数据，如果将该理论运用到 TextRank 关键词提取算法中，从全局出发，通过推理得到两个词汇节点间的关联，以此来构建候选关键词的词图，进而提取关键词，得到的提取结果应更加全面。

1.3 基于粗糙数据推理的改进算法

在经典的 TextRank 关键词提取算法中，文本内的候选关键词是通过共现关系构建的图模型，然后通过平均转移概率矩阵多次迭代计算各节点的权值，直至收敛。收敛后，将词按其权值降序排列，选取前 K 个词作为提取的关键词。这样的做法比较简洁、有效，但具有一定局限性。共现窗口这一规则的使用，只考虑了局部词汇之间的关联，因此可能会将与某一关键词局部联系紧密的一些词提出。但一篇文档的关键词并不仅仅局限于重要词汇周围的一些词，做文本关键词提取时要全面考虑文本中的词汇以及一些存在潜在关联的词，具有潜在关联的词汇会对整个迭代排序过程产生重要的影响，而这种潜在的关联可通过粗糙数据推理这一理论来发掘。因此，本文提出基于粗糙数据推理理论的改进 TextRank 算法。

首先，从词汇的词义出发，根据词汇间词义的相似性，对候选关键词进行划分。由于在一篇文档中可能会出现词自身不同但词义相近的一组词，对同

一重要的内容进行描述，则应考虑相应增加这组词汇的权重，以提高提取结果的准确度。经典的 TextRank 算法中并未从这一方面进行考虑，仅考虑词汇本身，从而忽略了与其词义相近的词对其贡献率。因此，改进的算法将词义考虑在内，通过词义对候选词进行划分，参与后续的关联推理，能够更加有效地提取关键词。

其次，引入粗糙推理空间 $W=(U,K,S)$，对关键词的提取问题进行结构化描述。其中 U 为论域，此处 U 为候选关键词构成的数据集；K 是等价关系的集合，且存在 $R\in K$，使对于 $a,b\in U$，$<a,b>\in R$ 当且仅当 a 与 b 相似；$S\subseteq U\times U$，定义为 $S=\{<u,v>|u,v\in U$ 且 u 与 v 之间存在关联$\}$。

最后，运用粗糙数据推理，假设推理关系如式(4)所示。

$$S=\{<w_1,w_4>,<w_2,w_6>,<w_3,w_6>,<w_6,w_5>\} \tag{4}$$

其中，$w_1\sim w_7$ 为从文本出发经过分词、过滤后筛选出的候选关键词，且此推理关系由推理中关联规则的关联度，即点互信息来确定。

同时对于等价关系 $R\in K$，假设 U 相对于 R 的划分为：

$$U/R=\{\{w_1,w_2,w_3\},\{w_4,w_6\},\{w_5,w_7\}\} \tag{5}$$

这里的等价划分是基于候选词之间的相似度。结合以上信息可得关键词提取中粗糙数据推理示意图，如图 1 所示。

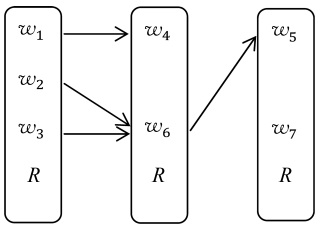

图 1　关键词提取中粗糙数据推理的示意图

图 1 中，在粗糙数据推理的过程中，对于候选词 w_1，算法基于相似度规则得出 w_2,w_3，则划分 w_1,w_2,w_3 为一个数据集，同理可划分 $w_4\sim w_7$；再基于点互信息这一推理中关联规则的关联度由 w_1 推理得到 w_4，同时推理得到 w_5,w_6,w_7。根据粗糙数据推理的定义，对于 w_1 有 $[w_1]_R=\{w_1,w_2,w_3\}$，$[w_1-R]=\{w_4,w_6\}$，$R^*([w_1-R])=\{w_4,w_6\}$，则 $w_1\Rightarrow_R w_6$。对候选词 w_6 有 $[w_6]_R=\{w_4,w_6\}$，$[w_6-R]=\{w_5\}$，$R^*([w_6-R])=\{w_5,w_7\}$，则 $w_6\Rightarrow_R w_7$。由 $w_1\Rightarrow_R w_6$，$w_6\Rightarrow_R w_7$，可得 $w_1|=_R w_7$。由上述可知，w_1 与 w_7 之间也存在潜在的关联关系，可为计算提供一定的贡献率。由上述规则建立候选关键词之间的关联，关联权值可作为贡献率添加到迭代计算过程中，以提高关键词提取的准确度。

改进算法的主要步骤为：

(1) 基于经典的 TextRank 算法，对目标文本进行预处理，包括分句、分词以及词性的过滤，得到候选关键词。

(2) 对候选关键词按照相似性划分不同的等价类。本文我们基于知网(HowNet)与同义词词林进行划分。对于任意两个词语 w_1,w_2，划分规则[20]如式(6)所示。

$$s=\lambda_1 s_1+\lambda_2 s_2 \tag{6}$$

其中，s_1,s_2 分别为利用知网与词林计算得到的相似度；λ_1,λ_2 是赋予 s_1,s_2 的两个权重，要求 $\lambda_1+\lambda_2=1$。由图 2 词语 w_1 与 w_2 在知网和词林的分布图规定 λ_1,λ_2 的取值。

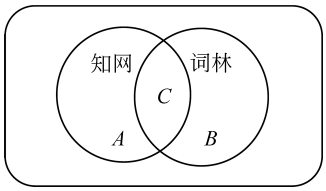

图 2　词语在知网与词林中的分布图

其中，λ_1,λ_2 取值策略[20]如下：

① 当 $w_1\in C,w_2\in C$ 时，分别基于知网与词林对 w_1,w_2 的相似度进行计算，记作 s_1 和 s_2，本文实验取 $\lambda_1=\lambda_2=0.5$；

② 当 $w_1\in A,w_2\in A$ 或者 $w_1\in B,w_2\in B$ 时，对 w_1 与 w_2 基于知网或词林进行相似度计算，记作 s_1 或 s_2，此时 λ_1 和 λ_2 一个为 1，另一个为 0；

③ 当 $w_1\in A,w_2\in B$ 时，基于词林中查找 w_2 的同义词集合，再依次与 w_1 基于知网计算相似度，取最大值，记为 s_1；若 w_2 在词林中无同义词，则取 $s_1=0.2$，此时 $\lambda_1=1,\lambda_2=0$；

④ 当 $w_1\in A,w_2\in C$ 时，首先基于知网计算 w_1 与 w_2 的相似度，记为 s_1，其次在词林中找 w_2 的同义词集合，并依此与 w_1 基于知网计算相似度，取最大值，记为 s_2；若 w_2 在词林中无同义词，则取 $s_2=s_1$，此时 $\lambda_1>\lambda_2$。本文实验取 $\lambda_1=0.6,\lambda_2=0.4$；

⑤ 当 $w_1\in B,w_2\in C$ 时，首先基于词林计算

w_1 与 w_2 的相似度，记为 s_2，其次在词林中找 w_1 的同义词集合，并依此与 w_2 基于知网计算相似度，取最大值，记为 s_1；若 w_1 在词林中无同义词，则取 $s_1=s_2$，此时 $\lambda_2>\lambda_1$。本文实验取 $\lambda_1=0.4$，$\lambda_2=0.6$。

其中，基于知网的词语相似度的计算[20]如式(7)、式(8)所示。

$$\mathrm{sim}(C_1,C_2)=\sum_{i=1}^{3}\beta_i\prod_{j=1}^{i}\mathrm{sim}_j(C_1,C_2) \tag{7}$$

$$\mathrm{sim}(W_1,W_2)=\max_{i=1\ldots m,j=1\ldots n}\{\mathrm{sim}(C_{1i},C_{2j})\} \tag{8}$$

式(7)中，$\mathrm{sim}_1(C_1,C_2)$是独立义原构成的集合所计算的相似度；$\mathrm{sim}_2(C_1,C_2)$是关系义原特征结构的相似度；$\mathrm{sim}_3(C_1,C_2)$是关系符号特征结构的相似度；参数 $\beta_i(1\leqslant i\leqslant 3)$是可调节的，且满足 $\beta_1+\beta_2+\beta_3=1$，经过实验，本文算法中 β_1、β_2、β_3 分别取 0.7、0.17 以及 0.13。式(7)得到的是义项的相似度，当词语中有多个义项时，就用式(8)计算所有义项组合中相似度最大的，即为两词的相似度。其中 m 为词 W_1 的义项个数，n 为词 W_2 的义项个数。

基于词林的词语相似度的计算[20]如式(9)所示。

$$\mathrm{sim}(C_1,C_2)=(1.05-0.5\mathrm{dis}(C_1,C_2))\sqrt{e^{\frac{-k}{2n}}} \tag{9}$$

其中，$\mathrm{dis}(C_1,C_2)$是词语编码 C_1、C_2 在树状结构中的距离函数；n 为分支层节点总数，表示两个词语最近公共父节点的直接子节点个数；k 表示最近公共父节点中两个词语所在分支的间隔距离。同理，当一个词对应多个编码时，用式(8)计算词语的相似度。

(3) 定义粗糙推理中关联规则的关联度[21]如式(10)所示。

$$\mathrm{PMI}(A,B)=p(A,B)/p(A)p(B) \tag{10}$$

其中，A 与 B 是文档中的两个候选关键词；$p(A,B)$表示 A 和 B 在同一句话中出现的概率；$p(A)$表示 A 出现的概率；$p(B)$表示 B 出现的概率。PMI 值越大相关性越强。

根据此关联度确定存在直接关联的候选关键词，即当 $\mathrm{PMI}(w_1,w_2)\neq 0$ 时，则 w_1，w_2 之间存在直接的关联，并将词 w_1，w_2 及其关联度存入关联集合中，同时根据此关联度可建立进行粗糙推理的推理关系 S；

其次，运用粗糙数据推理的规则，得到所有不同等价类中其余候选关键词间的相关性，并将这些词及其关联度也存入关联集合中。

(4) 根据步骤(3)中得到的关联集合去构建带权的候选关键词图；然后根据式(1)迭代计算各候选关键词的权重至收敛。改进 TextRank 算法的流程图如图 3 所示。

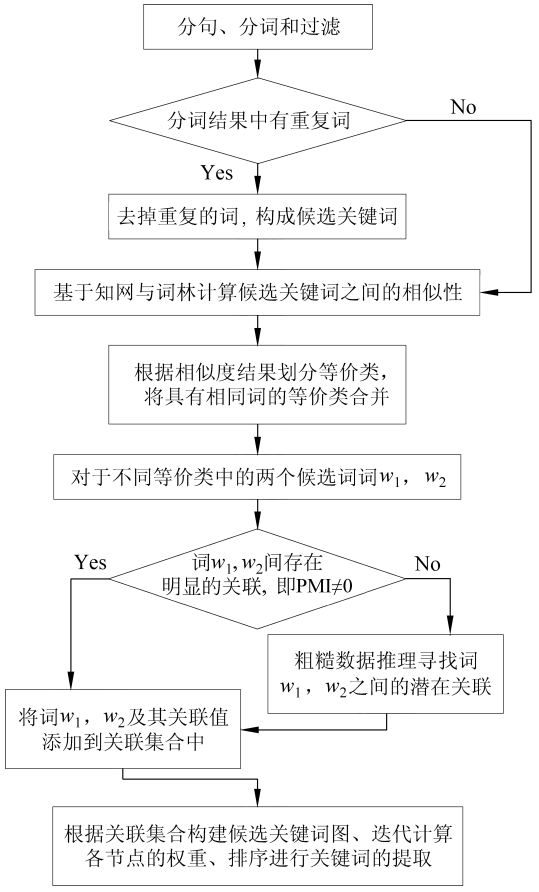

图 3 改进 TextRank 算法的流程图

2 实验结果与分析

2.1 实验数据及评价标准

实验选取来自搜狗实验室的全网新闻数据共 1.4 GB 作为测试文本的抽取依据。该数据集包含了 2012 年 6 月至 7 月期间，有关国内外社、体、文、娱等各个领域的新闻数据。随机抽取各领域的 1 050篇文本组成测试集合，测试算法效果。同时邀请了多名具有本科及以上学历的老师与学生，他们均为我校新闻系与中文系的师生，采用人工交叉标注的方式，为每篇文本提取 10 个关键词，并按其重要性排序给出。

除此之外，基于相同的测试集，对经典的 TF-IDF 算法、TextRank 算法以及本文算法的实验结果进

行对比。文中采用信息检索与分类领域常用的三个评测指标对比评价实验结果的质量，其中包含准确率(P)，其表示提取结果的准确度；召回率(R)，其表示提取结果对正确关键词的覆盖程度；以及 F 值，它是 P 值与 R 值调和平均的综合评价指标。三个指标的具体计算公式[22]如式(11)～式(13)所示。

$$P=\frac{\text{人工标注集合}\cap\text{提取集合}}{\text{提取集合}}\times 100\% \tag{11}$$

$$R=\frac{\text{人工标注集合}\cap\text{提取集合}}{\text{人工标注集合}}\times 100\% \tag{12}$$

$$F=\frac{2\times P\times R}{P+R} \tag{13}$$

实验环境的操作系统为 Windows 7(64 位)，本文提出的算法采用 Python 语言实现，分词和词性标注使用的是 Jieba 开源工具。同时，采用 Python 编程对实验所涉及到的其余两种对比算法进行复现。

2.2　实验结果

实验中发现有两个重要参数的取值会影响 TextRank 算法的关键词提取结果。共现窗口大小 ω、关键词个数 k，而基于统计特征的 TF-IDF 算法与本文算法的实现不受参数 ω 的影响。对 ω 这一参数的确定，我们设置提取关键词个数 $k=10$，对比窗口取值在[4,10]内，该算法提取结果的 F 值，对比结果如图 4 所示。

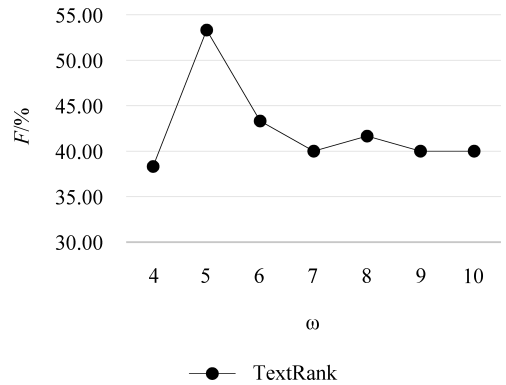

图 4　不同 ω 值下提取结果 F 值

由图 4 可以看出，在不同的 ω 取值下 TextRank 算法的提取效果不同。本文在相同的测试集下对 ω 的取值效果进行对比，发现当 $\omega=5$ 时原始 TextRank 算法的提取效果最好，因此为保证本文算法的有效性，将对比实验中 TextRank 算法的 ω 初始值设为 5。

设置初始窗口值 $\omega=5$，计算关键词个数取值在[3,10]内三种算法的准确率、召回率以及 F 值，计算结果如表 1 所示。

表 1　三种算法的实验结果对比

关键词个数	算法	P/%	R/%	F/%
3	TF-IDF	51.33	15.40	23.69
	TextRank	74.00	22.20	34.15
	本文算法	78.66	23.60	36.31
4	TF-IDF	49.00	19.60	28.00
	TextRank	70.00	28.00	40.00
	本文算法	76.00	30.40	43.43
5	TF-IDF	48.40	24.20	32.27
	TextRank	66.80	33.40	44.53
	本文算法	71.60	35.80	47.73
6	TF-IDF	42.67	25.60	32.00
	TextRank	60.33	36.20	45.25
	本文算法	69.00	41.40	51.75
7	TF-IDF	41.14	28.80	33.88
	TextRank	55.43	38.80	45.65
	本文算法	66.29	46.40	54.59
8	TF-IDF	38.50	30.80	34.22
	TextRank	52.00	41.60	46.22
	本文算法	61.50	49.20	54.67
9	TF-IDF	35.78	32.20	33.90
	TextRank	47.56	42.80	45.05
	本文算法	58.67	52.80	55.58
10	TF-IDF	33.80	33.80	33.80
	TextRank	44.40	44.40	44.40
	本文算法	55.40	55.40	55.40

同时为方便观察三种算法实验结果的变化，对算法的 P 值、R 值以及 F 值这三项指标做图，如图 5～图 7 所示。

图 5 描述的是在提取不同个数的关键词时三种算法准确率的变化趋势。从图中可以看出，随着关键词提取个数的增加，各算法的准确率均有所下降，但本文算法的准确率始终高于其余两种算法。因为本文算法所采用的粗糙数据推理规则，会将近似信息融入到数据推理的过程中，使数据之间的相互推理呈现出近似蕴含或非精确关联的特性，可以挖掘候选关键词之间的潜在关联。若将潜在的关联添加到各候选关键词权重的迭代计算中，则会得到更加准确的提取结果。因此相较于依据固定关联规则计算出词间关联算法，或是依赖统计词频的算法，本文

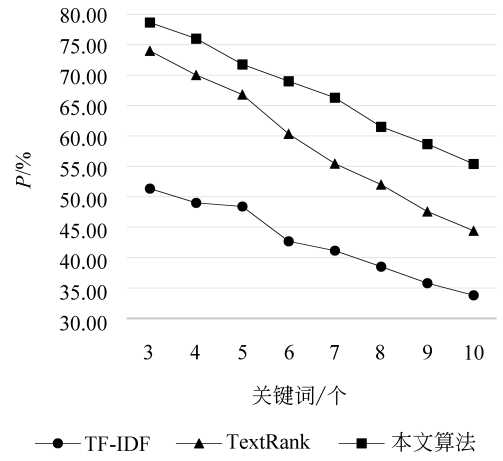

图5 三种算法 P 值对比图

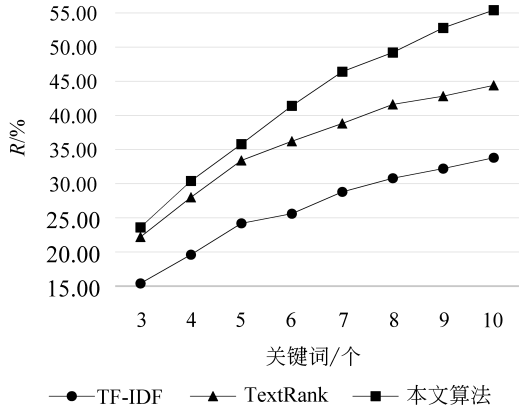

图6 三种算法 R 值对比图

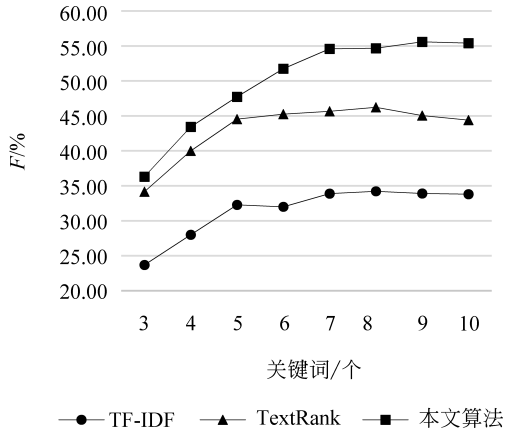

图7 三种算法 F 值对比图

算法的准确率在理论上都会比较高一些，其准确率 P 值会高于其余两种算法。

图6描述了在提取不同个数的关键词时，三种算法召回率的变化情况。图中本文算法的召回率要高于其余两种算法，同时随着关键词个数的增加，算法召回率的相对增长率也增加。这是因为TF-IDF算法过于依赖词频，完全没有用到词间的关联性；而经典的TextRank算法采用共现窗口原则，虽考虑了词间的关系，但由于其具有局限性，使得算法更倾向于将频繁词提出，这样可能会忽略词频较低但可描述文本主题的重要词汇。而本文算法所采用的粗糙数据推理可扩大关联范围，增加关联数据，能够增强提取结果对正确关键词的覆盖程度，提高算法召回率。特别是随着关键词个数的增加，词频的影响力降低，本文算法召回率增加的优势会更加明显。

图7描述的是在提取不同个数的关键词时，三种算法 F 值的取值情况。当对实验结果评价时，希望 P 值与 R 值同时都越高越好，但事实上在大部分情况下两者是矛盾的，因此要采用 F 值来对两者进行综合考虑，F 值可以反映整个算法的有效性。理论上，基于粗糙数据推理的关键词提取，由于其可以挖掘候选关键词之间的潜在关联，增加关联的候选词与范围，将潜在的关联添加到各候选关键词权重的迭代计算当中，提取结果会更加的准确，即算法也更加有效。从公式上看，根据实验结果，本文算法较其余两种算法都有较高的 P 值与 R 值，P 值与 R 值越高，则 F 值也会越高，其中较高的 F 值则可以说明算法的有效性。由图8可以看出，随着关键词个数的增加，本文算法对于其余两种算法 F 值的增长值也越高，即具有更加明显的提升效果。

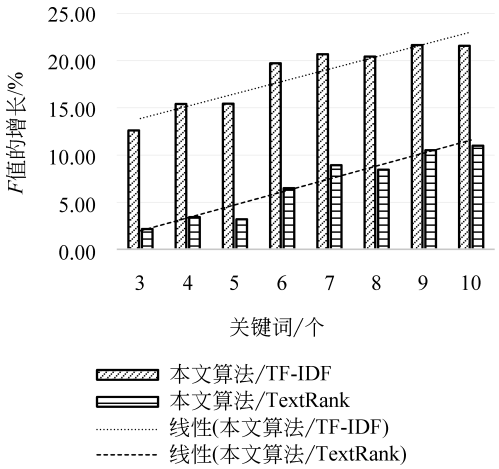

图8 本文算法对于其余两算法 F 值的增长

综上分析可知，本文算法的准确率(P)、召回率(R)以及综合评价指标 F 值均高于TF-IDF和TextRank算法，说明基于粗糙数据推理的TextRank算法提取效果更加有效。基于统计特征的TF-IDF算法与经典的TextRank算法从本质上都更依赖于词频，可能会优先将频繁出现的词汇提

出，但对于一篇文档尤其是中文文本而言，主题词可能并不会一直出现。因此基于粗糙数据推理的 TextRank 算法从文本整体出发，扩大关联的范围，增加关联的数据，通过粗糙数据推理建立词汇间的关联，可进一步提升算法的准确性。

本文算法在进行关键词的提取时，由于关联范围的扩大，需要进行关联推理的数据也随之增加，同时还要对这些数据进行语义计算，划分等价类，因此在算法取得较高准确率的同时，也增加了相应的计算。以下将从算法的运行时间以及物理内存占用两个方面对算法的效率进行分析。

实验表明，虽然 TF-IDF 算法的运行时间与物理内存占用较少，但其准确率 P、召回率 R 以及 F 值都远远低于其余两种方法，因此此处将对本文算法以及传统的 TextRank 算法的效率进行比较。

本文测试集采用的是搜狗实验室发布的全网新闻数据，其中单篇文本长度在 2 000 以上的较少，文本规模主要集中在 1 000 字以内。本文以文本规模（文本字数）划分测试集合，每个测试集合随机选取相应文本规模的 20 篇文章，对比两种算法的运行时间以及物理内存占用，对比结果如表 2、表 3 所示。

表 2　关键词提取算法运行时间对比

文本字数/字	本文算法/s	TextRank/s
300 以内	1.159 567	1.180 068
300～600	1.559 756	1.372 746
600～900	2.640 152	1.449 583
900～1 200	3.813 218	2.032 611
1 200 以上	6.630 551	2.185 125

从表 2 可以看出，当文本的字数在 300 字之内时，推理和语义计算的次数较少，则算法的运行时间也会较短，同时运行时间还低于传统 TextRank 算法的时间；随着文本规模的增大，推理与语义计算次数增多，本文算法相较于 TextRank 算法时间虽有所增加，但若文本字数少于 1 200 字时，其增加范围也在 1～2 s 之内，与 TextRank 效率相近。

表 3　关键词提取算法物理内存占用对比

文本字数/字	本文算法/MB	TextRank/MB
300 以内	67.123 1	68.119 2
300～600	67.753 9	68.184 9
600～900	67.896 1	68.248 05
900～1 200	67.994 3	68.335 9
1 200 以上	68.652 3	68.421 9

从表 3 可以看出，本文算法的物理内存占用大小与传统 TextRank 算法是相近的，且当文本的字数小于 1 200 字时，本文算法所占的内存空间要更小，效率更高。

3　结束语

通过对文本关键词提取问题的研究发现，词义与词汇间潜在的关联对关键词的提取结果有直接的影响。基于此出发点，本文提出一种基于粗糙数据推理的 TextRank 关键词提取算法。使用知网与词林计算词语相似度，并对候选关键词进行划分，再使用粗糙数据推理规则，发掘候选关键词之间潜在的关联关系。实验结果表明，基于粗糙数据推理的改进 TextRank 算法，将候选词之间的潜在关联考虑在内，进一步提高了关键词提取的准确性。下一步我们将对粗糙数据推理规则进一步细化与完善，从而得到更好的提取效果。

参考文献

[1] Turney P D. Learning algorithms for keyphrase extraction[J]. Information Retrieval, 2000, 2(4): 303-336.

[2] Florescu C, Caragea C. A position-biased PageRank algorithm for keyphrase extraction[C]//Proceedings of the 31st American Association for Artificial Intelligence, 2017.

[3] Jones K S. A statistical interpretation of term specificity and its application in retrieval[J]. Journal of Documentation, 2004, 28(1): 493-502.

[4] Salton G, Buckley C. Term-weighting approaches in automatic text retrieval[J]. Information Processing & Management, 1988, 24(5): 513-523.

[5] Blei D M, Ng A Y, Jordan M I. Latent Dirichlet allocation[J]. Journal of Machine Learning Research, 2003, 3(1): 993-1022.

[6] Mihalcea R, Tarau P. TextRank: Bringing order into text[C]//Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, 2004: 404-411.

[7] 李鹏，王斌，石志伟，等. Tag-TextRank：一种基于 Tag 的网页关键词抽取方法[J]. 计算机研究与发展，2012，49(11)：2344-2351.

[8] Yijun G, Tian X. Study on keyword extraction with LDA and TextRank combination[J]. Data Analysis and Knowledge Discovery, 2014, 30(7): 41-47.

[9] Hong Zhang, Yuan Liu. Improved TextRank algorithm research based on topic weighting[C]//Proceedings of the 2015 International Conference on Industrial Informatics, Machinery and Materials, 2015.

[10] Wan X. Timed TextRank: Adding the temporal dimension to multi-document summarization[C]//Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2007: 867-868.

[11] 肖辛格. 基于基本层次范畴改进 TextRank 算法的中文关键词抽取[D]. 武汉：华中师范大学硕士学位论文, 2017.

[12] Zuo X, Zhang S, Xia J. The enhancement of TextRank algorithm by using Word2Vec and its application on topic extraction[C]//Journal of Physics: Conference Series. IOP Publishing, 2017, 887(1): 012-028.

[13] 刘竹辰, 陈浩, 于艳华, 等. 词位置分布加权 TextRank 的关键词提取[J]. 数据分析与知识发现, 2018, 2(9): 74-79.

[14] 柳林青，余瀚，费宁，等.一种基于 TextRank 的单文本关键字提取算法[J].计算机应用研究，2018，35(03)：705-710.

[15] 徐馨韬，柴小丽，谢彬，等.基于改进 TextRank 算法的中文文本摘要提取[J].计算机工程，2019，45(03)：273-277.

[16] Yan S, Yan L, Wu J. Rough data-deduction based on the upper approximation[J]. Information Sciences, 2016, 373: 308-320.

[17] 周宁，朱昭昭.基于粗糙数据推理的 Criminisi 图像修复算法[J].激光与光电子学进展，2019，56(02)：84-91.

[18] 王汉萍. 粗糙集理论在文本挖掘的分类算法中的应用研究[D].青岛：中国海洋大学硕士学位论文，2003.

[19] Pawlak Z. Rough sets: Theoretical aspects of reasoning about data[M]. Kluwer Academic Publishers, 2012.

[20] 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.

[21] Turney P D, Littman M L. Unsupervised learning of semantic orientation from a hundred-billion-word corpus[J]. Artificial Intelligence, 2002.

周宁(1979—)，通信作者，博士，副教授，主要研究领域为自动推理技术、形式化方法。
E-mail: zhouning@mail.lzjtu.cn

朱昭昭(1995—)，硕士研究生，主要研究领域为自动推理技术、图像处理。
E-mail：zzz123567@qq.com

石雯茜(1992—)，硕士研究生，主要研究领域为自动推理技术、自然语言处理。
E-mail：0618591@stu.lzjtu.edu.cn