基于图排序算法的自动文摘研究综述

王俊丽 魏绍臣 管 敏
(同济大学电子与信息工程学院CAD中心 上海 201804)

摘 要 互联网技术的快速发展使得信息的采集和传播速度达到了空前的水平,海量的数据使得人们获取有价值的信息越发困难。自动文摘技术可以从海量的信息中提取出能代表原文重要内容且简洁精练的一段文字,高度压缩文档是解决信息超载问题的有效方法,因此自动文摘技术的研究引起人们越来越多的关注。目前诸如统计分析、机器学习技术以及语言学知识等在已有的自动文摘系统中都有所应用。对基于图排序算法的自动文摘的研究成果进行综述,首先阐述自动文摘以及图排序算法的基本知识,然后重点从图的构建、图排序、句子选择3个方面系统地介绍基于图排序算法的自动文摘的研究现状,最后在分析已有自动文摘系统的基础上,探讨了基于图排序算法的自动文摘的未来发展方向。

关键词 自动文摘,图排序,图模型,文本挖掘
中图法分类号 TP391.1 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.12.001

Survey on Graph Model-based Document Summarization

WANG Jun-li WEI Shao-chen GUAN Min
(CAD Center, School of Electronics and Information Engineering, Tongji University, Shanghai 201804, China)

Abstract With the rapid development of the Internet technologies, the speed of information transmission has reached unprecedentedly high levels. However, getting valuable information from mass data is becoming more and more difficult. Automatic summarization technologies allow us to extract a summary that represents the main idea of the original document, which have attracted much attention. Now many related technologies have been widely used in existing automatic summarization approaches, such as statistical analysis, machine learning technology, linguistic knowledge and etc. This paper summarized research works of summarization approaches based on graph-based ranking algorithms. First the basic knowledge of automatic summarization and graph-based ranking algorithms were elaborated. Then the summarization approaches based on graph ranking algorithms were introduced, mainly including three parts: construction of text graph, graph-based ranking, and sentences selection. Finally on the basis of analysis of existing approaches, the future development of graph-based summarization approaches was explored.

Keywords Automatic text summarization, Graph-based ranking, Graph model, Text mining

1 引言

互联网技术的快速发展使得信息急速地传播和分享,另一方面,海量的数据资源也使得获取有价值的信息愈发困难。自动文摘技术利用计算机对文档进行自动处理,生成包含原文档核心内容的摘要,实现对文档的压缩。自动文摘技术可以帮助人们快速高效地获取主要信息,节省用户的时间和资源,是解决信息超载问题的有效方法。

自动文摘技术于1958年[1]首次被提出,最初只应用在科技文献摘要等有限的领域。但近十几年来,由于互联网技术的飞速发展以及信息爆炸等问题的出现,人们对自动文摘技术的需求日益迫切,自动文摘因此成为研究热点并获得巨大发展,诸多新技术都成功运用在了这一领域。

根据处理媒介、输入文档、输出内容、文摘目的和处理语言的不同,自动文摘技术有多种分类方法。从研究者的角度,本文倾向于将自动文摘技术分为基于抽取的自动文摘和基于抽象的自动文摘。基于抽取的自动文摘通过对一篇或多篇文档中句子各种特征的计算,得出句子的重要程度,再通过适当的选择方法选出摘要句子形成文章摘要。基于抽象的自动文摘先识别出文本的主要内容,然后通过自然语言生成技术重新组织语言或添加新的文本单元生成文章摘要。由于自然语言生成技术的局限性,基于抽象的自动文摘相对困难,目前相关的研究与应用都很有限,大部分自动文摘系统都是基于抽取的。基于图排序算法的自动文摘是受PageRank和HITS

到稿日期:2014-12-08 返修日期:2015-03-27 本文受国家自然科学基金(61105047),港澳台科技合作项目(2013DFM10100),上海市科委项目(14JC1405800),国家科技支撑计划(2012BAF12B11)资助。

王俊丽(1978—),女,博士,副研究员,主要研究方向为互联网数据分析、自动文摘,E-mail:junliwang@tongji.edu.cn;魏绍臣(1990—),男,硕士生,主要研究方向为自然语言处理、自动文摘,E-mail:changevlxl@163.com;管 敏(1992—),女,硕士生,主要研究方向为互联网数据分析、隐私保护,E-mail:gm3416@163.com。

等算法的启发而来，也属于基于抽取的自动文摘方法，其主要思想是在构建的文本图上利用图排序算法得出句子或单词在文档中的权重。

图排序算法（例如 Kleinberg 的 HITS 算法[2]和 Google 的 PageRank 算法[3]）已经成功应用在了超链分析、引文分析、社交网络等领域。图排序算法的优点是在计算节点权重的过程中可以结合图的全局信息做出判断，而不仅仅是依赖某几个节点有限的信息。对于自然语言处理领域的文本图，图排序算法同样适用。基于图排序算法的自动文摘就是将这种考虑全局信息的排序方法应用在自动文摘系统中，其最大的优点是在计算句子权重的步骤中可以充分考虑词汇之间、词组之间或句子之间的全局关系。

本文第 2 节将从统计分析、主题发掘、篇章关系、机器学习和图模型等几个方面介绍基于抽取的自动文摘技术；第 3 节将介绍图排序算法的基本知识，并从图构建、图排序、句子选择 3 个方面重点介绍基于图排序算法的自动文摘的研究现状；第 4 节将讨论基于图排序的自动文摘的未来研究方向。

2 基于抽取的自动文摘技术

基于抽取的自动文摘的核心思想是：首先对一篇或者多篇文档中句子的各种特征进行统计分析，计算句子的重要程度，再通过适当的摘要方法选出摘要句子，形成文章摘要。其主要的处理步骤为：(1)计算句子的重要性；(2)以某种规则为句子排序；(3)选取句子形成摘要。上述每个步骤都是一系列技术的集合，为了阐明基于抽取的自动文摘目前所应用的技术，我们将其分为 5 个类别：基于统计的技术、基于主题的技术、基于篇章关系的技术、机器学习技术、基于图模型的技术。

2.1 基于统计的技术

统计模型在自然语言处理领域应用广泛，统计技术也是自动文摘最早应用的技术[1]。相对于其他技术，统计技术不需要复杂的建模，简单且易于实现。文献[1]使用词频来衡量句子在一篇文档中的重要性，其思想是频繁出现的单词最能代表文章主题（但是并不是所有频繁出现的单词都有代表性，类似“a”、“the”、“一个”这样的停用词不携带任何语义信息，不能用于权重计算）。TF-IDF 是目前应用最广泛的一种词频统计技术。TF-IDF 基本思想是单词的重要性与它在文档中出现的频率成正比，但与它在整个语料库中出现的频率成反比。通常来说，统计特征会与句子位置、句子长度、句子与文档题目的相似度等其他特征结合使用，以加强对文本单元权重衡量的精确性[4,5]。例如在新闻报道中，首段内容通常包含所报道事件的主要信息，而其他部分则是说明事件发生的细节和背景信息，将新闻报道首段中的句子赋予更高的权重是一个合理的策略。文献[6,7]表明仅靠统计技术和文本单元的基本特征（忽略其他更深层的知识、主题特征、话语关系等）也可以生成高质量的文摘。

2.2 基于主题的方法

文本单元的重要性不只决定于字面上的词语重复，还取决于文字背后的语义关联。基于主题的方法就是挖掘语义关联，将主题知识融入对文本单元权重的计算。文献[8]根据提示词识别生成摘要，这种技术根据句子所包含的特定短语或单词计算句子权重。例如，包含“in conclusion”或者“the aim of this paper”这种总结性短语的句子很可能代表了文章主题，应该获得高权重。此外，还有很多自动文摘方法利用了主题识别和主题分割，例如文献[9]利用聚类生成描述特定主题的模板；文献[10]提出的方法结合了局部主题识别，通过聚类识别子主题，再从每个子主题中根据统计分析抽取权重最高的句子作为摘要。

2.3 基于篇章关系的方法

除了以上提到的方法，从语言学的角度同样可以解决自动文摘问题。篇章关系（Discourse Relation）分析在自动文摘领域也有广泛应用。文献[11]提出的文摘方法以修辞结构理论（Rhetorical Structure Theory）[12]作为基础，进而扩展了修辞结构，通过 nucleus-satellite（nucleus 包含文档的基础信息，satellites 包含 nucleus 的额外补充信息）类型的篇章关系计算文本单元的重要性。文献[13]将 RST 理论与其他自动文摘方法结合，尽管实验结果证明这种融入语言学知识的混合方法对文摘质量并没有提升，但该文献指出这是因为该方法所依赖的语法分析器无法识别文档中所有的 RST 关系，如果有合适的分析器，语言学知识的融入必定可以提升文摘质量。文献[14]结合了统计分析和语言学知识，实验表明这种组合方法优于使用单一类型的技术。在自动文摘过程中，如果不能识别出文档中的全部实体，会导致指代垂悬（Dangling Anaphora）现象。一些方法[15,16]利用指代消解（Anaphora Resolution）技术解决这个问题，即首先对文本进行预处理，然后利用指代消解系统将所有代词替换成相应的实体，最后再进行摘要。

2.4 机器学习方法

二分分类器[17]、隐马尔科夫模型[18,19]和贝叶斯方法[20]是最早应用在自动文摘领域的机器学习方法。此外，其他机器学习方法在自动文摘领域的应用也很广泛。文献[21]提出了一种基于神经网络的单文档自动文摘系统 NetSum，该系统使用 RankNet 算法[22]计算句子权重。除了利用关键词词频以及句子位置等统计特征外，该系统还融入了维基百科词条以及维基百科用户的搜索记录。文献[23]提出了一种基于查询的多文档自动文摘系统 FastSum，该系统使用最小角回归（Least Angle Regression）选择关键特征，并使用支持向量回归（Support Vector Regression）对句子权重进行排序。文献[24]也使用了支持向量回归技术，并结合单词和短语词频、句子位置等统计特征以及基于语义和命名实体等其他特征共同训练分类器。使用机器学习方法进行自动文摘的优点是可以方便地测试各种特征组合的有效性，并且针对特定的特征，可以测试不同的机器学习算法并选择最优的算法；缺点是机器学习算法需要大规模语料库的支持。

2.5 基于图模型的方法

图模型算法在自动文摘领域有广泛应用，最早的工作可见文献[25]。文献[26]提出了一种基于亲和图的自动文摘方法，该方法通过考虑句子间的相似性结合主题信息抽取出高信息性和高独特性的句子，经过冗余削减后生成文摘。文献[27]利用 N-Gram 图抽取文档中的重要成分。文献[28]使用 WordNet 和 is-a 关系识别文档中的概念来构建文本图，这种方法在新闻和生物信息领域应用广泛。

基于图排序算法的自动文摘是图模型在自动文摘领域应用的一类特例，其因为良好的效果以及可扩展性成为了本领域的研究热点。接下来本文重点介绍基于图排序算法的自动

文摘的研究现状。

3 基于图排序算法的自动文摘

受图排序算法在各领域成功应用的启发，文献[29-31]首次提出了基于图排序算法的自动文摘方法，其主要思想是将文本单元(句子、词汇等)作为图的节点，将文本单元之间的关联(余弦相似度、同义性等)作为边，将文档表示成一个文本图；然后基于图排序算法计算节点的权值；再根据图排序结果采用某种策略选择文摘句子从而生成摘要。其本质是图的拓扑结构隐含着每个文本单元的重要性。

3.1 图排序算法

图排序算法是一种基于图结构来计算图中节点重要性的算法。本小节介绍两种主要的图排序算法：PageRank 和 HITS。

3.1.1 PageRank

PageRank[3]算法是链接分析的经典算法，其基本思想是同时考虑反向链接和常规链接对网页权重的影响：一个网页的反向链接越多，发出反向链接的网页权重越大，这个网页权重也越大。PageRank 算法利用网页之间的链接结构来构建一个以网页为节点、常规链接和反向链接为边的有向图。将其抽象为数学概念，就是构建一个具有本原转移概率矩阵的马尔科夫链，初始状态为赋予节点随机权重得到的向量；转移矩阵的不可约的特性保证了经过一定次数的转移之后的马尔科夫链存在一个平稳分布(也就得到了每个节点的最终权重)。

$$PR(p_i)=\frac{1-d}{N}+d\sum_{p_j\in M(p_i)}\frac{PR(p_j)}{L(p_j)} \tag{1}$$

式(1)为 PageRank 的迭代公式，$p_1,p_2,\cdots,p_N$ 代表所有节点，$M(p_i)$为链向 p_i 的节点集合，$L(p_j)$是节点 p_j 的出度，N 为图中节点总数，d 为阻尼系数，一般设置为 0.85。其思想是一个网页对另一个网页权重的贡献由其出度均分。由上述公式得出的 $PR(p_i)$之和为 1，于是上述公式可以写为以矩阵表示的形式，如式(2)所示。

$$\boldsymbol{R}=(\frac{1-d}{N}\boldsymbol{E}+d\boldsymbol{M})\boldsymbol{R} \tag{2}$$

式中，$\boldsymbol{R}$ 为 $PR(p_i)$组成的向量，即节点最终的权重分布向量，由于 $PR(p_i)$之和为 1，$\boldsymbol{R}$ 可以看作是一个概率分布，于是$|\boldsymbol{R}|=1$，又有 $\boldsymbol{ER}=1$，$\boldsymbol{E}$ 为元素全为 1 的矩阵，矩阵 $\boldsymbol{M}=(\boldsymbol{K}^{-1}\boldsymbol{A})^{\mathrm{T}}$，$\boldsymbol{A}$ 为图的邻接矩阵，$\boldsymbol{K}$ 为对角矩阵，其元素为每个节点的出度，这样得到的矩阵 $\boldsymbol{M}$ 的元素为

$$\boldsymbol{M}_{ij}=\begin{cases}1/L(p_j), & \text{如果 } j \text{ 链向 } i\\ 0, & \text{其他}\end{cases}$$

这种表示方法可以看出 PageRank 公式转化为了一个转移矩阵为$(\frac{1-d}{N}\boldsymbol{E}+d\boldsymbol{M})$的马尔科夫链。

3.1.2 HITS

HITS(Hyperlinked Induced Topic Search)[2]最初是一种网页排序算法。HITS 算法赋予网页 2 个特性：权威性(Authority，由外部链接来衡量)和中心性或者叫枢纽性(Hub，由链出的链接来衡量)。HITS 算法有 2 个基本假设：一个好的“权威”页面会被很多好的“枢纽”页面指向；一个好的“枢纽”页面会指向很多好的“权威”页面。HITS 算法的核心思想就是利用这两个基本假设，根据相互增强关系进行多轮迭代计算以更新页面的两个权值，直到权值稳定不再发生明显的变化为止。对于图中每个节点，HITS 算法最终会生成 2 个得分：权威性得分和中心性得分。

对于所有节点 V_i，根据式(3)更新其权威性。

$$auth(V_i)=\sum_{V_j\in In(V_i)}hub(V_j) \tag{3}$$

设 $\boldsymbol{A}$ 为图的邻接矩阵，上述公式可以用矩阵表示：

$$\boldsymbol{X}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{Y}$$

对于所有节点 V_i，根据式(4)更新其枢纽性。

$$hub(V_i)=\sum_{V_j\in Out(V_i)}auth(V_j) \tag{4}$$

用矩阵表示式(4)为：

$$\boldsymbol{Y}=\boldsymbol{AX}$$

在更新权威性和枢纽性的过程中，每个节点的权威性-枢纽性得分都会越来越大，变得发散，所以在每次迭代后，要对权威性和枢纽性分别进行标准化，标准化公式如式(5)所示。

$$auth(V_i)=\frac{auth(V_i)}{\sum_{i=1}^{N}(auth(V_i))^2} \tag{5}$$

式中，N 为图中的总节点数。枢纽性的标准化公式与权威性的标准化公式同理，标准化的结果使得所有节点的 $hub(V_i)$的平方和与 $auth(V_i)$的平方和都为 1。

3.2 基于图排序算法的自动文摘研究现状

图排序算法最初是为分析网页权重而被提出，以网页作为节点，根据网页之间的链接关系构建图的边，而超链接又分为反向链接和常规链接，所以生成的是一个无加权的有向图。但是在自动文摘领域，图是根据自然语言文本来构建的，节点为文档的文本单元，而边则是根据节点之间的相关性来构建的，这种文本单元的相关性一般是相互的，而且其强度通常也是有意义的，所以基于图排序算法的自动文摘一般构建的是加权或无加权的无向图。

对于无向图和加权图，图排序算法同样适用。在无向图中节点的入度和出度相等，而且在稀疏图中，无向图通常收敛得更快。随着图连通性的增强，图排序算法通常会在较少次数的迭代之后收敛，而且在连通性较强的图中，无向图和有向图的收敛曲线几乎重叠[32]。

对于加权图，可以将节点之间的权重 ω_{ij} 或 ω_{ji} 引入图排序算法中。式(6)－式(8)是以上两种图排序算法中引入权重后的公式。

$$PR(p_i)=\frac{1-d}{N}+d\sum_{p_j\in M(p_i)}\omega_{ji}\frac{PR(p_i)}{L(p_j)} \tag{6}$$

$$auth(V_i)=\sum_{V_j\in In(V_i)}\omega_{ji}hub(V_j) \tag{7}$$

$$hub(V_i)=\sum_{V_j\in Out(V_i)}\omega_{ij}auth(V_j) \tag{8}$$

相关实验证明[32]，在图排序过程中，加权图与无加权图的迭代次数和收敛曲线几乎相同。

基于图排序的自动文摘方法包含预处理、构建文本图、图排序和句子选择等一系列步骤，每个步骤都是一系列技术的集合。本文将基于构建图、图排序、句子选择这 3 个步骤，详细说明目前基于图排序算法的自动文摘技术的研究现状和一些典型的工作。

3.2.1 图的构建

图的构建是基于图排序算法的自动文摘的第一步，在这个步骤中要选择合适的文本单元作为图的节点，并利用文本单元之间的相关性生成边。边可以是有向或者无向、加权或

者非加权的。根据使用的文本单元的不同，主要有以下 3 种构建方法：句子作为节点、单词作为节点和混合成分作为节点。

(1)句子作为节点。通常来说，自动文摘算法会直接选择句子[29,30,32,33]作为节点，文献[29,30]提出的 LexRank 算法使用句子作为图节点，根据句子间余弦相似度构建边，若相似度大于阈值，就在两个节点间添加一条边；否则就没有边连接，从而生成无向无加权图。文献[32]提出的 TextRank 算法在 LexRank 的基础上做了改进，利用余弦相似度对边赋予权值，生成无向加权图。这两种构建图方法比较简单，虽然应用广泛，但是仅用余弦相似度衡量节点间的关联度忽略了其他可能有效的因素，例如主题信息或语言学知识等。

针对上述方法的不足，融入多种知识扩展边的权值是十分有代表性的改进方法。文献[33]中，边的构建综合考虑 4 个因素：余弦相似性、语义相似性、指代关系、篇章信息，通过结合这 4 种相关性来决定两个节点之间是否生成一条边。类似地，文献[34]提出了一种包含词汇、句法、语义的 3 层架构来衡量句子间的相关性。

冗余是自动文摘领域一直存在的问题之一，尤其是在多文档文摘中，多篇文档中的相似成分会加剧冗余。聚类可以将相似内容聚集在一起，对聚类簇进行统一处理不仅可以减少冗余，也融入了句子间的群组关系。文献[47]提出利用超图的概念来构建图(超图是一条边可以连接多个节点的图)。该算法首先使用 DBSCAN(Density Based Spatial Clustering of Applications with Noise)[35]算法对句子进行聚类(DBSCAN 算法的优点在于聚类的同时可以剔除一些不满足最小数量阈值的噪声簇)，然后根据两条规则生成文本图：首先，同一个聚类簇中的所有句子被一条超边所连接，边的权值为聚类簇与整篇文档的余弦相似度；其次，任意两个句子被一条超边所连接，超边权值为两个句子的余弦相似度。

也有一些自动文摘方法根据其他合理的假设进行文本图的构建。文献[36]指出一个句子的重要性应从信息代表性和独特性这两个方面来衡量，该方法根据句子之间的差异性构建边，生成加权无向图。句子间的差异性根据句子的正弦相似度以及 WordNet 来衡量，如式(9)所示。

$$DS(S_i,S_j)=\sin(S_i,S_j)*\sum_{k_1=1}^{\omega_1}\sum_{k_2=1}^{\omega_2}dissim(k_1,k_2) \tag{9}$$

其中，S_i,S_j 为文档中的任意两个句子，$\sin(S_i,S_j)$为 S_i,S_j 的正弦夹角，$dissim(k_1,k_2)$判断 S_i,S_j 中的任意一对单词是否在同一棵 WordNet 分类树中，取值 0 或 1。对于任意两个节点 S_i,S_j，如果 $\sin(S_i,S_j)$大于一定阈值，那么 S_i,S_j 之间就构建一条权值为 $DS(S_i,S_j)$的边；否则节点 S_i,S_j 之间就没有边连接。

(2)单词作为节点。虽然使用句子作为节点更加普遍、直接，句子选择也相对简单，但是这种方法忽略了词汇或词组之间的高阶相关性，有许多自动文摘工作以词汇作为文本图的节点，同样取得了良好的效果。文献[37]提出的 GraphSum 算法使用频繁项集作为节点。GraphSum 首先根据词的共现使用 Apriori 算法发掘项集，选出支持度大于一定阈值的项集，即将频繁项集作为图的节点；再利用频繁项集之间的提升度作为边的权值来构建图。提升度的值恒大于 0，当提升度为 1 时，频繁项集之间的提升效果为零，这时两个频繁项集之间没有关系；当提升度位于(0,1)间时，频繁项集之间是负相关关系；提升度大于 1 时，频繁项集之间是正相关关系。GraphSum 算法同时考虑了频繁项集之间的正负相关关系，设立了最大负相关 max−lift 和最小正相关 min+lift 两个阈值。当两个频繁项集之间的提升度位于(0,max−lift)间或大于 min+lift 时，在两个频繁项集节点之间建立边，并根据提升度对边进行正负的标注，从而构建出有标注的无向无加权图。

文献[38]提出文档中的词汇应该分为主题词和非主题词，主题词应比非主题词权重更高。该方法根据伯努利模型和语义信息理论，通过词的共现得出词汇之间的语义信息量，然后将归一化之后的语义信息量作为节点之间边的权值，以此生成无向加权图。

(3)混合成分作为节点。为了在排序中可以融入词汇或语义的影响，使得排序结果更精确，许多方法采用混合文本单元作为节点构建文本图。文献[39]同时使用句子和单词作为节点来构建文本图，但对于单词节点，不使用单词本身，而使用该单词在 WordNet 中的释义作为节点，这样就可以统一句子和单词之间相似度的计算。文献[40]对图排序算法作出了基于聚类的改进，方法是先对文档句子做聚类，然后使用句子、单词和聚类簇作为节点构建图。文献[41]提出了一种将维基百科与图排序结合的方法。该方法首先使用文档句子作为查询关键词以获得维基百科的搜索结果；然后抽取出所有搜索结果的题目作为该句子映射的概念集合；最后将所有句子和概念集合分别作为节点来构建二分图，每个句子和其对应的概念集合中的每一项之间都生成一条无权重的边。

3.2.2 图排序

图排序是整个算法中最为关键的部分。在这一步中，根据迭代公式对文本图进行迭代计算直到收敛，就得到每个节点的最终权重。本小节从 PageRank 变种、HITS 变种以及其他情况 3 个方面介绍图排序过程。

(1)PageRank 变种。LexRank 算法[29,30]首次将图排序算法引入自动文摘领域。其迭代公式如(10)所示。

$$p(u)=\frac{d}{N}+(1-d)\sum_{v\in adj(u)}\frac{p(v)}{deg(v)} \tag{10}$$

其中，u 为总数为 N 的节点中的任意一点，$p(u)$代表节点 u 的权值，$adj(u)$为链向 u 的节点集合，$deg(v)$是节点 v 的出度，d 为阻尼系数。LexRank 由 PageRank 启发而来，从式(10)也可看出：LexRank 的迭代公式与 PageRank 算法的迭代公式几乎相同。

LexRank 所使用的文本图是无加权图，对于加权图，需要对迭代公式做加权的改进。使用加权图的 TextRank 算法[32]的迭代公式如式(11)所示。

$$WS(V_i)=(1-d)+d*\sum_{V_j\in In(V_i)}\frac{\omega_{ji}}{\sum_{V_k\in Out(V_j)}\omega_{jk}}WS(V_j) \tag{11}$$

可以看到，与 PageRank 或 LexRank 公式不同的是，TextRank将原来均分节点权重的$\frac{1}{deg(v)}$替换为标准化的边权值$\frac{\omega_{ji}}{\sum_{V_k\in Out(V_j)}\omega_{jk}}$，这样，节点在传播权重过程中对关联度高的邻近节点的贡献更高。GraphSum 算法[37]使用频繁项集以及频繁项集之间的提升度来构建图，且同时考虑了节点间正负相关性，其迭代公式也是对 PageRank 公式的加权变形，如式(12)所示。

$$PR(N_i)=\frac{(1-d)}{|N|}+d*(\sum_{k=1}^{e}\frac{1}{\sqrt{C-(N_k)}}\frac{PR(N_k)}{C(N_k)}) \quad (12)$$

式中，$|N|$是节点总数，e是连向N_i的边总数，$C(N_k)$是节点N_k的出度，d是阻尼系数，$C(N_k)$代表负相关的出度。可以看到，相较于TextRank，GraphSum算法使用节点间的负相关关系进行加权：如果一个节点对其他节点有负的提升度，那么这个节点对所有节点的贡献都要被削弱，削弱比例为$\frac{1}{\sqrt{C-(N_k)}}$。

文献[42]从基于查询的自动文摘角度提出了一种LexRank算法的变种：Biased-LexRank，该算法将马尔科夫链转移到自身的概率进行加权。加权方法是根据先决条件对节点赋予基本权值，其迭代公式如式(13)所示。

$$LR(u)=d\frac{b(u)}{\sum_{z\in C}b(z)}+(1-d)\sum_{v\in adj[u]}\frac{w(v,u)}{\sum_{z\in adj(u)}w(v,z)}LR(v) \quad (13)$$

其中，$LR(u)$为节点u的Biased-LexRank权重，C为文本构建的无向图中所有节点，$adj[u]$为与u相邻的所有节点，$w(v,u)$为节点v,u之间边的权值，$b(u)$为节点u的基础权值，d为阻尼系数，用以权衡基础权值和权重传播对最终排序的影响，在该文献中虽然以节点u基于查询条件的权值作为$b(u)$，但这种方法有很强的通用性，Biased的意义不只局限于查询，同样可以适用于通过其他约束条件得到的权值。

(2)HITS变种。基于HITS算法的变种适用于构建二分图的自动文摘方法。例如，文献[41]结合维基百科将句子映射为维基百科的概念生成二分图，利用句子与维基百科概念之间的相互关联进行迭代排序，迭代公式如式(14)所示。

$$y_j^{(k+1)}=\sum_{i\in N_j}x_i^{(k)},x_i^{(k+1)}=\sum_{j\in M_i}y_i^{(k)} \quad (14)$$

其中，$x_i^{(k)}$代表句子s_i的得分，N_j为与概念y_j相连的所有句子的集合，$y_j^{(k)}$代表概念z_j的得分，M_i为与句子x_i相连的所有概念集合，初始设置$x_i^{(0)}=1/\sqrt{n}$。此外，该文献还提出了对上述方法的扩展，根据标准化的词条点击率对概念与句子之间的边赋予权值。迭代公式如式(15)所示。

$$y_j^{(k+1)}=\sum_{i\in N_j}g_{ij}x_i^{(k)},x_i^{(k+1)}=\sum_{j\in M_i}h_{ij}y_j^{(k)} \quad (15)$$

需要注意的是，由于二分图的特性，赋予权重的过程将图变为了有向图。其中$g_{ij}=P(z_j|s_i)$代表句子s_i指向概念z_j的权重。$h_{ij}=P(s_i|z_i)$代表概念z_j指向句子s_i的权重，根据贝叶斯公式，计算$P(s_i|z_j)$的公式为：

$$P(s_i|z_j)=\frac{P(z_j|s_i)P(s_i)}{P(z_j)} \quad (16)$$

其中，$P(s_i)$和$P(z_j)$代表句子和概念的先验权重，如果没有额外的知识，可以将其设置为$1/m$和$1/n$，m和n分别为句子和概念的总数量。

(3)其他。PageRank或HITS算法的变种不适用于使用混合文本单元作为节点的文本图，但迭代公式的基本思想都是相同的：根据节点出度或边权值传播节点权重并利用参数均衡不同权重的影响。文献[40]使用句子、单词和聚类簇作为节点来构建图，根据3种成分之间的权重相互影响关系进行迭代排序，迭代公式如式(17)—式(19)所示。

$$r(s_i)=\alpha_1*\sum_{j=1}^{k}\boldsymbol{W}_{SC}(i,j)*r(c_j)+\beta_1*\sum_{l=1}^{m}\boldsymbol{W}_{ST}(i,l)*r(t_l) \quad (17)$$

$$r(c_j)=\alpha_2*\sum_{i=1}^{n}\boldsymbol{W}_{SC}(i,j)*r(s_j)+\gamma_1*\sum_{l=1}^{m}\boldsymbol{W}_{CT}(j,l)*r(t_l) \quad (18)$$

$$r(t_l)=\beta_2*\sum_{i=1}^{n}\boldsymbol{W}_{ST}(i,l)*r(s_i)+\gamma_2*\sum_{j=1}^{k}\boldsymbol{W}_{CT}(j,l)*r(c_j) \quad (19)$$

其中，$\boldsymbol{W}_{SC}(i,j)$为句子与聚类簇之间边的权重，用余弦相似度表示，$\boldsymbol{W}_{ST}(i,l)$和$\boldsymbol{W}_{CT}(j,l)$根据单词是否在句子或聚类簇中取值0或1。

3.2.3 句子选择

句子选择是基于图排序自动文摘的最终步骤。这一步中，利用节点权重以及节点和句子之间的相关关系，根据相应的选择策略选出最终的摘要句子，得到文章摘要。

(1)直接选择。LexRank算法以句子作为节点来构建图，经过图排序计算每个句子在文档中的权重。摘要句子选择过程中，通常根据实际需求选择权重排名top-n的句子或者总字数满足一定数量的句子，然后依照句子在原文的顺序组合生成最终的摘要。一般来说，大部分基于图排序算法的自动文摘方法都是使用句子作为图节点，因此也都是用这种方法直接选择摘要句子。

(2)去除冗余。除了在图构建阶段利用聚类算法之外，句子选择阶段是去除冗余的最佳时机。文献[43]的去冗余方法十分有代表性。其做法是：图排序之后只选出一个权重最大的句子归入摘要，然后将迭代过程中用于计算的邻接矩阵与此节点相关的行列设置为0，最后再次进行图排序，选择下一个句子。其减少冗余的思想是：一旦一个句子被选中，那么其余的文摘句中与这个句子有信息重叠的概率要降低到最小。这种方法的缺点是在减少冗余的同时也去除掉了已选中句子对图排序权重分布的全局影响，在一定程度上削弱了后续文摘句子的代表性；重复进行图排序也严重影响了自动文摘系统的性能。

(3)贪心算法。贪心算法不仅可以同时考虑句子多种得分的影响，还可以融入冗余削减，并且同时适用于普通文摘和基于查询的文摘。通常来说，使用贪心算法是句子选择的最佳方案，使用单词作为节点的文摘方法也可使用贪心算法将图排序结果应用于句子权重的衡量。GraphSum算法[21]利用频繁项集作为图的节点，在得到频繁项集的权重后利用贪心算法的思想选择句子，十分有代表性。该算法定义了两种衡量因素来选择文摘句子：句子重要性和覆盖度。句子重要性得分用于其所包含的频繁项集权重的标准化和，如式(20)所示。

$$SR(S_{jk})=\frac{\sum_{i|n_i\subseteq t_{jk}}PR_{ik}}{N_{t_{jk}}} \quad (20)$$

其中，$SR(S_{jk})$代表句子S_{jk}的重要性得分，t_{jk}代表经过文档预处理后的句子S_{jk}，$\sum_{i|n_i\subseteq t_{jk}}PR_{ik}$表示图中所有被$t_{jk}$包含的节点$n_i$的权值和，$N_{t_{jk}}$表示图中所有被$t_{jk}$包含的节点的数量。句子的覆盖度得分用于衡量句子对图中节点的包含程度，用长度等同于图中节点数的一维向量SC_{jk}表示。SC_{jk}的元素为$1_{t_{jk}}(n_i)$，数值为0或1，表示句子t_{jk}是否包含节点n_i，如式(21)所示。

$$1_{t_{jk}}(n_i)=\begin{cases}1, & \text{if } n_i\subseteq t_{jk}\\ 0, & \text{otherwise}\end{cases} \quad (21)$$

GraphSum算法利用贪心算法选择句子的基本步骤是：

1. 定义向量SC^*作为检测摘要包含度的向量，长度等同于图中节点数，初始为0。

2. 首先选择句子覆盖度最高(向量含 1 最多)的句子 $max\_one\_sentences$，再从 $max\_one\_sentences$ 中选择句子重要性最高的句子得到 SC_{best}，并将句子归入摘要。

3. 将向量 SC^* 与选择出的摘要句子做逻辑或运算 $SC^* = SC^*$ OR SC_{best}(这一步将向量 SC^* 中表示摘要已包含的节点的位设为 1)。

4. 将剩余的所有句子与 SC^* 的反向量做逻辑与运算，$SC_i = SC_i$ AND $\overline{SC^*}$。由于已经有句子节点在步骤 2 中被收入了摘要，这样就使得 SC_i 中与这些节点相关的位变为 0，这一步可以使得在接下来的循环中排除已经收入到摘要的句子的影响，减少摘要冗余。

5. 重复步骤 2－4，直到 SC^* 向量全部为 1，即摘要已经包含所有节点，就得到了最终的摘要。

3.3 小结

从上文对基于图排序算法的自动文摘的介绍可以看出，每种自动文摘方法都是一个包含预处理、构建文本图、图排序和句子选择等一系列步骤的系统，而每个步骤都是一系列技术的集合。已有的系统中：在图的构建方面，除了应用最广泛的余弦相似度度量外，基于关联规则挖掘[37]、信息论[38]等衡量文本单元相关性的方法也有所应用，以及利用超图[44]、聚类[40]、WordNet 以及维基百科的方法[36,39,41]也有所应用；在图排序方面，现有的系统大多是对 PageRank 或 HITS 算法基于所构建的文本图做出相应的改进，例如 TextRank 和 GraphSum 算法将 PageRank 算法的权重传播做了加权改进，Biased-LexRank[42]将马尔科夫链转移到自身的概率进行了加权，文献[41]从加权 HITS 算法中获得启发；也有方法进一步扩展了现有的图排序方法，例如文献[39]用全局排序的结果对聚类簇做动态更新，然后利用更新过后的聚类簇对句子进行重新排序；文献[40]将文本图中句子间的相互影响扩展为句子、单词和聚类簇三者之间的相互影响；在句子选择方面，大部分系统利用句子作为节点，在图排序后直接使用 top-n 的选择方式，但对于未直接使用句子作为图节点的方法，由于获得句子权重排名需要结合图排序结果以及其他多种因素，因此基于贪心思想的方法[37]使用相当广泛。

4 发展趋势

尽管自动文摘技术已经发展了超过 50 年，但是由于新需求的出现以及基础技术的革新，这仍然是一个充满活力的领域，因此基于图排序算法的自动文摘的未来发展趋势也应主要包括两个方面：1)由新需求驱动；2)由新技术驱动，改进现有的文摘方法或提出新的文摘方法。

随着网络技术以及 Web2.0 的发展，出现了一些专注于用户需求或海量数据处理的新的文摘类型：1)基于情绪的摘要：Web2.0 时代出现了大量诸如博客的高自主性的文本，文本单元权重的主观性很强，结合作者情绪生成的摘要才最能代表原文的主要内容；2)个性化摘要：不同用户对信息需求的程度不同，应根据用户特点生成专属摘要，满足用户个性化的需要；3)更新摘要：假设用户已经了解了某一主题或事件的背景知识，只需要生成包含这一主题的最新进展的摘要；4)概况摘要：根据海量数据，针对某一主题或者实体生成概览性的摘要，该摘要应尽可能包括这一主题或实体的所有方面的信息。新的摘要类型带来了新的挑战，基于情绪的摘要需要结合情感分析技术；个性化摘要依赖对用户个人信息的合理建模；更新摘要需要充分利用已有的摘要信息，确保得到的摘要与已有摘要的重叠度最小。新的摘要类型也带来了新的研究方向，针对新的摘要类型已经有一些相关研究。文献[45]使用包含用户固定信息以及反馈信息的关键词向量代表用户，通过计算用户关键词向量与文档句子的相关性抽取个性化摘要。文献[46]提出了过滤特征的概念，生成更新摘要的过程中，利用过滤特征可以排除已经存在于历史文摘中的内容。文献[47]针对时序动态文摘提出了当前文档集与历史文档集之间内容差异性的建模方法，这种方法为生成更新摘要提供了可行的思路。在基于图排序算法的自动文摘中，可以根据不同的新需求，在文本图构建、图排序和句子选择方面做出相适应的改进。近年来，DUC&TAC 会议的主题任务也由一般摘要转向了特定类型的摘要。需求是自动文摘技术发展的原动力，因此如何改进现有的或提出新的基于图排序的自动文摘方法以适应新需求应该是值得深入研究的一大热点。

文本图的构建是图排序的基础，文本单元之间的相关性度量是构建文本图的核心。自然语言处理作为一门蓬勃发展的学科，其研究成果层出不穷，为自动文摘研究提供了许多新的思路与可借鉴的成果，文本单元之间的相关性度量方面也出现了许多最新的研究成果。文献[48]提出了一种基于依存句法解析的单词相似性度量方法；文献[49]针对句子相似性度量提出了一种基于本体学习和语法规则的混合方法，克服了传统的基于概念相似性的方法的鲁棒性不强的问题。聚类在基于图排序算法的自动文摘中应用广泛，文献[50]提出了一种模糊关系聚类算法，相较于硬聚类，该算法允许聚类单元不同程度地属于所有聚类簇，在自然语言处理领域，文档的句子很可能不只与一个主题相关，因此这种模糊关系聚类算法更适合于句子聚类。因此在基于图排序算法的自动文摘中，使用新的聚类方法、度量方法等构建文本图或改进现有的构图方法应该是该领域的一大趋势。新技术不断涌现，其应用应不止局限于文本图的构建，如何将新技术融入图排序或句子选择步骤中也是值得深入研究的一大热点。

结束语 本文对基于图排序算法的自动文摘研究现状进行了综述，首先阐述了自动文摘的基本知识，然后从统计分析、主题发掘、篇章关系、机器学习和图模型 5 个方面介绍了基于抽取的自动文摘所应用的技术，进而阐述了 PageRank 和 HITS 两种图排序算法，说明了图排序算法的数学原理，然后从图的构建、图排序、句子选择 3 个方面详细探讨了目前基于图排序算法的自动文摘所应用的技术及其发展现状，最后探讨了在这一领域未来的发展趋势，受新需求和新技术两个方面的驱动，基于图排序的自动文摘方法已经在个性化摘要、基于情感的摘要、基于语义的文本图构建等技术和应用方面受到该领域研究者们的广泛关注，并将取得新的研究进展。

参 考 文 献

[1] Luhn H P. The automatic creation of literature abstracts [J]. IBM Journal of Research and Development, 1958, 2(2): 159-165

[2] Kleinberg J M. Authoritative sources in a hyperlinked environment[J]. Journal of the ACM (JACM), 1999, 46(5): 604-632

[3] Page L, et al. The PageRank citation ranking: Bringing order to the Web[R]. Stanford InfoLab, 1999

[4] Barzilay R, Elhadad M. Using lexical chains for text summarization[M]// ACL Workshop on Intelligent Scalable Text Summarization, 1997: 10-17

[5] Lin C-Y, Hovy E. Identifying topics by position [C]// Proceedings of the fifth conference on Applied natural language processing. 1997: 283-290

[6] Orasan C, Pekar V, Hasler L. A Comparison of Summarisation Methods Based on Term Specificity Estimation [C] // Proc of Lre. 2007: 35-39

[7] Orăsan C. Comparative Evaluation of Term-Weighting Methods for Automatic Summarization[J]. Journal of Quantitative Linguistics, 2009, 16(1): 67-95

[8] Edmundson H P. New methods in automatic extracting [J]. Journal of the ACM (JACM), 1969, 16(2): 264-285

[9] Li P, Jiang J, Wang Y. Generating templates of entity summaries with an entity-aspect model and pattern mining[C] // Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. 2010: 640-649

[10] 秦兵，刘挺，李生. 基于局部主题判定与抽取的多文档文摘技术[J]. 自动化学报, 2005, 30(6): 905-910
Qin Bing, Liu Ting, Li Sheng. Multi-Document Summarization Based on Local Topics Identification and Extraction[J]. Acta Automatica Sinica, 2005, 30(6): 905-910

[11] Marcu D. Discourse trees are good indicators of importance in text[M] // Advances in Automatic Text Summarization. 1999: 123-136

[12] William M, Thompson S. Rhetorical structure theory: Towards a functional theory of text organization[J]. Text, 1988, 8(3): 243-281

[13] Khan A U, Khan S, Mahmood W. MRST: A New Technique For Information Summarization[C]// WEC (2). 2005: 249-252

[14] Da Cunha I, et al. A new hybrid summarizer based on vector space model, statistical physics and linguistics [M] // MICAI 2007: Advances in Artificial Intelligence. Springer, 2007: 872-882

[15] Orasan C. The influence of personal pronouns for automatic summarisation of scientific[C] // Proceedings of the Discourse Anaphora and Anaphor Resolution Colloquium. 2004: 5

[16] Mitkov R, et al. Anaphora resolution: To what extent does it help NLP applications? [M]// Anaphora: Analysis, Algorithms and Applications. Springer, 2007: 179-190

[17] Kupiec J, Pedersen J, Chen F. A trainable document summarizer [C]// Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 1995: 68-73

[18] Schlesinger J D, et al. Understanding machine performance in the context of human performance for multi-document summarization[OL]. http://citeseerx. ist. psu. edu/viewdoc/cittons?doi=10. 1. 1. 5. 23

[19] Conroy J M, O'leary D P. Text summarization via hidden markov models[C] // Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2001: 406-407

[20] Aone C, Okurowski M E, Gorlinsky J. Trainable, scalable summarization using robust NLP and machine learning[C] // Proceedings of the 17th international conference on Computational linguistics-Volume 1. Association for Computational Linguistics, 1998: 62-66

[21] Svore K M, Vanderwende L, Burges C J. Enhancing Single-Document Summarization by Combining RankNet and Third-Party Sources[C]// EMNLP-CoNLL. Citeseer, 2007: 448-457

[22] Burges C, et al. Learning to rank using gradient descent[C]// Proceedings of the 22nd International Conference on Machine Learning. ACM, 2005: 89-96

[23] Schilder F, Kondadadi R. FastSum: fast and accurate query-based multi-document summarization[C] // Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers. Association for Computational Linguistics, 2008: 205-208

[24] Li S, et al. Multi-document summarization using support vector regression[C]// Proceedings of DUC. Citeseer, 2007: 45-50

[25] Mani I, Bloedorn E. Multi-document summarization by graph search and matching [OL]. http://axXiv. preprint. cmp-lg/9712004

[26] Wan X, Xiao J. Towards a unified approach based on affinity graph to various multi-document summarizations[C] // European Conference on Research & Advance Technology for Digital Libraries. 2007: 297-308

[27] Giannakopoulos G, Karkaletsis V, Vouros G. Testing the use of N-gram Grahps in Summarization Sub-tasks[C]// Natural Language Processing. 2008: 324-334

[28] Morales L P, Esteban A D, Gervás P. Concept-graph based biomedical automatic summarization using ontologies [C] // Proceedings of the 3rd Textgraphs Workshop on Graph-Based Algorithms for Natural Language Processing. Association for Computational Linguistics, 2008: 53-56

[29] Erkan G, Radev D R. LexRank: Graph-based lexical centrality as salience in text summarization[J]. J. Artif. Intell. Res. (JAIR), 2004, 22(1): 457-479

[30] Erkan G, Radev D R. LexPageRank: Prestige in Multi-Document Text Summarization[C]// EMNLP. 2004: 365-371

[31] Mihalcea R, Tarau P. TextRank: Bringing order into texts[J/OL]. http://digital. library. unt. edu/ark/67531/metadC30962

[32] Mihalcea R. Graph-based ranking algorithms for sentence extraction, applied to text summarization[C]// Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions. Association for Computational Linguistics, 2004: 20

[33] Ferreira R, et al. A Four Dimension Graph Model for Automatic Text Summarization[C]// 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT). IEEE, 2013: 389-396

[34] Ferreira R, et al. A new sentence similarity assessment measure based on a three-layer sentence representation[C]// Proceedings of the 2014 ACM symposium on Document engineering. ACM, 2014: 25-34

[35] Ester M, et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]// Kdd. 1996: 226-231

（下转第 39 页）

求，但是对于一个复杂的软件系统来说，事实上还是有很多的信息并没有完全地表现出来。比如说有关代码的责任人、代码修改的最后提交时间等。

在软件度量信息方面，本文的设计并没有很好地把代码耦合等逻辑关系显示出来，而耦合关系的可视化也是软件演化可视化的重要组成部分，将来可以通过交通路线等来展现不同类之间的耦合关系。

在三维布局上，由于二维迭代布局分布的局限，如果某些代码扩张特别严重，动画或许在视觉感受上会很突兀。

用户的视觉体验会决定用户判断的敏感度，因此界面的美观程度也有待提升，使其更符合城市的感官体验。

5.2 未来的展望

当前本文的城市模型里还有很多元素并没有被加入使用，比如说公路、人，甚至交通情况等。这些元素还可以承载更多的软件开发过程中的情况，比如说人在一座大楼里面工作，可以象征着开发人员在写代码或者修正代码等，这些比喻需要更精巧的思考，以获得最真实的效果和更好的用户体验。

在维度的设计上，还有很多其他可视化元素没有被使用，比如海拔、颜色饱和度、透明度等。这些元素可以用来表达更多的度量信息。例如，透明度可以表示最后更改时间，颜色饱和度可以表示开发人数量，海拔可以用来表述一个类或者包是否是核心算法的类或包等。

另外，布局上也可以突破二维的限制，通过架设立交桥等其他设想来表现更好的逻辑耦合关系连接，让整个设计既能符合人的社会认知，又能更好、更丰富地表现系统内部的各种信息等。

可视化工具已经成为当前软件维护开发的重要组成部分，而将抽象的软件系统结构比拟成用户所熟悉的现实环境的具体形象，可以更好地提升用户体验，让一些抽象的特征更加直观，从而使软件管理者更好地掌握关键信息，快速做出正确的决策。相信随着设计的进一步完善，它一定能为开发人员提供更大的帮助。

结束语 本文通过对软件开发过程进行建模并使用三维动画做演示，将抽象的软件开发过程对应成城市的发展过程，使管理者更好地获得软件开发过程的反馈，从而提高了软件开发效率，确保了软件开发质量。

参 考 文 献

[1] Ben S. Control Flow and Data Structure Documentation: Two Experiments [J]. Communications of ACM, 1982, 25(1): 55-63

[2] Limberger D, Wasty B, Trumper J, et al. Interactive Software Maps for Web-Based Source Code Analysis[C]// Proceedings of the International Web3D Conference, 2013. ACM, 2013, 475(3): 91-98

[3] Wettel R, Lanza M. Visualizing software systems as cities[C]// VISSOFT. 2007. Banff, Ont., 2007: 92-99

[4] Beyer D, Hassan A E. Animated Visualization of Software History using Evolution Storyboards[C]// WCRE 2006. Benevento, Italy, 2006: 199-210

[5] Langelier G, Sahraoui H, Poulin P. Exploring the evolution of software quality with animated visualization[C]// Visual Languages and Human-Centric Computing. 2008: 13-20

[6] Few S. Show me the numbers: Designing Tables and Graphs to Enlighten[M]. Analytics Press, 2004

[7] Wettel R, Lanza M. Program Comprehension through Software Habitability[C]// ICPC. 2007. Banff, Alberta, Canada, 2007: 231-240

(上接第 7 页)

[36] Ramesh A, Srinivasa K, Pramod N. SentenceRank—A graph based approach to summarize text[C]// 2014 Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT). IEEE, 2014: 177-182

[37] Baralis E, et al. GraphSum: Discovering correlations among multiple terms for graph-based summarization[J]. Information Sciences, 2013, 249: 96-109

[38] Goyal P, Behera L, McGinnity T M. A Context-Based Word Indexing Model for Document Summarization[J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 25(8): 1693-1705

[39] Cai X Y, Li W J. Ranking Through Clustering: An Integrated Approach to Multi-Document Summarization[J]. IEEE Transactions on Audio Speech and Language Processing, 2013, 21(7): 1424-1433

[40] Zhang Z C, Ge S S, He H S. Mutual-reinforcement document summarization using embedded graph based sentence clustering for storytelling [J]. Information Processing & Management, 2012, 48(4): 767-778

[41] Sankarasubramaniam Y, Ramanathan K, Ghosh S. Text summarization using Wikipedia[J]. Information Processing & Management, 2014, 50(3): 443-461

[42] Otterbacher J, Erkan G, Radev D R. Biased LexRank: Passage retrieval using random walks with question-based priors[J]. Information Processing & Management, 2009, 45(1): 42-54

[43] Hariharan S, Ramkumar T, Srinivasan R. Enhanced graph based approach for multi document summarization[J]. Int. Arab J. Inf. Technol., 2013, 10(4): 334-341

[44] Wang W, et al. Exploring hypergraph-based semi-supervised ranking for query-oriented summarization[J]. Information Sciences, 2013, 237: 271-286

[45] Diaz A, Gervas P. User-model based personalized summarization [J]. Information Processing & Management, 2007, 43(6): 1715-1734

[46] Li S, Wan W, Wang C. TAC 2008 update summarization task of ICL[C]// Proceedings of the Text Analysis Conference (TAC). 2008: 132-139

[47] 刘美玲，等. 动态多文档文摘模型[J]. 软件学报，2012，23(2)：289-298
Liu Mei-ling, et al. Dynamic Multi-Document Summarization Model[J]. Journal of Software, 2012, 23(2): 289-298

[48] Minkov E, Cohen W W. Adaptive graph walk-based similarity measures for parsed text[J]. Natural Language Engineering, 2014, 20(3): 361-397

[49] Lee M C, Chang J W, Hsieh T C. A Grammar-Based Semantic Similarity Algorithm for Natural Language Sentences[J]. Scientific World Journal, 2014(1): 437162

[50] Skabar A, Abdalgader K. Clustering Sentence-Level Text Using a Novel Fuzzy Relational Clustering Algorithm[J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 25(1): 62-75