|  |  |
| --- | --- |
| **民大校徽** | 中央民族大学学士学位论文 |
| **Bachelor Thesis of Minzu University of China** |

基于TextRank的中文文本摘要和关键词提取

|  |  |  |  |
| --- | --- | --- | --- |
| 姓 名： | 薛志豪 | | |
| 学 号： | 16046023 | 年 级： | 2016级 |
| 院 系： | 信息工程学院 | | |
| 专 业： | 软件工程 | | |
| 指导教师： | 邱莉榕 | | |

2020年4月12日

摘 要

当代人们每天制造着大量数据，每天传播信息，每天也接收无数信息。现如今的人们处于信息爆炸的时代，飞速发展的网络技术、层出不穷的新媒体，每天制造着无法计数的信息。在信息越加丰富多元的同时，如何在海量信息中快速搜寻和准确辨识所需要的内容，成为人们面对的最大难题，也是当代群众浏览活动的最大困扰。在线上线下每天处理的海量文档数据中，只有极少数的学术论文，绝大部分的文章不包含任何的关键词信息，有的甚至连标题都没有。我们需要一种高效的方法实现中文文本关键词提取和文本摘要，这就是本文将讨论的问题。

本文依据TextRank算法设计了一个中文文本关键词提取和自动文本摘要的系统。主要包含以下方面的内容：

（一）依据TextRank算法的关键词提取技术。

在该算法中默认窗口为5，以窗口内词频（Term Frequency）为初始值，选取图运算后TR值最大的几位作为关键词。

（二）融合BM25的TextRank文本摘要技术。

该技术选取BM25算法作为句间相似度的计算方法，并以IDF值为词特征，最终依据TextRank图运算获取句子的TR值。选中TR值最的几句，以原文顺序拼接得到文本摘要。

本文还涉及引入ICTCLAS中文分词、词性过滤、去停用词、相似度计算方法等内容，并最终在关键词提取和文本摘要上取得了良好的效果。

关键词： 关键词提取；文本摘要；TextRank；BM25

**Abstract**

Contemporary people make a lot of data every day, spread information every day, and receive countless information every day. People are now in an era of information explosion. Rapidly developing network technologies and endless new media are creating countless information every day. While information is becoming richer and more diverse, how to quickly search and accurately identify the required content in massive information has become the biggest problem faced by people and the biggest problem for contemporary mass browsing activities. Among the massive document data processed daily online and offline, there are only a few academic papers, most of which do not contain any keyword information, and some do not even have titles. Therefore, we need an efficient method to achieve Chinese text keyword extraction and text summary, which is the issue that this article will discuss.

This paper designs a system for Chinese text keyword extraction and automatic text summarization based on the TextRank algorithm. It mainly includes the following aspects:

(1) Keyword extraction technology based on TextRank algorithm.

In the algorithm, the default window is 5, the term frequency in the window is used as the initial value, and the digits with the largest TR value after graph operation are selected as keywords.

(2) TextRank text summarization technology integrated with BM25.

This technology selects the BM25 algorithm as the calculation method for the similarity between sentences, and uses the IDF value as the word feature, and finally obtains the TR value of the sentence according to the TextRank graph operation. Select the few sentences with the highest TR value and splice them in the original text order to get a text summary.

This article also involves the introduction of ICTCLAS Chinese word segmentation, part-of-speech filtering, removal of stop words, similarity calculation methods, etc., and finally achieved good results in keyword extraction and text summary.

**Key Words:** Keyword Extraction; Text Summarization; TextRank；BM25

目录

[**一 绪论** 7](#_Toc39432364)

[（一）研究背景 7](#_Toc39432365)

[（二）研究意义 7](#_Toc39432366)

[（三）国内外研究现状 8](#_Toc39432367)

[1、关键词提取 8](#_Toc39432368)

[2、文本摘要 9](#_Toc39432369)

[**二 相关研究与技术** 11](#_Toc39432370)

[（一）中文分词技术 11](#_Toc39432371)

[（二）去停用词 12](#_Toc39432372)

[（三）关键词提取概要 12](#_Toc39432373)

[1. 关键词定义 13](#_Toc39432374)

[2. 关键词提取 13](#_Toc39432375)

[3. 关键词提取的过程 13](#_Toc39432376)

[（五）相关算法与模型 14](#_Toc39432377)

[1. 基于统计的TF-IDF算法[22] 14](#_Toc39432378)

[2. 基于图论的TextRank图算法[16] 15](#_Toc39432379)

[3. BM25算法[24] 15](#_Toc39432380)

[**三 基于TextRank关键词提取技术的设计** 17](#_Toc39432381)

[（一）总体架构图 17](#_Toc39432382)

[（二）预处理文本 17](#_Toc39432383)

[1. ICTCLAS分词 17](#_Toc39432384)

[2. 去停用词 18](#_Toc39432385)

[3. 预处理部分流程图 18](#_Toc39432386)

[（三）TextRank图算法 19](#_Toc39432387)

[1. 创建词汇窗口 19](#_Toc39432388)

[2. 创建图模型 20](#_Toc39432389)

[3. 进行图运算 20](#_Toc39432390)

[**三 融合BM25的TextRank文本摘要方法与设计** 21](#_Toc39432391)

[（一）总体架构图 21](#_Toc39432392)

[（二）文本预处理 21](#_Toc39432393)

[1. 分句 21](#_Toc39432394)

[2. 分词与去停用词 21](#_Toc39432395)

[（三）相似度计算 22](#_Toc39432396)

[1. 计算IDF词特征 22](#_Toc39432397)

[2. 计算句间BM25值，构建相似度矩阵 22](#_Toc39432398)

[（四）图迭代计算 23](#_Toc39432399)

[（五）提取摘要句与重排序 23](#_Toc39432400)

[**三 实验结果与总结** 24](#_Toc39432401)

[（一）实验结果 24](#_Toc39432402)

[（二）总结与展望 24](#_Toc39432403)

[**参考文献** 26](#_Toc39432404)

[**附 录** 28](#_Toc39432405)

[附录1 实验所用文本 28](#_Toc39432406)

[附录2 实验文本分句结果 28](#_Toc39432407)

[附录3 英文文献 29](#_Toc39432408)

[附录4 文献翻译 34](#_Toc39432409)

[**致 谢** 41](#_Toc39432410)

**一 绪论**

本部分是论文第一章内容，主要讲述论文研究的背景、研究意义以及国内外研究现状。

（一）研究背景

互联网的快速发展连接起万事万物，人们足不出户就可知悉天下事，坐居家中就可静等外卖快递，遥居天涯也能实时面晤。随着时代信息化和数字化进程，网络联入千千万万的设备,连接起千千万万的人。每天都有企业推出越来越多的网络服务，同时每天都制造着巨大的信息流。据估计，2020年网络数据将超过350亿TB[1]。数据量的急速膨胀显示了网络内信息、资源量的庞大。

在这个信息爆炸的时代，人们被形形色色的信息淹没，真正需要的信息常会被数不清的无关内容掩盖，总让人们感到无所适从。曾经在信息检索领域，只能依靠人工方式整理堆积如山的尺牍文件。传统的人工整理手段具有较高的准确度，但效率和成本往往不可接受。信息时代庞大的信息量更使得人工方式不再可能实现，于是人们尝试让计算机来进行这一工作。

要将网络资源化作人们实实在在的便利，面临着层出不穷的问题。丰富的信息为我们提供巨大的便利，也增加了人们检索数据和利用信息的复杂度。如何有效地组织、检索信息便成为了网络技术发展过程中最重要的议题。本文依据TextRank算法提出的关键词提取和文本摘要技术，就是基于该背景。

（二）研究意义

面对海量的数据资源，用户如何从中搜寻、识别所需要的资源呢？从计算机技术大发展之后，人们便开始使用计算机进行信息的存储和处理。在传统信息传播中，人们往往标注出文章的关键词或提取一个文本摘要来概括文本信息。关键词和文本摘要帮助用户快速把握文章内容，捕捉关键信息，是如今信息检索的关键技术。

我们所说的关键词，是指与文章主题密切关联的词语集合，常常作为文章摘要的依据。文本关键词可以帮助用户从大量文本中迅速把握文章的中心内容，协助用户准确检索得到自己所需要的文档。但在现实中，除少数学术类论文会包含关键词的标注外，绝大部分资源并不包含类似标识，甚至连标题都没有。曾经信息传播行业，通过人工方式手动提取关键词。这种方式往往具有极高的准确度，但效率低下，尤其面对海量文本时，其成本显得不可接受。因此，自人们利用计算机处理数据以来，便产生了依托计算机的数据处理和信息检索技术。计算机实现关键词提取技术，具有效率高但准确度低的特点。数十年来，人们一直致力于开发更高效的算法和跟准确的数据处理技术。

文本摘要技术比关键词提取技术更加复杂。文本摘要是对一个文本进行概括、提炼，得到的一段简介的描述性文字。本文中即将讨论的自动摘要技术，是通过计算机技术从文本中提炼出高度概括的语句，拼接形成有代表意义的摘要文字的技术。随着信息领域的扩张，文本摘要技术也越来越多地表现出不可替代的作用。

近二十年以来，人们积极钻研自然语言处理技术，提出了众多计算机自动提取关键词的算法。关键词提取技术是自然语言处理和数据挖掘技术的基础性工程之一，探索一个更加准确高效的关键词提取方法意义非凡[6]。目前主流的文本摘要技术大多从原文中选取代表性语句，再按原文顺序拼接为摘要文本。通过研究各种成熟模型，本文依据TextRank算法设计了一个关键词提取技术，以及融合BM25的TextRank文本摘要技术。

（三）国内外研究现状

1、关键词提取

上世纪五十年代，Luhn[7]首次提出了自助标引相关的技术。此后，人们在该技术领域持续研究五十余年。这项技术的研究一直延续到上世纪九十年代，出现关于继承自动标引技术的自动关键词提取技术[8]。

二十世纪末，随着全文索引技术广泛应用，同时传统自助标引方法发效率问题也遇到了天花板。自此，自助标引相关的技术便走上了下坡路，逐步为世人所抛弃[9]。二十世纪九十年代末到现在，全文索引技术也越来越难适应新的时代。人们需要更加精确的标注结果，这一时期涌现的自动摘要、文本聚类、文档分类和主题检测等新兴互联网服务也对信息检索技术提出了更高的要求。而这些服务都极度依赖于一项准确而高效的关键词提取技术，解决好关键词提取相关的技术才能实现高质量的网络信息服务。自此，关键词提取相关技术成为了技术领域的风口，这一时期得到了快速发展。

关键词提取技术的研究最早在国外发起，现今成熟的关键词提取解决方案大多由国外专家提出。其中，产生了众多极具实践价值的算法和模型[10]。1999年，Witten[11]借助朴素贝叶斯的方法进行关键词特征模型的建立，开发了KEA系统。该系统将关键词的抽取视作有监督学习的过程，并基于此建立了全新的预测模型。此后，Turney[12]还率先将机器学习中遗传算法和决策树的方法引入关键词提取技术。Turney将关键词的提取作为一个文本分类问题解决——将文本中候选词分为关键词和非关键词两类，再依据大量已标注训练集得到训练模型。很显然该方法过于粗糙，也十分依赖训练集的规模和质量，且对于不同类型文本的关键词提取效果差异很大。

基于有监督学习方法打关键词提取技术，在实现以及实现后的通用性、准确度方面并不能提供一个很好的解决方案，此类技术往往并不可靠。于是，出现了如今主流的众多基于无监督训练方案的关键词提取技术。Matsuo和Ishizuka[13]共同提出了一种可应用于单个文本且不需要一个高质量标注数据集的关键词提取技术。他们提出一种数据压缩算法获取一项特征值n-gram，过滤非关键词元素，依据权重排序获取文本关键词，就是N-gram filtrationt算法。Hulth[14]提出了从学术论文中提取关键词的算法Rule Induction——在词频统计的基础上加入了句法分析相关的方法。同时，还对词语的词性进行区分，得到一种准确度较高的自动提取算法。后来，Liu和Tonella[15]还依据TF-IDF算法提出了一种关键词自动提取技术，原理简单，且效果不俗。近年来，受PageRank算法启发，陆续有学者将图运算的方法引入关键词提取领域，产生了著名的TextRank算法。

国内关键词提取技术大约开始于上世纪八十年代末，同时期国际上相关研究成果已经比较多了。国内很多学者另辟蹊径，专注于中文文本关键词提取相关技术的研究，获得了很多有价值的研究成果。如果不考虑中文分词等差异，大部分关键词提取算法都适用于中文。2008年，杨[16]将TextRank算法运用到中文关键词提取领域。基于TextRank算法的关键词提取算法成为，中文关键词提取的优秀解决方案之一。

2、文本摘要

与关键词提取技术相类似的，文本摘要技术同样随着信息的膨胀变得火热。文本摘要技术对于用户把握文章基本信息，筛选重要内容至关重要。现代搜索技术是为用户检索、筛选、压缩信息，快速提取关键内容。精准高效的文本摘要技术，有利于帮助节省用户浏览成本，提升用户体验。

文本摘要技术从上世纪五十年代开始获得学者关注，几十年来出现了许多自动文本摘要技术。传统的自动文本摘要方法大多依据统计学方法，依据语料库构建。上世纪90年代，互联网技术的极速膨胀，带来了迫切的文本摘要技术的需求。新世纪机器学习、人工智能技术的蓬勃发展，也为文本摘要技术带来了许多新思路和方法。

1995年，Kupiee[17]基于朴素贝叶斯模型设计了一种文本摘要方法，该模型将句子视为相互独立的事件，通过朴素贝叶斯方法获取权重。1999年，Aone在此基础上提出了新的解决方案——Dimsim系统，考虑到更多的文本特征。2004年后，图算法被引入文本摘要领域，形成了新的研究方向。2004年，Mihalcca[16]和Tarau已经PageRank算法，提出了运用于文本摘要的TextRank算法。目前，基于图算法的比较通用的模型，就是TextRank。

80初，国内学者逐渐注意到文本摘要相关的研究。王永根开始了中文文本摘要技术的研究。90年代末，王开铸提出了基于理解文摘的文本摘要模型。在这之后，中文文本摘要技术的研究开始吸引越来越多的学者涌入。随着2004年图模型引入文本摘要领域，涌现了许多中文摘要解决方案[18]。

主流的文本检索技术大致可分为两种，有监督方法和无监督方法[19]。有监督的文本摘要方法大致有朴素贝叶斯、向量机方法等机器学习方法。无监督的文本摘要方法有着更广泛的应用，TextRank以其简洁高效等特点广受青睐。

**二 相关研究与技术**

这一部分将会介绍本文研究过程中会涉及到的技术与概念，例如关键词提取和文本摘要技术的描述，以及IDF、TextRank和BM25等算法和模型。

（一）中文分词技术

中文是典型的象形文字，与英文等字母文字不同，中文词与词之间没有空格作为间隔。中文以字为基础，而且单个字往往无法表示具体含义，而是有多个字构成有实际含义的短语。中文里词与词的分割是基于语句的，没有明确间隔。对于关键词提取，最基本且最重要的就是准确地进行分词。在进行中文信息检索时，通常需要先进行分词，其他对于内容、语义等的理解都与分词的质量关联。中文分词作为中文自然语言处理的基础性工程，直接影响自然语言处理对语义的把握准确度[20]。

本文采用中科院计算技术研究所开发的NLPIR-ICTCLAS系统进行中文分词。ICTCLAS如今各种功能详备，性能可靠。目前，ICTCLAS系统最新2016版本已至此中英文混合分词和基于交叉信息熵算法的关键词提取功能（包括新词与已知词）。如今成熟的分词工具有很多， NLPIR-ICTCLAS主要有四点突出优势[21]：

（1） 性能最优

ICTCLAS采取PDAT知识库管理技术，在高效率和高准确性方面均获得了提升，能够管理百万容量的词典库，单机最高检索速度能达到100万条/秒，并且内存占用不足字典库容量的3/2。由于该技术的重大突破，ICTCLAS单机分词速度达到996KB/s，分词精度超过98.45%，各种词典压缩容量不足3M，成为了如今世界上性能最好的中文分词工具。

（2） 统一的计算理论框架

ICTCLAS根据层叠隐马尔可夫模型构建，获得了最好的实践效果。该成果在理论上和实践上都得到了印证，证明了该模型的先进性。

（3） 良好的可移植性

ICTCLAS系统使用C/C++完成编写，目前支持包含Linux、FreeBSD、Windows的几乎所有主流系统，并且提供了支持C/C++、C#、Java、Python语言调用方式，适用范围广，使用灵活。

（4） 良好的扩展性

ICTCLAS所有功能都是模块化设计的，可以自由组装、自行调用。用户可以根据需求，组装最适合自身的分词系统。

（二）去停用词

对停用词是指在信息检索过程中，为了节省存储空间和提高处理速度，在进行预处理过程中会自动过滤的字或词，这种词被称为stop\_words（停用词）。在英文文本中，停用词往往是一些冠词、量词之类，如the、a。而在中文文本中，停用词常常表现为虚词、符号和一些无意义字符片段。在进行自然语言处理之前，剔除这些无意义的停用词，对于后续提取关键词的效率和准确度都大有裨益。停用词主要包括虚词这种无实义的词、助词等只为句法结构服务的词。停用词往往与文本主题无关，不但不能代表文本内容，其高频次还会影响关键词的提取，增加语言处理的复杂度。去停用词还能减少词条数、缩小候选词规模，对于提高精度和效率至关重要。

表 1 部分停用词表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 停用词表（节选） | | | | | |
| a  an  and  are  as  at  be  but  by  for  if  in  into  is  it  no  not  of  on  or | such  that  the  their  then  there  these  they  this  to  was  will  with  !  "  #  $  %  &  ' | ::  ;  <  =  >  >>  ?  @  A  Lex  [  \  ]  ^  \_  `  exp  sub  sup  | | 一个  一些  一何  一切  一则  一则通过  一天  一定  一方面  一旦  一时  一来  一样  一次  一片  一番  一直  一致  一般  一起 | 上下  上升  上去  上来  上述  上面  下  下列  下去  下来  下面  不  不一  不下  不久  不了  不亦乐乎  不仅  不仅...而且  不仅仅 | 不妨  不定  不对  不少  不尽  不尽然  不巧  不已  不常  不得  不得不  不得了  不得已  不必  不怎么  不怕  不惟  不成  不拘  不择手段 |

（三）关键词提取概要

1. 关键词定义

关键词是指与文章主题直接关联，能够概括文章核心内容的重要词汇，英文称为keyword。我们看关键词字面意思，很容易将关键词理解为单个的词，事实上能够准确概括文本的往往是多个词构成的关键短语。研究表明，大部分关键词实际上都是一个短语结构，即多个词的组合。这主要是因为短语结构包含明确语义，能够准确表达文章主题，最适合成为文本概括的关键词。关键单词的提取难度较小，但是关键短语的提取相对比较复杂。

关键词按其所表示的文本范围不同，可以分为“局部关键词”和“全文关键词”。局部关键词指能够概括全文某一部分、段落、章节的关键短语，它能够准确表达一定范围内的内容，准确性较高。全文关键字是对整个文本内容的凝练和概括，涉及范围广，与主题联系紧密。

2. 关键词提取

关键词提取也被称为关键词标引，即一种将文本中具有代表性、具有辨识性的词汇提取出来的技术，是从文本中提取出能概括文档主题、并与其他文本区分的词语。

关键词提取有两种基本方式：传统的人工方法和如今的计算机关键词提取技术。在计算机技术大规模运用之前，人们要整理大量文档的关键词，需要雇佣人工进行概括、整理。人工方式提取关键词具有准确性高、概括度强的特点，但是主观性强烈，而且效率极其低下。计算机实现的关键词提取技术，是指通过计算机技术实现主题概括。计算机关键词提取技术具有效率高但是准确度差的特点，所以关键词提取技术所依赖的算法优劣直接关系到关键词提取的效率和质量。

关键词提取是如今自然语言处理的基础领域和核心技术。在信息检索、情感分析、数据挖掘和文本分类集类领域，高质量的关键词提取可以提高整个系统的效率和成效，尤其对于大规模文本处理具有深刻意义。

3. 关键词提取的过程

关键词提取过程中，将一个文本视作一个词语集合，该词语集合众多词具有聘书、位置、词长、词性等特征属性。而关键词提取技术，就是依据这些特征判断哪些词可作为关键词，并从中提取出来。所以，关键词提取技术主要有以下两个部分：

（1） 对文档进行与处理，获取关键词的候选词集合

这是一个基础性的问题，往往发生在文本预处理阶段，需要进行一系列的词语抽取和过滤 得到候选词集合。首先对文本进行分词操作，将文本转化为一个词汇集合。该集合中会包含很多符号以及虚词、助词等与关键词提取无关的字词，所以还需要提出这一部分字符。目前主流的方法是依据停用词表，过滤这一类无关字符。

（2） 依据一定算法分析各项特征，从候选词中选出最适合作为关键词的集合

这一部分就是关键词提取技术的核心部分，依据一定算法考虑各项特征值产生关键词权重，并选定关键词。关键词提取算法中可考虑的特征有词频、位置、词性、词长和反文本频率等，依据一定统计学方法，如互信息量、最大熵、交叉信息熵等进行加权操作。最终得到文本中候选词的权重，以及权重排序，选中权值大的候选词作为文本关键词。

（五）相关算法与模型

1. 基于统计的TF-IDF算法[22]

介绍TF-IDF算法是因为后续基于TextRank的关键词提取和文本摘要设计中，涉及到词频、逆文档频率等特征，同时TF-IDF特征也常用来做相似度比较，所以将此算法作为引入。

TF-IDF算法的核心原理是，词的重要性与它的频数成正比，而且与它在语料库中的频数成反比。即，一个关键词在语料库中的文档频率会很低，而在该文本中词频却会较高。

1. 词频TF（term frequency）

词频指的是某一词语在文件中的频率。能够作为关键词的词语，往往在文本中出现得很频繁，即在单个文本中词与的重要程度与其出现次数呈现一定正比关系。而为了削弱长文本与短文本的差异，我们选择词频作为特征值引入算法。词频TF如下公式（1）：

|  |  |
| --- | --- |
|  | 公式（1） |

在该公式中，ni代表词语ti在文档dj中出现的频数，Σknij表示文档dj中所有词语的频数之和。

1. 逆文档频率IDF（Inverse Document Frequency）

逆文档本质上是一种一直噪音的加权方式，该方法认为文档频率越小的词汇对于单一文本越重要，而文本频率越大则越不能作为代表性词语[23]。在IDF的计算方法中，只在少数文档中出现的词比在很多文档中出现的词语更具有代表性，也因此会获得更大的权重。IDF的计算公式如下公式（2）：

|  |  |
| --- | --- |
|  | 公式（2） |

其中，N代表语料库中文本总数，N(x)代表语料库中包含词x的文档数。因为N/N(x)在语料库训练中往往回事一个远远小于1的小数，很容易下溢出，所以我们通过取对数的方式使得IDF的值更加平滑。但还有一个问题，就算语料库很庞大，还是可能有一些词语在语料库中出现频数为0，这就可能导致计算公式中分母为0。于是IDF有另一个修正后的计算公式，一般也都用这个计算公式（3）：

|  |  |
| --- | --- |
|  | 公式（3） |

TF-IDF算法的最终值TFIDF(x) = TFi,j \* IDFi ；即词i在文本j中出现的词频乘于词i在语料库中的逆文档频率IDFi得到TD-IDF算法的最终权值。TF-IDF算法原理简单，但其有效性在自然语言处理领域已经得到了印证。TF-IDF作为一种统计类的权重计算方法，在关键词提取领域效用非常。

2. 基于图论的TextRank图算法[16]

TexRank算法根据PageRank算法改进而来，是PageRank在文本处理中的应用。PageRank的核心思想是：

1. 如果有很多网页链接到同一网页，那就说明该网页价值高，相应的PR值较大；
2. 如果一个网页被另一个PR值很大的网页链接，也说明该网页重要，PR值较大。

PageRank算法设计时将整个网络空间看作一个有向图，每一个网页是图上一个节点，网页之间的连接、引用是图上的有向边。算法运行时，为每一结点赋予一个初值，每一个引用会将引用网页的权值的一部分按比例分给被引用网页，该计算公式如下公式（4）：

|  |  |
| --- | --- |
|  | 公式（4） |

该公式中表示页面i被页面j引用；式经过调整使其运算结果能够最终收敛。

而TextRank与PageRank算法是一脉相承的，思想也很相似：

1. 如果一个单词出现在其他单词后的频数很高，则说明它很重要，TR值大；
2. 如果一个单词出现在另一个TR值很高的词后面，也说明它很重要，TR值提高。

这样的话，我们可以写出公式（5）：

|  |  |
| --- | --- |
|  | 公式（5） |

该公式表示单词i的权重取决于与在i前面的各个点j组成的(j,i)这条边的权重，以及j这个点到其他其他边的权重之和。该模型也被成为词图模型。关于运用到关键词提取和文本摘要中的技术细节，将在后续章节介绍。

3. BM25算法[24]

传统的Text Rank算法运用于文本摘要时，通过重叠词计算句间相似度。这种相似度计算方法缺点明显，无法正确反映文本特征。

BM25算法通常用来做搜索相关性，它的基本思想是：将一个查找语句Query分解为语素集合{qi}，计算每一个语素与搜索文本Doc的相关性，最终将所有语速相关性求和，便得到查找语句Query与待检索文本Doc的相关性。一般公式如下公式（6）：

|  |  |
| --- | --- |
|  | 公式（6） |

其中Q表示搜索语句，D表示待检索文本，qi表示语素（一般是Q的分词结果），Wi是语素qi的权值（一般是qi在Q中词频），R(qi,D)是语素qi与D的相关性（本文选择IDF值作为相关性特征）。

BM25最关键的参数就是R(qi,D)，词与文档的相关性特征。最通用的会选择IDF值作为相关性特征，本文BM25算法中IDF计算公式如下公式（7）：

|  |  |
| --- | --- |
|  | 公式（7） |

其中N为文档总量，n(qi)为含有语素qi的文档数，计算可得逆文档频率IDF值。

而相关性特征R(qi,D)的计算方法仍比较复杂，其一般公式如下公式（8）（9）：

|  |  |
| --- | --- |
|  | 公式（8） |

|  |  |
| --- | --- |
|  | 公式（9） |

其中k1，k2，b都是调节参数，本文中k1设为1.5，b设为0.75。fi是qi在Q中词频，dl为文档D长度，avgdl是文本集平均文本长度。参数b是用于调节文档长度对相关性影响的。

**三 基于TextRank关键词提取技术的设计**

本章节将介绍基于TextRank算法的关键词提取技术的详细设计，包括总体架构、分词技术、去停用词、图算法等。

（一）总体架构图

如下图，从左至右分别是分词与去停用词、构建词汇图模型、TextRank迭代运算等部分，最终输出关键词序列。

图 1 基于TextRank的关键词提取总体架构图

待测文本

有序词表

分词与去停用词

划分词窗口

构建图模型

TextRank迭代

排序

关键词

（二）预处理文本

### ICTCLAS分词

不论是待训练语料库，还是测试文本，都必须先进行文本分词。每一个文本都被分词处理为词条组成的集合，这些词条才是后续处理的基本单元。本文中采用中国科学院计算技术研究所开发的ICTCLAS系统实现中英文混合分词，ICTCLAS不光支持分词和词性标注，还支持词频统计，为后续处理提供很大方便。

表 2 分词结果展示

|  |
| --- |
| 部分分词结果示例 |
| 据/p 俄罗斯/nsf 卫星网/user 8月/t 11日/t 发布/v 美国/nsf 《/wkz 国家/n 利益/n 》/wky 杂志/n 刊登/v 的/ude1 文章/n 称/v ，/wd 中国/ns 购买/v 俄/b 制/v 苏/b -/wp 27/m 第四/m 代/q 战机/n ，/wd 为/p 本国/rzs 空军/n 翻开/v 了/ule 现代史/n 的/ude1 页/q 章/n 。/wj 从/p 那时/rzt 起/f ，/wd 中国/ns 空军/n 日益/d 强大/a 。/wj 中国/ns 空军/n 长期以来/dl 落后/a 于/p 像/v 美国/nsf 这样/rzv 的/ude1 世界/n 大国/n ，/wd 从/p 2008年/t 起/vf 中国/ns 开始/v 研制/v 堪/vg 舆/n 美国/nsf F/o -/wp 22/m 猛禽/n 战机/n 和/cc F/o -/wp 35/m 闪电/n -/wp II/n 相/d 媲美/vi 的/ude1 第五/m 代/q 战机/n J/n -/wp 20/m 和/cc J/n -/wp 31/m ，/wd 不仅/c 用/p 它们/rr 装备/v 本国/rzs 空军/n ，/wd 而且/c 还/d 在/p 国际/n 市场/n 销售/vn 。/wj 它/rr 让/v 中国/ns 具有/v 了/ule 远程/b 打击/vn 的/ude1 能力/n ，/wd 能/v 达到/v 西/f 太平洋/ns 的/ude1 任何/rz 地点/n 。/wj J/n -/wp 31/m 可能/v 成为/v J/n -/wp 20/m 的/ude1 有力/a 补充/vn ，/wd 是/vshi 理想/a 的/ude1 战机/n ，/wd 能/v 在/p 西/f 太平洋/ns 切断/v 重要/a 地区/n 。/wj J/n -/wp 31/m 升空/vi 后/f ，/wd 完全/ad 能/v 应/v 对/p 美国/nsf 的/ude1 F/o -/wp 35/m 。/wj 这些/rz 战机/n 能/v 从/p 根本/n 上/f 改变/v 中国/ns 同/p 美国/nsf 以及/cc 同/p 台湾/ns 地区/n 冲突/vn 的/ude1 走向/vn 。/wj 如果/c 中国/ns 大陆/n 通过/p 台湾/ns 海峡/n 进攻/v 台湾/ns (/wkz 解放军/n 每年/r 都/d 要/v 进行/vx 这/rzv 方面/n 的/ude1 演练/vn )/wky ，/wd 因为/c 拥有/v 最/d 先进/a 的/ude1 战机/n 而/cc 具有/v 的/ude1 空中/s 优势/n 是/vshi 解放军/n 进攻/v 取胜/vi 的/ude1 关键/n 因素/n 。/wj 这/rzv 无疑/d 应当/v 引起/v 美国/nsf 的/ude1 不安/an ，/wd 无论是/c 从/p 战略/n 上/f ，/wd 还是/c 从/p 战术/n 上/f 以及/cc 从/p 机动性/n 上/f ，/wd 美国/nsf 《/wkz 国家/n 利益/n 》/wky 警告/v 。/wj |

如上表所示分词结果，以空格为间隔，结果会标注词性。例如“战机/n”，名词；“而且/c”，连词；“。/wj”，标点。

### 去停用词

本文中根据哈工大停用词表（共计2700余条）进行停用词过滤，提出与文本关键词提取无关的词条，压缩文本处理规模。经过分词和停用词过滤后产生的词条集合就是候选词集合了，可以作为关键词提取阶段的输入信息。

本文所描述的去停用词过程，会过滤数次、连词、叹词、拟声词、介词、量词等，还会屏蔽停用词表中包含的词语。部分停用词表请参考第二章表1。

### 预处理部分流程图

图 2 分词与去停用词流程图

导入文本

ICTCLAS分词系统

停用词表

去停用词

有序词表

文档

单文本经过预处理后会形成一个有序的词表，上节分词内容，预处理后结果如下：

表 3 预处理结果展示

|  |
| --- |
| 预处理后结果 |
| {俄罗斯\nsf, 8月\t, 11日\t, 发布\v, 美国\nsf, 国家\n, 利益\n, 杂志\n, 刊登\v, 文章\n, 称\v, 中国\ns, 购买\v, 制\v, 战机\n, 空军\n, 翻开\v, 现代史\n, 章\n, 中国\ns, 空军\n, 强大\a, 中国\ns, 空军\n, 落后\a, 美国\nsf, 世界\n, 大国\n, 2008年\t, 中国\ns, 研制\v, 堪\vg, 舆\n, 美国\nsf, 猛禽\n, 战机\n, 闪电\n, II\n, 相\d, 媲美\vi, 战机\n, J\n, J\n, 装备\v, 空军\n, 国际\n, 市场\n, 销售\vn, 中国\ns, 打击\vn, 能力\n, 西\f, 太平洋\ns, 地点\n, J\n, J\n, 补充\vn, 理想\a, 战机\n, 西\f, 太平洋\ns, 切断\v, 地区\n, J\n, 升空\vi, 应\v, 美国\nsf, 战机\n, 改变\v, 中国\ns, 美国\nsf, 台湾\ns, 地区\n, 冲突\vn, 走向\vn, 中国\ns, 大陆\n, 台湾\ns, 海峡\n, 进攻\v, 台湾\ns, 解放军\n, 演练\vn, 拥有\v, 先进\a, 战机\n, 空中\s, 优势\n, 解放军\n, 进攻\v, 取胜\vi, 关键\n, 因素\n, 无疑\d, 美国\nsf, 不安\an, 战略\n, 战术\n, 机动性\n, 美国\nsf, 国家\n, 利益\n, 警告\v,} |

很容易看出，已过滤助词、连词和停用词，获得了一个有序表，表元素就是语素。

（三）TextRank图算法

### 创建词汇窗口

TextRank算法是依据PageRank算法建立的，本质是一个图算法。我们设定词汇窗口为5，与词汇q距离5以内的词汇，都会为词汇q提供一个贡献值，贡献值的流通方法参考上一章公式，而词汇TR值的初值依据窗口内词数设定。

表 4 词窗口展示

|  |
| --- |
| 部分窗口展示（窗口值为5） |
| 11日=[8月, 俄罗斯, 利益, 发布, 国家, 美国]  2008年=[世界, 中国, 堪, 大国, 研制, 美国, 舆, 落后]  8月=[11日, 俄罗斯, 发布, 国家, 美国]  II=[J, 媲美, 战机, 猛禽, 相, 美国, 闪电]  J=[II, 切断, 升空, 国际, 地区, 地点, 太平洋, 媲美, 市场, 应, 战机, 理想, 相, 空军, 美国, 能力, 补充, 装备, 西]  不安=[关键, 因素, 战术, 战略, 无疑, 机动性, 美国]  世界=[2008年, 中国, 大国, 研制, 空军, 美国, 落后]  中国=[2008年, 世界, 冲突, 刊登, 制, 台湾, 国际, 地区, 堪, 大国, 大陆, 太平洋, 市场, 应, 强大, 战机, 打击, 改变, 文章, 杂志, 海峡, 现代史, 研制, 称, 空军, 章, 美国, 翻开, 能力, 舆, 落后, 西, 购买, 走向, 进攻, 销售]  优势=[先进, 关键, 取胜, 战机, 拥有, 空中, 解放军, 进攻]  俄罗斯=[11日, 8月, 发布, 美国]  先进=[优势, 台湾, 战机, 拥有, 演练, 空中, 解放军]  关键=[不安, 优势, 取胜, 因素, 无疑, 美国, 解放军, 进攻]  冲突=[中国, 台湾, 地区, 大陆, 美国, 走向]  切断=[J, 升空, 地区, 太平洋, 应, 战机, 理想, 西]  刊登=[中国, 利益, 国家, 文章, 杂志, 称, 美国, 购买] |

### 创建图模型

如上表所示，窗口内词都会提供一个贡献值给原词汇，我们据此便可以构建窗口内词汇到原词汇的一条有向边

。。。

图 3 图模型部分示例展示

如上图所示，“11日”收到来自6个词汇的贡献值，并将向右侧6个词汇提供提供贡献值，“11日”窗口内词数为6，所有其TR初值为6。

### 进行图运算

图运算模型建立完成后就是进行迭代运算了。设定迭代运算次数上限为200，最大迭代误差为0.001f。

表 5 TextRank迭代结果展示

|  |
| --- |
| 部分迭代结果展示（top20条） |
| 美国:3.8274937;中国:3.405885;战机:3.3051665;J:1.8571391;空军:1.6860673;进攻:1.407505;台湾:1.3851693;地区:1.3647405;国家:1.34468;解放军:1.2305794;利益:1.226108;西:1.2042682;太平洋:1.2014244;无疑:0.89724874;杂志:0.8872335;因素:0.88611233;文章:0.88280547;关键:0.87925905;猛禽:0.87711215;刊登:0.8757052; |

于是我们便可以选取TR值最大的前10条或20条作为关键词序列了。该实验文本经过了11次迭代得到目标结果。

我们可以取[美国;中国;战机;J;空军;进攻;台湾;地区;国家;解放军;利益;西;太平洋;无疑;杂志;因素;文章;关键;猛禽;刊登;]作为关键词。应该说，还是很有概括性的。

**三 融合BM25的TextRank文本摘要方法与设计**

本章节将介绍基于融合BM25的TextRank文本摘要技术的详细设计，包括文本预处理、句间相似度、图运算等内容。

（一）总体架构图

如下图所示，本文摘要方法包含分局与分词、相似度计算、TextRank图运算以及语句重排序等模块。

待测文本

分句词频集合

分句与分词

建立相似度矩阵

构建图模型

IDF算法

BM25算法

相似度计算模块

TextRank迭代

文本摘要

提取摘要句并重排序

图 4 融合BM25的TextRank文本摘要技术架构图

（二）文本预处理

### 分句

基于TextRank的文本摘要方法是要从原文中选取有价值的语句，构成摘要文本。文本中的单句被视为图运算的节点，任两节点之间都有运算边，边的权值为句间相似度，再迭代运算。所以该文本摘要方法的第一步就是分句，本文以[。？?！!；;]为分句间隔。详细分句结果见附录2所示。

### 分词与去停用词

该步骤可参考上一章分词和去停用词步骤。区别在于，本章分词的单元是句子，并且输出结果是词频集合，而不是有序表。

导入句子

ICTCLAS分词系统

停用词表

去停用词

词频集合

文档分句

图 5 文本摘要预处理流程图

（三）相似度计算

上一步骤得到的是分句的词频集合，本节将讨论句间相似度的计算方法。本文选择BM25算法作为相似度计算方法，选择IDF作为词特征。这里需要特别解释IDF，我们将单文本分句，然后统计词qi出现于各句的频数，计算逆文本频率

### 计算IDF词特征

我们将IDF值作为词特征，将每一个分句都视为一个文档，总句数为文档总数，并据此统计IDF值。试验文本部分IDF值展示如下：

表 6 部分IDF值展示

|  |
| --- |
| 部分IDF值示例 |
| 11日:2.7080502011022096;2008年:2.7080502011022096;8月:2.7080502011022096;II:2.7080502011022096;J:1.7676619176489945;不安:2.7080502011022096;世界:2.7080502011022096;中国:0.7884573603642702;优势:2.7080502011022096;俄罗斯:2.7080502011022096;先进:2.7080502011022096;关键:2.7080502011022096;冲突:2.7080502011022096;切断:2.7080502011022096;刊登:2.7080502011022096;利益:2.151762203259462;制:2.7080502011022096;升空:2.7080502011022096;发布:2.7080502011022096;取胜:2.7080502011022096; |

### 计算句间BM25值，构建相似度矩阵

根据第二章公式（6）可计算出各语句之间的相似度，作为图运算边的权值，可以构建一个相似度矩阵。

图 6 TextRank图模型示例

BM

BM

BM

BM

BM

BM

BM

BM

BM

（四）图迭代计算

分句和相似度矩阵就可以进行迭代运算了。该迭代运算以句子与其他语句的总和相似度为初值。同样设定最大迭代次数200，最大误差0.001f。

表 7 图迭代结果展示

|  |
| --- |
| 迭代结果 |
| {0.7999757788281731/ 1.1667595263673693/ 0.7347247099734802/ 0.15000000000000002/ 1.3240888396721087/ 1.5278160917881811/ 1.6435856347999025/ 0.832871145587601/ 0.15000000000000002/ 0.6626669744911712/ 0.8188513849688095/ 0.9111709266857193/ 0.7602282663197968/ 1.0333947086434558/ 0.8609206045338191/ 0.7580023993295545/ 1.765852055397935/ 0.9610864983346381/ 0.7317756694003548/ 0.702013652382172/ 0.15000000000000002/ 0.15000000000000002/ 1.0042151324957542/} |

上述值与分句结果一一对应，选取其中TR值最大的几句序号，便可以得到摘要候选句。

（五）提取摘要句与重排序

图运算迭代后就可以得到句子TR值排序，将需要长度的摘要句提出并恢复原文顺序，拼接后就可以得到摘要文本了。

表 8 摘要结果展示

|  |
| --- |
| 摘要结果 |
| 中国购买俄制苏-27第四代战机。中国空军日益强大。中国空军长期以来落后于像美国这样的世界大国。从2008年起中国开始研制堪舆美国F-22猛禽战机和F-35闪电-II相媲美的第五代战机J-20和J-31。J-31可能成为J-20的有力补充。能在西太平洋切断重要地区。美国《国家利益》警告。 |

在限定摘要长度150字时，得到上表9所示摘要文字，共6句按原文顺序排列。比较好地概括了实验文本的内容。

**三 实验结果与总结**

本章将展示最终实验成果，并评估所取得的成效，对本文研究内容做出总结和展望。

（一）实验结果

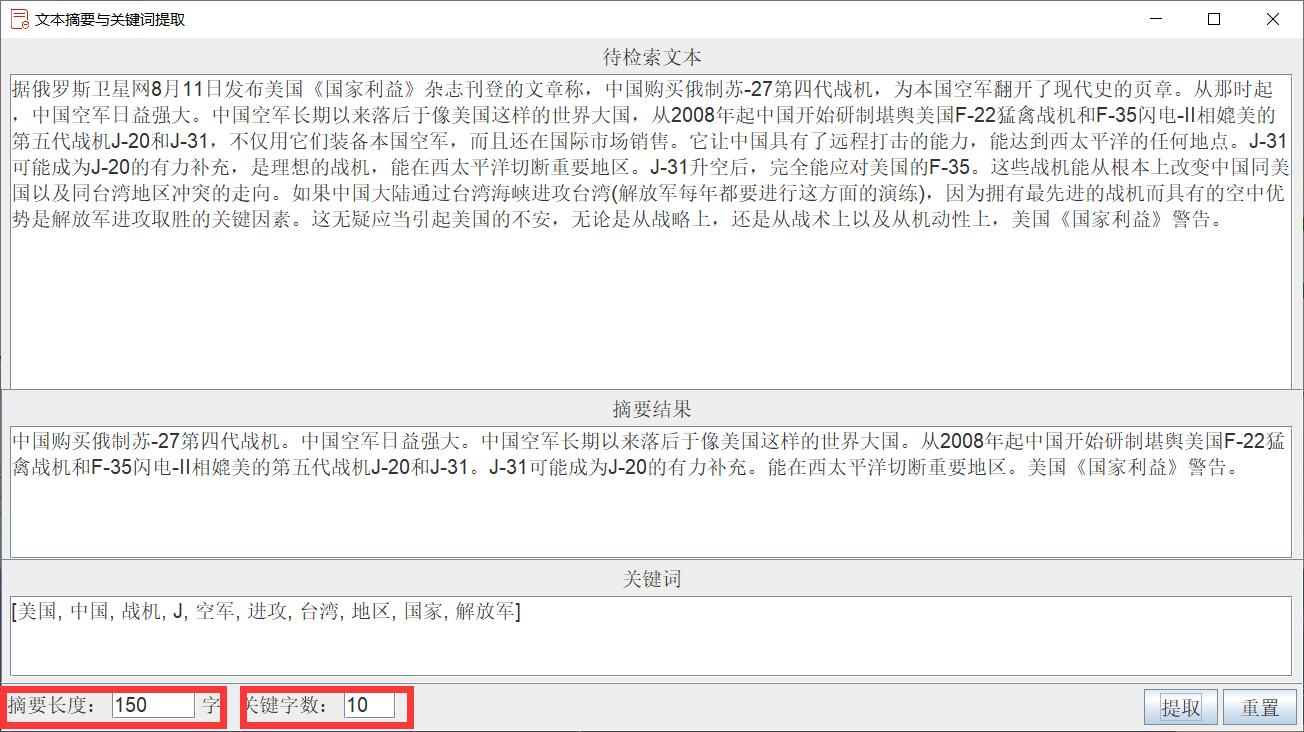
 本文研究内容是基于Java 1.8和IDEA 2020.1环境开发，算法完成后还依据Java swing设计了一个图形界面，展示效果如下：

图 7 应用界面展示

输入文本后，可选摘要长度和关键字数，并最终输出摘要结果和关键词序列。本文基于TextRank算法的两个不同应用，经多次试验，总体达到了较好效果。

在做这个设计作业之前，我曾依据TF-IDF做过一个关于关键词提取的作业设计。但是TF-IDF的算法需要依赖于一个完备语料库训练idf值，而且对于不同类别文本实现效果差异极大，这类方法运行成本很高，实现成果也差强人意。后来，我就尝试能不能完成一个不依赖于语料库质量的关键词提取作业设计。

TextRank算法是有独特优势的，它实现简单，不依赖于外部参数输入，而且能够取得较好效果。尤其是基于TextRank算法的文本摘要设计，相似度算法有多种选择方案。我看现在成熟的方案有建立词袋模型，依据机器学习相关技术求句间相似度。但我最终选择的BM25算法，因为BM25算法原本就是用于评价搜索相关性的方案，实现简单。

（二）总结与展望

事实上，本文所实现的TextRank关键词提取和文本摘要设计是相当粗糙的，如今有很多针对它的改进方案。

1、句间相似度目前比较好的解决方案是神经网络技术，通过训练大规模数据集实现，可以是其具有更好的实用性，使实验结果呈现更稳定。

2、本文关键词提取算法事实上是一种完全的统计学方法，未考虑词性、词长、语义等诸多特征。很多学者都提出了融合多特征的解决方案，可以取得很大进步。

3、本文设计的两个设计都并不适用于微博等短文本，微博短文本关键词的提取和热点分析近年来倍受学者关注。

**参考文献**

1. 于朝晖.CNNIC发布第44次《中国互联网络发展状况统计报告》[J].网信军民融合,2019(09):30-31.
2. 李晨.网络搜索引擎与专家检索系统框架和模型研究[D].北京邮电大学,2009.
3. 王毅.基于 web 的信息抽取方法研究[J].科技与生活,2010(13):11.
4. 姜舟.关键短语抽取及相关技术研究[D].哈尔滨工业大学,2010.
5. 王莉,许凯.浅谈文本数据自动标引系统的设计[J].图书馆理论与实践,2013(6):95-97.
6. 杨林. 基于文本的关键词提取方法研究与实现[D].安徽工业大学,2013.
7. H. P. Luhn. A Statistical Approach to Mechanized Encoding and Searching of Literry Information[J]. IBM Journal of Research and Development , 1957, 1(4): 309-317.
8. 张静,自动标引技术的回顾与展望[J].现代情报,2009(4):221-225
9. 罗准辰.关键词抽取的研究与实现[D].国防科学技术大学,2008.
10. 杨凯艳. 基于改进的TFIDF关键词自动提取算法研究[D].湘潭大学,2015.
11. Niraj Kumar, Kannan Srinathan. Automatic keyphrase extraction from scientific documents usingn-gramfiltrationtechnique[C].ProceedingsofDocEng'08Conference,2008:199-208.
12. Turney P. D. Learning Algorithms for Keyphrase Extration[J]. Information Retrieval. 2000, 2(2): 303-336.
13. Matsuo Y., Ishizuka M..Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information[J]. International Journal on Artificial Intelligence Tool, 2003, 13:2004-2008.
14. Hulth. An improved automatic keyword extraction given more linguistic knowledge[A]. In: the 2003 Conference on Empirical Methods in Natural Language Processing[C], 2003:216-233.
15. Liu Y. , Ciliax B. J. ,Borges K., etc. Comparison of Two Schemes for Automatic Keyword Extraction from Medline for Functional Gene Clusteringe[A]. In: the 2004 IEEE Computational Systems Bioinformatics Conference(CSB 2004)[C], 2004.
16. 李娜娜. 基于TextRank的文本自动摘要研究[D].山东师范大学,2019.
17. 徐馨韬. 基于Doc2Vec和改进的TextRank的中文单文档摘要研究[D].中国电子科技集团公司电子科学研究院,2019.
18. 陈达. 基于TextRank和Word2Vec的短文本自动摘要算法研究[D].武汉理工大学,2018.
19. 完颜丹丹. 基于词嵌入的文本摘要系统的设计与实现[D].海南大学,2017.
20. 周恕义,徐晓亮.基于字典的中文切词系统在北京市精品课程平台中的设计与实现[J].中国 教育技术装备,2010(9):3-5.
21. 戴耿毅. ICTCLAS汉语词法分析系统的研究与改进[D].浙江工业大学,2012.
22. 徐文海,温有奎.一种基于 TFIDF 方法的中文关键词抽取算法[J].情报理论与实 践,2008(2):298-302.
23. 施聪莺, 徐朝军, 杨晓江.TFIDF 算法研究综述[J].计算机应用,2009,29:167-170.
24. 李楠,陶宏才.一种新的融合BM25与文本特征的新闻摘要算法[J].成都信息工程大学学报,2018,33(02):113-118.
25. 李志强,潘苏含,戴娟,胡佳佳.一种改进的TextRank关键词提取算法[J/OL].计算机技术与发展, 2020(03): 1-5.
26. 门家乐.基于TextRank的关键词提取算法[J].电子世界,2018(15):31-32.
27. 杜海舟,陈政波,钟孔露.基于上下文关系和TextRank算法的关键词提取方法[J].上海电力学院学报,2017,33(06):607-612.
28. 曹洋. 基于TextRank算法的单文档自动文摘研究[D].南京大学,2016.
29. 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( 11) : 2 344-2 351．

**附 录**

附录1 实验所用文本

本文中关键词提取和文本摘要所引用实验文本均为以下语篇，来自于网络新闻（美媒：解放军歼20可媲美F22 美大为不安-新华网 http://www.xinhuanet.com/mil/2015-08/13/c\_128123590.htm）。

表 9 本文中所引用实验文本

|  |
| --- |
| 实验文本 |
| 据俄罗斯卫星网8月11日发布美国《国家利益》杂志刊登的文章称，中国购买俄制苏-27第四代战机，为本国空军翻开了现代史的页章。从那时起，中国空军日益强大。中国空军长期以来落后于像美国这样的世界大国，从2008年起中国开始研制堪舆美国F-22猛禽战机和F-35闪电-II相媲美的第五代战机J-20和J-31，不仅用它们装备本国空军，而且还在国际市场销售。它让中国具有了远程打击的能力，能达到西太平洋的任何地点。J-31可能成为J-20的有力补充，是理想的战机，能在西太平洋切断重要地区。J-31升空后，完全能应对美国的F-35。这些战机能从根本上改变中国同美国以及同台湾地区冲突的走向。如果中国大陆通过台湾海峡进攻台湾(解放军每年都要进行这方面的演练)，因为拥有最先进的战机而具有的空中优势是解放军进攻取胜的关键因素。这无疑应当引起美国的不安，无论是从战略上，还是从战术上以及从机动性上，美国《国家利益》警告。 |

附录2 实验文本分句结果

表 10 实验文本分句结果

|  |
| --- |
| 分句结果示例 |
| [据俄罗斯卫星网8月11日发布美国《国家利益》杂志刊登的文章称]、[中国购买俄制苏-27第四代战机]、[为本国空军翻开了现代史的页章]、[从那时起]、[中国空军日益强大]、[中国空军长期以来落后于像美国这样的世界大国]、[从2008年起中国开始研制堪舆美国F-22猛禽战机和F-35闪电-II相媲美的第五代战机J-20和J-31]、[不仅用它们装备本国空军]、[而且还在国际市场销售]、[它让中国具有了远程打击的能力]、[能达到西太平洋的任何地点]、[J-31可能成为J-20的有力补充]、[是理想的战机]、[能在西太平洋切断重要地区]、[J-31升空后]、[完全能应对美国的F-35]、[这些战机能从根本上改变中国同美国以及同台湾地区冲突的走向]、[如果中国大陆通过台湾海峡进攻台湾(解放军每年都要进行这方面的演练)]、[因为拥有最先进的战机而具有的空中优势是解放军进攻取胜的关键因素]、[这无疑应当引起美国的不安]、[无论是从战略上]、[还是从战术上以及从机动性上]、[美国《国家利益》警告] |

附录3 英文文献

本文献节摘自论文Mihalcea R, Tarau P. TextRank: Bringing order into texts[C]. Association for Computational Linguistics, 2004.

**Abstract**

In this paper, we introduce TextRank – a graph-based ranking model for text processing, and show how this model can be successfully used in natural language applications. In particular, we propose two innovative unsupervised methods for keyword and sentence extraction, and show that the results obtained compare favorably with previously published results on established benchmarks.

**Introduction**

Graph-based ranking algorithms like Kleinberg’s HITS algorithm (Kleinberg, 1999) or Google’s PageRank (Brin and Page, 1998) have been successfully used in citation analysis, social networks, and the analysis of the link-structure of the World Wide Web. Arguably, these algorithms can be singled out as key elements of the paradigm-shift triggered in the ﬁeld of Web search technology, by providing a Web page ranking mechanism that relies on the collective knowledge of Web architects rather than individual content analysis of Web pages. In short, a graph-based ranking algorithm is a way of deciding on the importance of a vertex within a graph, by taking into account global information recursively computed from the entire graph, rather than relying only on local vertex-speciﬁc information.

Applying a similar line of thinking to lexical or semantic graphs extracted from natural language documents, results in a graph-based ranking model that can be applied to a variety of natural language processing applications, where knowledge drawn from an entire text is used in making local ranking/selection decisions. Such text-oriented ranking methods can be applied to tasks ranging from automated extraction of keyphrases, to extractive summarization and word sense disambiguation (Mihalcea et al., 2004).

In this paper, we introduce the TextRank graphbased ranking model for graphs extracted from natural language texts. We investigate and evaluate the application of TextRank to two language processing tasks consisting of unsupervised keyword and sentence extraction, and show that the results obtained with TextRank are competitive with state-of-the-art systems developed in these areas.

**Text as a Graph**

To enable the application of graph-based ranking algorithms to natural language texts, we have to build a graph that represents the text, and interconnects words or other text entities with meaningful relations. Depending on the application at hand, text units of various sizes and characteristics can be added as vertices in the graph, e.g. words, collocations, entire sentences, or others. Similarly, it is the application that dictates the type of relations that are used to draw connections between any two such vertices, e.g. lexical or semantic relations, contextual overlap, etc. Regardless of the type and characteristics of the elements added to the graph, the application of graphbased ranking algorithms to natural language texts consists of the following main steps:

1. Identify text units that best deﬁne the task at hand, and add them as vertices in the graph.

2. Identify relations that connect such text units, and use these relations to draw edges between vertices in the graph. Edges can be directed or undirected, weighted or unweighted.

3. Iterate the graph-based ranking algorithm until convergence.

4. Sort vertices based on their ﬁnal score. Use the values attached to each vertex for ranking/selection decisions.

In the following, we investigate and evaluate the application of TextRank to two natural language processing tasks involving ranking of text units: (1) A keyword extraction task, consisting of the selection of keyphrases representative for a given text; and (2) A sentence extraction task, consisting of the identiﬁcation of the most “important” sentences in a text, which can be used to build extractive summaries.

**Keyword Extraction**

The task of a keyword extraction application is to automatically identify in a text a set of terms that best describe the document. Such keywords may constitute useful entries for building an automatic index for a document collection, can be used to classify a text, or may serve as a concise summary for a given document. Moreover, a system for automatic identiﬁcation of important terms in a text can be used for the problem of terminology extraction, and construction of domain-speciﬁc dictionaries.

he simplest possible approach is perhaps to use a frequency criterion to select the “important” keywords in a document. However, this method was generally found to lead to poor results, and consequently other methods were explored. The state-ofthe-art in this area is currently represented by supervised learning methods, where a system is trained to recognize keywords in a text, based on lexical and syntactic features. This approach was ﬁrst suggested in (Turney, 1999), where parametrized heuristic rules are combined with a genetic algorithm into a system for keyphrase extraction - GenEx - that automatically identiﬁes keywords in a document. A different learning algorithm was used in (Frank et al., 1999), where a Naive Bayes learning scheme is applied on the document collection, with improved results observed on the same data set as used in (Turney, 1999). Neither Turney nor Frank report on the recall of their systems, but only on precision: a 29.0% precision is achieved with GenEx (Turney, 1999) for ﬁve keyphrases extracted per document, and 18.3% precision achieved with Kea (Frank et al., 1999) for ﬁfteen keyphrases per document.

More recently, (Hulth, 2003) applies a supervised learning system to keyword extraction from abstracts, using a combination of lexical and syntactic features, proved to improve signiﬁcantly over previously published results. As Hulth suggests, keyword extraction from abstracts is more widely applicable than from full texts, since many documents on the Internet are not available as full-texts, but only as abstracts. In her work, Hulth experiments with the approach proposed in (Turney, 1999), and a new approach that integrates part of speech information into the learning process, and shows that the accuracy of the system is almost doubled by adding linguistic knowledge to the term representation.

In this section, we report on our experiments in keyword extraction using TextRank, and show that the graph-based ranking model outperforms the best published results in this problem. Similar to (Hulth, 2003), we are evaluating our algorithm on keyword extraction from abstracts, mainly for the purpose of allowing for a direct comparison with the results she reports with her keyphrase extraction system. Notice that the size of the text is not a limitation imposed by our system, and similar results are expected with TextRank applied on full-texts.

**TextRank for Keyword Extraction**

The expected end result for this application is a set of words or phrases that are representative for a given natural language text. The units to be ranked are therefore sequences of one or more lexical units extracted from text, and these represent the vertices that are added to the text graph. Any relation that can be deﬁned between two lexical units is a potentially useful connection (edge) that can be added between two such vertices. We are using a co-occurrence relation, controlled by the distance between word occurrences: two vertices are connected if their corresponding lexical units co-occur within a window of maximum N words, where N can be set anywhere from 2 to 10 words. Co-occurrence links express relations between syntactic elements, and similar to the semantic links found useful for the task of word sense disambiguation (Mihalcea et al., 2004), they represent cohesion indicators for a given text.

The vertices added to the graph can be restricted with syntactic ﬁlters, which select only lexical units of a certain part of speech. One can for instance consider only nouns and verbs for addition to the graph, and consequently draw potential edges based only on relations that can be established between nouns and verbs. We experimented with various syntactic ﬁlters, including: all open class words, nouns and verbs only, etc., with best results observed for nouns and adjectives only, as detailed in section 3.2.

The TextRank keyword extraction algorithm is fully unsupervised, and proceeds as follows. First,the text is tokenized, and annotated with part of speech tags – a preprocessing step required to enable the application of syntactic ﬁlters. To avoid excessive growth of the graph size by adding all possible combinations of sequences consisting of more than one lexical unit (ngrams), we consider only single words as candidates for addition to the graph, with multi-word keywords being eventually reconstructed in the post-processing phase.

Next, all lexical units that pass the syntactic ﬁlter are added to the graph, and an edge is added between those lexical units that co-occur within a window of N words. After the graph is constructed (undirected unweighted graph), the score associated with each vertex is set to an initial value of 1, and the ranking algorithm described in section 2 is run on the graph for several iterations until it converges – usually for 20-30 iterations, at a threshold of 0.0001.

Once a ﬁnal score is obtained for each vertex in the graph, vertices are sorted in reversed order of their score, and the top T vertices in the ranking are retained for post-processing. While T may be set to any ﬁxed value, usually ranging from 5 to 20 keywords (e.g. (Turney, 1999) limits the number of keywords extracted with his GenEx system to ﬁve), we are using a more ﬂexible approach, which decides

the number of keywords based on the size of the text. For the data used in our experiments, which consists of relatively short abstracts, T is set to a third of the number of vertices in the graph.

During post-processing, all lexical units selected as potential keywords by the TextRank algorithm are marked in the text, and sequences of adjacent keywords are collapsed into a multi-word keyword. For instance, in the text Matlab code for plotting ambiguity functions, if both Matlab and code are selected as potential keywords by TextRank, since they are adjacent, they are collapsed into one single keyword Matlab code.

Figure 2 shows a sample graph built for an abstract from our test collection. While the size of the abstracts ranges from 50 to 350 words, with an average size of 120 words, we have deliberately selected a very small abstract for the purpose of illustration. For this example, the lexical units found to have higher “importance” by the TextRank algorithm are (with the TextRank score indicated in parenthesis): numbers (1.46), inequations (1.45), linear (1.29), diophantine (1.28), upper (0.99), bounds (0.99), strict (0.77). Notice that this ranking is different than the one rendered by simple word frequencies. For the same text, a frequency approach provides the following top-ranked lexical units: systems (4), types (3), solutions (3), minimal (3), linear (2), inequations (2), algorithms (2). All other lexical units have a frequency of 1, and therefore cannot be ranked, but only listed.

**Sentence Extraction**

The other TextRank application that we investigate consists of sentence extraction for automatic summarization. In a way, the problem of sentence extraction can be regarded as similar to keyword extraction, since both applications aim at identifying sequences that are more “representative” for the given text. In keyword extraction, the candidate text units consist of words or phrases, whereas in sentence extraction, we deal with entire sentences. TextRank turns out to be well suited for this type of applications, since it allows for a ranking over text units that is recursively computed based on information drawn from the entire text.

**TextRank for Sentence Extraction**

To apply TextRank, we ﬁrst need to build a graph associated with the text, where the graph vertices are representative for the units to be ranked. For the task of sentence extraction, the goal is to rank entire sentences, and therefore a vertex is added to the graph for each sentence in the text.

The co-occurrence relation used for keyword extraction cannot be applied here, since the text units in consideration are signiﬁcantly larger than one or few words, and “co-occurrence” is not a meaningful relation for such large contexts. Instead, we are deﬁning a different relation, which determines a connection between two sentences if there is a “similarity” relation between them, where “similarity” is measured as a function of their content overlap. Such a relation between two sentences can be seen as a process of “recommendation”: a sentence that addresses certain concepts in a text, gives the reader a “recommendation” to refer to other sentences in the text that address the same concepts, and therefore a link can be drawn between any two such sentences that share common content.

Other sentence similarity measures, such as string kernels, cosine similarity, longest common subsequence, etc. are also possible, and we are currently evaluating their impact on the summarization performance.

The resulting graph is highly connected, with a weight associated with each edge, indicating the strength of the connections established between various sentence pairs in the text. The text is therefore represented as a weighted graph, and consequently we are using the weighted graph-based ranking formula introduced in Section 2.2.

After the ranking algorithm is run on the graph, sentences are sorted in reversed order of their score, and the top ranked sentences are selected for inclusion in the summary.

Figure 3 shows a text sample, and the associated weighted graph constructed for this text. The ﬁgure also shows sample weights attached to the edges connected to vertex 94, and the ﬁnal TextRank score computed for each sentence. The sentences with the highest rank are selected for inclusion in the abstract. For this sample article, the sentences with id-s 9, 15, 16, 18 are extracted, resulting in a summary of about 100 words, which according to automatic evaluation measures, is ranked the second among summaries produced by 15 other systems (see Section 4.2 for evaluation methodology).

**Why TextRank Works**

Intuitively, TextRank works well because it does not only rely on the local context of a text unit (vertex), but rather it takes into account information recursively drawn from the entire text (graph).

Through the graphs it builds on texts, TextRank identiﬁes connections between various entities in a text, and implements the concept of recommendation. A text unit recommends other related text units, and the strength of the recommendation is recursively computed based on the importance of the units making the recommendation. For instance, in the keyphrase extraction application, co-occurring words recommend each other as important, and it is the common context that enables the identiﬁcation of connections between words in text. In the process of identifying important sentences in a text, a sentence recommends another sentence that addresses similar concepts as being useful for the overall understanding of the text. The sentences that are highly recommended by other sentences in the text are likely to be more informative for the given text, and will be therefore given a higher score. An analogy can be also drawn with PageRank’s “random surfer model”, where a user surfs the Web by following links from any given Web page. In the context of text modeling, TextRank implements what we refer to as “text surﬁng”, which relates to the concept of text cohesion (Halliday and Hasan, 1976): from a certain concept C in a text, we are likely to “follow” links to connected concepts – that is, concepts that have a relation with the current concept C (be that a lexical or semantic relation). This also relates to the “knitting” phenomenon (Hobbs, 1974): facts associated with words are shared in different parts of the discourse, and such relationships serve to “knit the discourse together”.

Through its iterative mechanism, TextRank goes beyond simple graph connectivity, and it is able to score text units based also on the “importance” of other text units they link to. The text units selected by TextRank for a given application are the ones most recommended by related text units in the text, with preference given to the recommendations made by most inﬂuential ones, i.e. the ones that are in turn highly recommended by other related units. The underlying hypothesis is that in a cohesive text fragment, related text units tend to form a “Web” of connections that approximates the model humans build about a given context in the process of discourse understanding.

**Conclusions**

In this paper, we introduced TextRank – a graphbased ranking model for text processing, and show how it can be successfully used for natural language applications. In particular, we proposed and evaluated two innovative unsupervised approaches for keyword and sentence extraction, and showed that the accuracy achieved by TextRank in these applications is competitive with that of previously proposed state-of-the-art algorithms. An important aspect of TextRank is that it does not require deep linguistic knowledge, nor domain or language speciﬁc annotated corpora, which makes it highly portable to other domains, genres, or languages.

附录4 文献翻译

以下是所摘录文献的中文翻译。

**摘要**

在本文中，我们介绍了TextRank –一种基于图的文本处理排名模型，并说明了该模型如何在自然语言应用程序中成功使用。特别是，我们提出了两种创新的无监督方法来提取关键字和句子，并表明所获得的结果与以前在已建立的基准上发表的结果相比具有优势。

**引入**

诸如Kleinberg的HITS算法（Kleinberg，1999）或Google的PageRank（Brin和Page，1998）之类的基于图的排名算法已成功地用于引文分析，社交网络以及万维网链接结构的分析中。 可以说，通过提供一种依靠Web架构师的集体知识而不是Web页面的单独内容分析的Web页面排名机制，可以将这些算法选为Web搜索技术领域中触发的范式转换的关键要素。 简而言之，基于图的排名算法是一种通过确定从整个图递归计算的全局信息，而不是仅依赖于局部顶点特定信息，来确定图内顶点重要性的一种方法。

将类似的思路应用于从自然语言文档中提取的词汇图或语义图，将得出基于图的排名模型，该模型可应用于多种自然语言处理应用程序，其中从整个文本中提取的知识用于本地化排名/选择决定。这种面向文本的排名方法可以应用于从自动提取关键短语到提取摘要和词义消歧的任务（Mihalcea等，2004）。

在本文中，我们为从自然语言文本中提取的图引入了基于TextRank图的排名模型。我们调查并评估了TextRank在两种语言处理任务（包括无监督的关键字和句子提取）的应用中的应用，并表明，通过TextRank获得的结果与在这些领域开发的最新系统相比具有竞争力。

**文本化为图模型**

为了使基于图的排名算法能够应用于自然语言文本，我们必须构建一个表示文本的图，并将具有有意义关系的单词或其他文本实体互连。取决于手头的应用，可以将各种大小和特征的文本单元添加为图形中的顶点，例如单词，搭配，整个句子或其他。类似地，是由应用程序决定用于在任意两个这样的顶点之间绘制连接的关系的类型，例如：词汇或语义关系，上下文重叠等。不管添加到图中的元素的类型和特征如何，基于图的排名算法在自然语言文本中的应用包括以下主要步骤：

1.确定最能定义手头任务的文本单元，并将其添加为图形中的顶点。

2.确定连接这些文本单元的关系，并使用这些关系在图形的顶点之间绘制边。边缘可以是有向的或无向的，加权的或未加权的。

3.迭代基于图的排名算法，直到收敛为止。

4.根据最终分数对顶点进行排序。使用附加到每个顶点的值进行排名/选择决策。

在下文中，我们调查和评估TextRank在两个涉及文本单元排名的自然语言处理任务中的应用：（1）关键字提取任务，包括选择代表给定文本的关键字短语； （2）句子提取任务，由识别文本中最“重要”的句子组成，可用于构建提取性摘要。

**关键字提取**

键字提取应用程序的任务是自动在文本中识别最能描述文档的一组术语。这样的关键字可能构成用于为文档集合建立自动索引的有用条目，可以用于对文本进行分类，或者可以用作给定文档的简要摘要。此外，用于文本中重要术语自动识别的系统可用于术语提取和领域专用词典的构建。

最简单的方法可能是使用频率标准来选择文档中的“重要”关键字。然而，通常发现该方法导致较差的结果，因此，探索了其他方法。当前，该领域的最新技术由有监督的学习方法代表，其中训练有素的系统可以根据词汇和句法特征识别文本中的关键字。这种方法最早是在（Turney，1999）中提出的，在该方法中，参数化的启发式规则与遗传算法结合在一起，成为一个用于词组提取的系统-GenEx-该系统可以自动识别文档中的关键字。在（Frank et al。，1999）中使用了一种不同的学习算法，其中将朴素贝叶斯（Naive Bayes）学习方案应用于文档集合，并且在与（Turney，1999）中使用的相同数据集上观察到了改进的结果。 Turney和Frank均未报告其系统的召回情况，而仅报告了精度：GenEx（Turney，1999年）对每个文档提取的五个关键词短语的准确度达到29.0％，而Kea（Frank等人，则达到18.3％）。 （1999年），每个文档有十五个关键词。

最近，（Hulth，2003）将词汇和句法特征相结合，将监督学习系统应用于从摘要中提取关键词的事实证明，与以前发表的结果相比，有了明显的改进。正如Hulth所建议的那样，从摘要中提取关键字比从全文中提取关键字更广泛，因为Internet上的许多文档都不能以全文形式获得，而只能以摘要形式获得。在她的工作中，Hulth使用了（Turney，1999）中提出的方法进行了实验，并采用了一种将语音信息的一部分整合到学习过程中的新方法，并表明通过向该术语添加语言知识，系统的准确性几乎翻了一番。表示。

在本节中，我们报告使用TextRank进行关键字提取的实验，并表明基于图形的排名模型在此问题上的表现优于最佳公开结果。与（Hulth，2003）相似，我们正在评估从摘要中提取关键字的算法，主要目的是为了与她用其关键字短语提取系统报告的结果进行直接比较。请注意，文本的大小不是我们的系统强加的限制，在全文本上应用TextRank可以预期得到类似的结果。

**TextRank用于关键字提取**

此应用程序的预期最终结果是一组代表给定自然语言文本的单词或短语。因此，要排序的单位是从文本中提取的一个或多个词汇单位的序列，它们代表添加到文本图中的顶点。可以在两个词汇单元之间定义的任何关系都是可以在两个这样的顶点之间添加的潜在有用的连接（边）。我们使用的是共现关系，由单词出现之间的距离控制：如果两个顶点在最大N个单词的窗口内同时出现，则两个顶点相连，其中N可以设置为2到10个单词。共现链接表达了句法元素之间的关系，并且类似于发现对单词义消歧任务有用的语义链接（Mihalcea等，2004），它们表示给定文本的衔接指标。

可以使用句法过滤器来限制添加到图中的顶点，句法过滤器仅选择语音的特定部分的词汇单元。例如，可以仅考虑名词和动词以添加到图中，从而仅基于可在名词和动词之间建立的关系来绘制潜在边缘。我们尝试了各种语法过滤器，包括：所有开放类单词，仅名词和动词等，其中仅针对名词和形容词观察到了最佳结果，如第3.2节所述。

TextRank关键字提取算法完全不受监督，其操作过程如下。首先，对文本进行标记，并使用部分语音标签进行注释-这是实现应用语法过滤器所需的预处理步骤。为了避免通过添加由一个以上词汇单位（ngrams）组成的序列的所有可能组合来避免图形大小过度增长，我们只考虑将单个单词作为添加到图形中的候选单词，并最终在帖子中重建多单词关键字处理阶段。

接下来，将所有通过语法过滤器的词汇单元添加到图中，并在N个单词的窗口内同时出现的那些词汇单元之间添加边。构建完图后（无向非加权图），将与每个顶点关联的分数设置为初始值1，并在图上运行第2节中描述的排名算法多次迭代，直到收敛为止（通常为20- 30次迭代，阈值为0.0001。

一旦获得了图形中每个顶点的最终分数，便以其分数的倒序对顶点进行排序，并保留排名中的前T个顶点以进行后处理。尽管T可以设置为任何固定值，通常为5到20个关键字（例如（Turney，1999）限制了用他的GenEx系统提取的关键字数量为5），但我们使用的是更灵活的方法，

基于文本大小的关键字数。对于我们的实验中使用的数据（由相对简短的摘要组成），将T设置为图中顶点数量的三分之一。

在后处理期间，在文本中标记了由TextRank算法选择为潜在关键字的所有词法单位，并将相邻关键字的序列折叠为一个多单词关键字。例如，在Matlab code for plotting ambiguity functions中，如果Matlab和code都被TextRank选择为潜在关键字，则由于它们是相邻的，因此它们会折叠为一个关键字Matlab code。

*实验文本*

*Compatibility of systems of linear constraints over the set of natural numbers. Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types.*

*结果*

*TextRank选中的关键字:*

*linear constraints; linear diophantine equations; natural numbers; nonstrict inequations; strict inequations; upper bounds*

*人工选中的关键字*

*linear constraints; linear diophantine equations; minimal generating sets; nonstrict inequations; set of natural numbers; strict inequations; upper bounds*

如上显示了为我们的测试集合中的摘要构建的示例。虽然摘要的大小从50到350个字不等，平均大小为120个字，但出于说明目的，我们特意选择了一个非常小的摘要。对于此示例，通过TextRank算法发现具有更高“重要性”的词汇单位是（括号中指出了TextRank得分）：数字（1.46），不等式（1.45），线性（1.29），双色子（1.28），上位（0.99），界限（0.99），严格（0.77）。请注意，该排名与简单单词频率呈现的排名不同。对于同一文本，频率方法提供以下排名最高的词汇单位：系统（4），类型（3），解决方案（3），最小（3），线性（2），不等式（2），算法（2 ）。所有其他词法单位的频率为1，因此无法排名，只能列出。

**摘要句提取**

我们研究的另一个TextRank应用程序包括用于自动摘要的句子提取。 从某种意义上说，句子提取的问题可以看作与关键字提取类似，因为这两个应用程序都旨在识别对于给定文本而言更具“代表性”的序列。 在关键字提取中，候选文本单元由单词或短语组成，而在句子提取中，我们处理整个句子。 事实证明，TextRank非常适合此类应用程序，因为它允许对文本单元进行排名，该排名是根据从整个文本中得出的信息进行递归计算的。

**TextRank用于摘要提取**

要应用TextRank，我们首先需要构建一个与文本关联的图形，其中图形顶点代表要排名的单位。对于句子提取的任务，目标是对整个句子进行排名，因此，为文本中的每个句子在图上添加了一个顶点。

关键字提取所使用的共现关系不能在这里应用，因为所考虑的文本单元明显大于一个或几个单词，并且“共现”对于如此大的上下文不是有意义的关系。相反，我们定义了一个不同的关系，如果两个句子之间存在“相似性”关系，则该关系将确定两个句子之间的联系，其中“相似性”是根据其内容重叠程度来衡量的。两个句子之间的这种关系可以看作是“建议”的过程：一个句子针对文本中的某些概念，为读者提供“建议”以引用文本中针对相同概念的其他句子，因此可以在共享共同内容的任何两个这样的句子之间绘制链接。

其他句子相似性度量（例如字符串核，余弦相似性，最长公共子序列等）也是可能的，并且我们目前正在评估它们对汇总性能的影响。

生成的图形高度关联，每个边的权重相关联，表明文本中各个句子对之间建立的连接强度。因此，文本表示为加权图，因此，我们使用第2.2节中介绍的基于加权图的排名公式。

在图上运行排序算法后，将句子按其得分的相反顺序排序，然后选择排名最高的句子以将其包括在摘要中。

*实验文本*

*3: BC-HurricaineGillbert,09-11 339*

*4: BC-HurricaineGillbert,0348*

*5: Hurricaine Gilbert heads toward Dominican Coast*

*6: By Ruddy Gonzalez*

*7: Associated Press Writer*

*8: Santo Domingo, Dominican Republic (AP)*

*9: Hurricaine Gilbert Swept towrd the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains, and high seas.*

*10: The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph.*

*11: "There is no need for alarm," Civil Defense Director Eugenio Cabral said in a television alert shortly after midnight Saturday.*

*12: Cabral said residents of the province of Barahona should closely follow Gilbert’s movement.*

*13: An estimated 100,000 people live in the province, including 70,000 in the city of Barahona, about 125 miles west of Santo Domingo.*

*14. Tropical storm Gilbert formed in the eastern Carribean and strenghtened into a hurricaine Saturday night.*

*15: The National Hurricaine Center in Miami reported its position at 2 a.m. Sunday at latitude*

*16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo. 16: The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm.*

*17. The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.*

*18: Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds, and up to 12 feet to Puerto Rico’s south coast.*

*19: There were no reports on casualties.*

*20: San Juan, on the north coast, had heavy rains and gusts Saturday, but they subsided during the night.*

*21: On Saturday, Hurricane Florence was downgraded to a tropical storm, and its remnants pushed inland from the U.S. Gulf Coast.*

*22: Residents returned home, happy to find little damage from 90 mph winds and sheets of rain.*

*23: Florence, the sixth named storm of the 1988 Atlantic storm season, was the second hurricane.*

*24: The first, Debby, reached minimal hurricane strength briefly before hitting the Mexican coast last month.*

*结果*

*TextRank给出的文本摘要*

*Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil De− fense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas. The National Hurricane Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo. The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm. Strong winds associated with Gilbert brought coastal flooding, strong southeast winds and up to 12 feet to Puerto Rico’s south coast.*

*人工摘要文本*

*Hurricane Gilbert is moving toward the Dominican Republic, where the residents of the south coast, especially the Barahona Province, have been alerted to prepare for* *heavy rains, and high wind and seas. Tropical storm Gilbert formed in the eastern* *Carribean and became a hurricane on Saturday night. By 2 a.m. Sunday it was about 200 miles southeast of Santo Domingo and moving westward at 15 mph with winds of 75 mph. Flooding is expected in Puerto Rico and in the Virgin Islands. The second hurricane of the season, Florence, is now over the southern United States and down− graded to a tropical storm.*

图3显示了一个文本样本，以及为此文本构造的相关加权图。该图还显示了连接到顶点94的边缘上附着的样本权重，以及为每个句子计算的最终TextRank分数。选择具有最高等级的句子以包括在摘要中。对于此样本文章，提取具有id-s 9、15、16、18的句子，从而得到大约100个单词的摘要，根据自动评估方法，该摘要在其他15个系统生成的摘要中排名第二（请参阅评估方法的第4.2节）。

**为什么TextRank有效**

直观地讲，TextRank很好地工作，因为它不仅依赖于文本单元（顶点）的本地上下文，而且还考虑了从整个文本（图形）中递归得出的信息。

通过基于文本的图形，TextRank可以识别文本中各个实体之间的连接，并实现推荐的概念。文本单元会推荐其他相关的文本单元，并根据做出推荐的单元的重要性来递归计算推荐的强度。例如，在关键短语提取应用程序中，共同出现的单词相互推荐很重要，而通用上下文可以识别文本中单词之间的连接。在识别文本中的重要句子的过程中，一个句子会推荐另一个句子，该句子针对相似的概念，有助于整体理解文本。文本中其他句子强烈推荐的句子对于给定的文本可能更具信息性，因此将获得更高的分数。也可以使用PageRank的“随机浏览者模型”来进行类比，在该模型中，用户可以通过跟踪任何给定网页上的链接来浏览网页。在文本建模的上下文中，TextRank实现了我们所谓的“文本冲浪”，它与文本衔接的概念有关（Halliday和Hasan，1976年）：从文本中的某个概念C来看，我们很可能会“遵循”链接到关联的概念-即与当前概念C有关联的概念（即词汇或语义关系）。这也与“编织”现象有关（Hobbs，1974）：与单词相关的事实在话语的不同部分共享，这种关系起到“将话语编织在一起”的作用。

通过其迭代机制，TextRank超越了简单的图形连接性，并且还能够基于链接到的其他文本单元的“重要性”对文本单元进行评分。 TextRank为给定应用程序选择的文本单元是文本中相关文本单元最推荐的文本单元，优先考虑由

最有影响力的，也就是其他相关部门强烈推荐的那些。潜在的假设是，在具有凝聚力的文本片段中，相关的文本单元倾向于形成连接的“网络”，该网络近似于人类在语篇理解过程中围绕给定上下文构建的模型。

**总结**

在本文中，我们介绍了TextRank –一种基于图的文本处理排名模型，并说明了如何将其成功用于自然语言应用程序。 特别是，我们提出并评估了两种创新的无监督方法来提取关键字和句子，并表明TextRank在这些应用程序中获得的准确性与先前提出的最新算法相比具有竞争力。 TextRank的一个重要方面是它不需要深入的语言知识，也不需要领域或语言注释的语料库，这使其可以高度移植到其他领域，体裁或语言中。

**致 谢**

感谢邱莉榕老师在毕设和论文撰写过程中给予的指导，感谢同学给予的帮助。本论文的完成离不开你们的鼓励和指点，谢谢！