基于概率主题模型的文本摘要自动生成技术研究

朱铁丹

2012 年 **6** 月

中图分类号：
UDC 分类号：

基于概率主题模型的文本摘要自动生成技术研究

作 者 姓 名　　朱铁丹
学 院 名 称　　计算机学院
指 导 教 师　　刘玉树　教授
答辩委员会主席　　汤志忠　教授
申请学位级别　　工学博士
学 科 专 业　　计算机应用技术
学位授予单位　　北京理工大学
论文答辩日期　　2012 年 6 月

Research on Automatic Text Summarization Technologies Based on Probalistic Topic Model

Candidate Name:	Zhu Tiedan
School or Department:	Computer Science & Technology
Faculty Mentor:	Prof. Liu Yushu
Chair, Thesis Committee:	Prof. Tang Zhizhong
Degree Applied:	Doctor of Philosophy
Major:	Techonology of Computer Application
Degree by:	Beijing Institute of Technology
The Date of Defence：	June, 2012

基于概率主题模型的文本摘要自动生成技术研究

北京理工大学

研究成果声明

本人郑重声明：所提交的学位论文是我本人在指导教师的指导下进行的研究工作获得的研究成果。尽我所知，文中除特别标注和致谢的地方外，学位论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京理工大学或其它教育机构的学位或证书所使用过的材料。与我一同工作的合作者对此研究工作所做的任何贡献均已在学位论文中作了明确的说明并表示了谢意。

特此申明。

签　　名：　　日期：

关于学位论文使用权的说明

本人完全了解北京理工大学有关保管、使用学位论文的规定，其中包括：①学校有权保管、并向有关部门送交学位论文的原件与复印件；②学校可以采用影印、缩印或其它复制手段复制并保存学位论文；③学校可允许学位论文被查阅或借阅；④学校可以学术交流为目的，复制赠送和交换学位论文；⑤学校可以公布学位论文的全部或部分内容（保密学位论文在解密后遵守此规定）。

签　　名：　　日期：

导师签名：　　日期：

摘　　要

当前正处于一个信息爆炸的时代。如何提高阅读的效率，从而快速地从海量的资源中获取有用的信息，成为一个迫在眉睫的问题。自动文本摘要技术是解决这一问题的有力工具，有着广泛的应用前景。本文围绕文本摘要自动生成过程中的文本单元相似性度量、文摘句自动抽取、文摘句排序、自动文本摘要的评价等关键技术展开了一系列研究，主要取得了以下几个方面的成果：

（1）研究了文本单元的相似性度量问题。提出了基于概率主题模型的相似性度量方法 LDASim。该方法通过潜在狄利克雷分配对文档集进行建模，构建潜在主题空间，把不同粒度的文本单元映射为同一潜在主题空间中的向量。与传统的相似性度量方法相比，LDASim 可以把握文本单元中的深层语义关系，有效解决向量稀疏问题，并能够对各种不同粒度的文本单元进行统一的度量。通过实际的应用验证了 LDASim 的有效性，为相关的研究工作奠定了基础。

（2）研究了概率主题模型下文摘句的抽取问题。提出了基于差分进化和概率主题模型的 DEPTM 句子抽取算法。该算法把基于单词统计的表层距离和基于潜在主题的语义距离相结合，通过差分进化的方法对句子进行聚类主题划分；通过静态权重与动态权重相结合的逐句递减规则，保证了文摘与原文内容的整体相似性。实验证明该算法的效果好于 SumBasic 系统和 NGD 方法以及 DUC2006 会议的参会系统。

（3）研究了文摘句之间的多种评判标准与排列顺序之间的关系。从本位相似度和语境相似度两方面来刻画句子间的相似关系，提出了基于相关距离的文摘句排序算法 SOCD；从原文档集中学习句子间的前置概率和后置概率关系，提出了基于原文顺序概率的文摘句排序算法；融合四种顺序标准，提出了基于概率主题模型的层次性文摘句排序算法 BSOP。这三种算法从不同的角度弥补了传统算法中由于顺序标准的片面性而带来的不足。实验结果表明，SOCD 算法和 SOSOP 算法优于传统的 MO 和 CO 算法，BSOP 算法优于 AGL 算法。

（4）研究了自动文本摘要的评价方法。基于概率主题模型下的潜在主题向量间的相似度，本文提出了 LS 文本摘要自动评价方法。该方法综合了文档相关的 LS-T、句子相关的 LS-S、句子对应的 LS-M 三种评测；利用这三种评测定义了 LS 可读性评分、LS 无冗余性评分、LS 全面性评分；将三种评分结合在一起得到最终评分 LS-Score。

实验表明 LS 评价方法效果稳定，适用范围广泛，优于目前普遍采用的 ROUGH 和 Pyramids 等评价方法。

关键词：自动文本摘要；概率主题模型；潜在狄利克雷分配；文摘句抽取；文摘句排序；自动文摘的评价；相似性度量

Abstract

At present，we are living in an age of information explosion. We are facing an extreme problem, how to improve the efficiency of reading so that we can grasp the useful information quickly from the huge mass resources. The Automatic Text Summarization technology is a powerful tool to solve the problem and has wide application prospect. According to the characteristic of Automatic Text Summarization, this paper does a series research which resolve similarity measure of text unit, sentence extraction, sentence ordering for summarization, automatic evaluation to summarization and so on. The main innovative achievements are as follows:

(1) The similarity measure for text unit has been researched. The similarity mearsure method LDASim, which is based on the Probabilistic topic model, is proposed. Through the Latent Dirichlet Allocation, the corpus is modeled and the space of latent topic is created. Text units of different granularity are mapped to vectors in a same space of latent topic. Compared to the traditional similarity methods, LDASim can solve the problem of sparseness and can measure the similarity between text units of different granularity. The practicability of LDASim is proved in the experiments, which provides the useful information for further research.

(2) The sentence extraction problem based on probabilistic topic model has been researched. A sentence extraction algorithm based on Differential Evolution and probabilistic topic model (DEPTM) is proposed. It combines the surface distance based on words statistics and the semantic distance based on latent topic. Sentences are clustered through a Differential Evolution method. Combining the static weight and dynamic weight of sentences, aiming at the similarity between the summary and the corpus, through eliminating the useless sentences progressively, the similarity between summary and the corpus is ensured. The experiment shows that DEPTM is better than SumBasic system, NGD method and systems in DUC2006.

(3) Diverse relations between sentences are researched. Combing the isolated

similarity and context similarity, the Sentence Ordering algorithm based on Correlative Distance (SOCD) is proposed. Obtaining the preposition probability and postposition probability of sentences from corpus, the Sentence Ordering algorithm based on Source Ordering Probability (SOSOP) is proposed. Integrating four criterions by a support vector machine, a Bottom-up Sentence Ordering algorithm based on Probabilistic topic model (BSOP) is proposed. The algorithms proposed here can amend the traditional algorithm of unilateral criterion. The experiment shows that SOCD and SOSOP are better than MO or CO algorithms, BSOP is better than AGL alrogithm.

(4) The evaluation for Automatic Text Summarization has been researched. Based on the similarity in the latent topic space of probabilistic topic model, an evaluation named LS for text summarization is proposed. LS contains three evaluating: document-level method LS-T, sentence-level method named LS-S, sentence-metch method named LS-M. Based on the three evaluating, LS-Readability score, LS-Irredundant score and LS-Total score are defined. The final LS-Score is calculated through the three scores above. The experiment shows that LS evaluation has steady effect. It fit for different situations and outperforms ROUGH or Pyramids.

Key Words: Automatic Text Summarization; Probabilistic Topic Model; Latent Dirichlet Allocation; Sentence Extraction; Sentence Ordering; Evaluation for Text Summary; Similarity Measure

目　　录

图索引

表索引

第1章 绪论

随着科学技术的不断进步，Internet 的迅猛发展，如今的世界可以说是一个信息大爆炸的时代。无限的信息与人类有限的承载能力之间的矛盾日益突出。文本摘要便是人们快速获取文本信息的一种有效方式。手工编写的摘要固然精美准确，但是其编写费时费力，无法满足人们迅速增长的信息获取需求。因此，人们越来越渴望能用计算机来代替人工，自动的生成人们所需要的摘要。于是自动文本摘要技术应运而生。

1.1 自动文本摘要技术简介

1.1.1 自动文本摘要概述

文本摘要通常被定义为——“摘要是一篇文本，它从一篇或多篇源文本中产生，总结了原文本中的重要信息，而其长度却不及源文本的一半，通常要短很多。”[1]这个简单的定义，给出了摘要的三个特点：

（1）可读性。摘要本身是一篇文本，可以供人们阅读理解。

（2）概括性。摘要概括了源文本的主要内容。

（3）压缩性。摘要的长度很短。

自动文本摘要（Automatic Text Summarization，ATS）是自然语言处理技术（Nature Language Processing，NLP）的一个重要的子领域。所谓自动文本摘要就是指：由计算机自动从原文本中提取内容，采用压缩的形式，将最主要的内容呈现给用户[5]。

目前，自动文本摘要技术主要分两种类型：抽取型自动文摘（extraction）和理解型自动文摘（abstraction）。

抽取型自动文摘，就是从原文本中找出最能概括原文内容的关键句子，抽取出这些关键句子来组成文摘。通常不需要很多的语法语义分析，而是采用统计学的方法。通过统计如单词出现频率、句子位置等表面特征，来给句子评分，找出评分较高的句子组成文摘。

理解型自动文摘，又称为自写型文摘，则是要在充分理解原文本主要内容的基础

上，由计算机生成文摘句。这种方法难度较大，需要深入分析文本的语法语义，对自然语言理解技术的发展有较高的依赖。

二者最主要的区别在于：抽取型文摘中，句子来自于原文；理解型文摘中，句子由计算机生成。所以抽取型文摘，其研究重点是对句子的评价，找出最能表现原文内容的句子；而理解型文摘，其研究重点是语言的表达形式，即如何生成符合语法的句子。

Luhn 在 1958 年的论文中提出了自动文本摘要的问题[2]，通常被认为是自动文摘研究的始祖。当时主要采用的都是抽取型的自动文摘。在上世纪七十年代到九十年代，自然语言理解技术有了较大的发展。这时人们对自动文摘的研究逐渐转移到理解型文摘，并在一些特定领域内取得了一定的成果。上世纪九十年代以后，自然语言理解技术的发展相对缓慢，而统计语言学又逐渐兴起。自动文摘技术的研究又重新回归到以抽取型文摘为主的道路上来。近年来自动文摘技术取得了较为丰硕的成果，大多是抽取型文摘。

根据文摘处理的文本对象不同划分，自动文摘可以分为：单文档文摘（single-document summarization）和多文档文摘（multi-document summarization）。所谓单文档文摘，就是指处理的对象是一篇文档，一篇文档生成一篇文摘。而多文档文摘，是指处理的对象不仅仅是一篇文档，而是由多篇文档构成的文档集，将文档集中所有文档的内容综合概括成一篇文摘。

早期的自动文摘技术研究，都是基于单文档的。近年来随着网络上文本信息量的激增，多文档文摘成为了人们的研究热点。多文档文摘与单文档文摘既有联系又有区别。多文档文摘可以利用很多单文档文摘中的成熟技术，但多文档文摘又有其自身的特点，不是单文档文摘技术的简单重复应用。

1.1.2 自动文本摘要技术的发展与现状

早期的单文档自动文摘技术研究多集中于技术性的文本。Luhn 最早于 1958 年提出自动文摘的问题。Luhn 认为一个单词在文档中出现的次数（即词频）是其在该文档中重要性的体现[2]。Luhn 首先将所有的单词进行词根还原，并去掉停用词（stop words）。然后统计每个单词的出现次数，并按照降序排列。接下来根据句子中所包含的单词来计算每个句子的重要程度，将句子按照重要程度排序，取排在最前面的几个

句子构成摘要。

除了词频之外，句子位置也是一项重要的句子特征。Baxendale 研究了 200 段文字[3]，发现在其中约 85%的段落中，第一句就是该段落的主题句，而 7%的段落中，最后一句是主题句。因此，当需要选择一个段落的主题句时，选择首句或末句通常都能得到不错的结果。这种方法简单而有效，在之后的各种自动文摘系统中有广泛的应用。

Edmundson 等于 1969 年提出了一个自动文摘方法[4]。除了词频和句子位置以外，又增加了两项特征：线索词和标题。将四个特征赋予不同的权值来对句子进行评分。实验表明，约有 44%的自动抽取结果与人工抽取结果相符。

上世纪七十年代，自动文摘系统的代表是 Bush 等人开发的 ADAM 系统[6]，借助多种规则判断句子是否应该成为文摘句。八十年代末到九十年代初，美国 GE 公司 Rau 等人开发的 SCISOR 系统[7]。SCISOR 系统采用的是理解型文摘，只针对特定的新闻领域。

Ono 等人于 1994 年提出了一个用于日语解释性著作的计算模型[19]，详细说明了抽取修辞结构的过程，用二叉树便是句子块之间的关系。这个结构使用一系列的自然语言理解步骤进行抽取：句子分析、修辞关系抽取、分段、候选生成、偏好判断。评价是基于修辞关系的相关重要性。在下面的步骤中，修辞结构树的节点被修剪以减少句子同时保持其重要部分。对段落进行同样的处理，最终生成摘要。使用 30 篇日语新闻文章进行测试评价，关键句的覆盖率达 51%，最重要关键句的覆盖率达 74%。

Kupiec 于 1995 年提出了采用贝叶斯分类器的自动文摘方法[8]。通过贝叶斯分类器，将原文本中的句子分成两类：值得抽取和不值得抽取。采用的特征，在 Edmundson[4] 的四项特征基础上，又增加了两项：句子长度和大写字母的出现。Aone 等于 1999 年也提出了一个利用贝叶斯分类器的自动文摘系统，称为 DimSum[9]。其中使用了词频（term frequency，tf）和反向文档频率（inverse document frequency，idf）来确定特征词。除了单个单词之外，还统计了相关联的名词对。同时，同义词、缩写名称等也进行了简单的处理。

Barzilay 等人于 1995 年使用一系列的语言学分析来完成自动文摘任务[17]。首先定义了语汇链（lexical chain），即文本中相关单词的序列。作者的方法是先将文本分段，然后找出语汇链，使用强语汇链来鉴别值得抽取的句子。他们提出了内聚性的概念，表示文本的不同部分之间的相关程度。内聚性不仅存在于单词之间，也存在于单词序

列之间，这样就形成了语汇链。作者把语汇链作为文摘表示的基本来源。发现语汇链主要有三个步骤：（1）选择候选单词集。（2）对每一个候选单词，根据相关准则找到适当的链。（3）如果找到，把该单词插入链中。其中相关准则采用 Wordnet 中单词的距离来衡量。简单名词和复合名词被用来作为寻找候选单词的起点。最后用强语汇链来创建摘要。

Lin 和 Hovy 于 1997 年专门对句子位置这一重要特征进行了研究[10]。提出了仅使用句子位置这一项特征的方法，作者称之为“位置方法”（position method）。根据文章的篇章结构，出现在特殊位置的句子（如摘要、各级标题等）更有可能是文章的主题句。针对不同类型的文章，做了不同的调整，以使位置方法达到最佳效果。

Marcu 于 1998 年描述了一个独特的自动文摘方法[21]，与之前的工作不同，并不假设文档中的句子组成一个序列。他使用了带有传统自动文摘特征的基于讲述的启发式算法。使用的度量标准有：基于聚类的标准、基于标记的标准、修辞聚类技术、基于形状的标准、基于标题的标准、基于位置的标准、基于连通性的标准。把这些评分做一个加权的线性组合。为了找到最佳组合，作者计算了训练集的最大化 F 值权重。使用了类似 GSAT[22]的算法，在七维空间里进行贪心搜索。在他的结果中，最好的 F 值达到 75.42%，比基线高 3.5%。

1999 年，Lin 又尝试用决策树代替贝叶斯分类器进行自动文摘抽取[11]。他研究了许多特征以及它们对句子抽取的影响。比较新颖的特征的有查询特征（句子包含的查询词越多，评分越高）、信息检索特征（IR signature，语料库中 m 个最突出的特征）、数字数据（包含数字的句子被赋予布尔权值 1）、恰当名称（proper name，包含恰当名称的句子被赋予布尔权值 1）、代词形容词（包含代词或形容词的句子被赋予布尔权值 1）、星期、月份、引用等等。作者用多种基线做了实验，比如仅使用位置特征、多项特征的简单加和等等。在比较机器抽取文摘与手工抽取文摘的相似程度时，决策树的方法在整体的数据集上具有明显的优势。但在三个主题中，简单的特征组合效果更好。Lin 推测这是由于某些特征之间相互独立造成的。

Couroy 和 O’leary 于 2001 年提出了基于隐马尔科夫模型（hidden Markov model，HMM）的自动文摘方法[12]。之所以采用序变模型，是为了计算句子之间的本地独立性（local dependencies）。仅使用了三个特征：句子位置、句子包含的单词数、句子单词与文档单词的相似性。隐马尔科夫模型构建如下：它包含 2s+1 个状态，其中有 s 个“摘要状态”和 s+1 个“非摘要状态”，交替排列；非摘要状态可以不进行跳转，

而摘要状态则允许跳转到下一个摘要状态。使用训练集对该模型进行训练，计算每一个跳转概率的极大似然估计值，从而组成跳转概率矩阵。实验表明此种方法具有一定的可行性。

Osborne 认为现有的方法都假设了特征之间相互独立，于是于 2002 提出了 log-linear 模型[13]，以消除独立性假设的影响。设 c 是一个标签，s 是要标注的对象，f_i 是第 i 个特征，λ_i 是特征 f_i 的权重，那么 Osborne 使用的 log-linear 模型可以用如下公式表示：

$$P(c \mid s) = \frac{1}{Z(s)} \exp(\sum_i \lambda_i f_i(c,s)) \qquad (1\text{-}1)$$

其中 $Z(s) = \sum_c \exp(\sum_i \lambda_i f_i(c,s))$。在实际应用中，只有两个标签：摘要句和非摘要句。对应的权值通过共轭梯度下降法训练得到。经过实验证明，log-linear 模型的效果好于朴素贝叶斯分类器模型。

在 2001 年 2 月的 DUC 会议上，有一项任务是对一篇新闻生成一个 100 字的摘要。当时设定了一个基线方法，就是直接取新闻稿的前 n 个句子构成文摘。令人惊讶的是，参会的所有系统，效果均不及此基线。Nenkova[14]对这一基线进行了深入的分析。这一令人惊讶的结果与新闻业的惯例有关——通常新闻稿都把最重要的内容放在最前面。2002 年以后，DUC 会议就取消了单篇新闻的摘要任务。

Svore 于 2007 年提出了一个基于神经网络的算法[15]，并使用了第三方数据集。Svore 从文章的每个句子的标签和特征中训练了一个模型，以便给测试文档中的句子评级。评级通过 RankNet[16]完成。作者设计了一个基于对的神经网络算法并使用梯度下降法进行训练。在训练过程中使用了软标记，而不是像以往的方法那样使用硬标记：文摘句或非文摘句。

多文档文摘，即从多篇文档中抽取出一篇摘要。从 90 年代开始，这一研究工作得到越来越受到重视，甚至已经推广到了多语言领域[33][34]。很多研究成果都是针对新闻类文档集。互联网上的一些新闻聚类系统为这项研究工作提供了数据基础，例如 Google News，Columbia NewsBlaster，News In Essence 等。多文档文摘和单文档文摘的主要区别，就在于信息来源于多篇文档，这些文档之间即有重复又有补充，甚至偶尔还有矛盾。因此多文档文摘任务中，不仅仅要鉴别文档的主要内容，还要去除冗余，而且要发现矛盾点以保证整篇摘要的一致性和完整性。

哥伦比亚大学的 NLP 小组于 1995 年开发了一个名为 SUMMONS 的自动文摘系

统[18]是一个将已有技术扩展而成的模板驱动信息理解系统。开始时，多文档文摘需要大量的语言理解和生成技术。随着众多研究人员的加入和统计语言学的不断发展，句子抽取成为了多文档文摘的主流。

衡量两个句子之间的相似程度，这一技术在多文档文摘的抽取中有着广泛的应用。有人使用这种相似性度量进行聚类，通过聚类对文档进行主题划分[24]，或从聚类里生成复合句[25]。有些方法则通过对句子相似性的判断，不断的将含有新内容的句子加入文摘中，称为最大边际相关性（maximal marginal relevance）[26]。

McKeown 等人曾经声称，抽取技术对多文档文摘是无效的[18][24]。几年之后，这个说法就被推翻，基于抽取技术的自动文摘系统 MEAD[27]诞生，在新闻类的多文档文摘问题中有出色的表现。像 SUMMONS 这样的系统，应用于专业领域，目标是生成一种简介，以展示最新的新闻报道的区别和进展，其重心是放在信息的表达方式上，如何组织语言把信息恰当的传递给用户。而像 MEAD 这样的系统，是对所有领域都通用的，其所关心的重点是内容而不是形式。因此前一种系统需要在语言生成技术上下功夫，以产生符合语法的连贯一致的文摘。而后一种系统则更像一种信息检索系统，从多篇文档中找出重要的信息。

SUMMONS 是第一个多文档文摘系统。它处理的是窄领域的单一事件：关于恐怖主义的新闻。把针对一个事件的不同新闻机构的不同时间的报道，整合成一篇相关信息的简介。SUMMONS 并不是直接阅读原始文本，而是阅读一个数据库，该数据库是由基于模板的信息理解系统事先构建好的。一个完整的多文档文摘系统应该由两个子系统连接而成的：一个是文本处理系统，从原始文本中抽取信息，填写模板槽，把原始的自然语言文本转化成结构化的信息；二是摘要生成系统，从结构化的信息当中生成摘要。SUMMONS 的结构就包含这样的两个主要部件：一个内容规划器，通过输入模板的组合来选择应该包含在摘要中的信息；另一个是语言生成器，选择恰当的单词用符合语法的形式表达信息内容。其中后者是采用已有的语言生成工具 FUF/SURGE 系统开发的。内容规划器，是通过摘要操作符生成的，也就是一系列的启发式的规则。其中一些操作需要解决冲突，也就是不同来源之间相互矛盾的信息。

上面这个框架只能适用于特定的领域，因为领域范围越窄，就越有可能设计出合适的模板。接下来的问题是如何拓宽领域，这一工作到 1999 年有了进展[24,25]。这时输入可以使一个相关的文档集，比如搜索引擎中响应一个查询而得到的搜索结果。系统从鉴定主题开始，所谓主题就是相似文本单元的集合。这里的文本单元，通常是自

然段落。这个工作可以看做是一个聚类问题。为了计算文本单元之间的相似程度，可以把这些文本单元映射成由特征组成的向量。特征主要包括 TF-IDF 权重、名词短语、特殊名词、从 Wordnet 得到的同义词集、动词的语义类集等。对每一对文本单元，计算一个向量以表示不同特征的匹配程度。从数据中学习决策规则，用来对文本单元进行分类：相似或不相似。把相似的文本单元归入同一主题。

一旦主题划分好了，系统就进入第二阶段：信息融合（information fusion）。这一阶段的目标是决定每个主题里哪些句子应该被选为文摘句。作者并没有仅仅挑选一个句子作为主题的代表，而是提出了一个算法，比较和交叉每个主题中的短语的谓词参数结构，选择重复率最高的加入文摘中。通过 Collin[28]的统计语法分析器，把句子转换成依赖树，捕捉谓词参数结构，确定功能角色，去掉区别词和辅助词。

比较算法开始递归地遍历这些依赖树，把相同的节点加入输出树。一旦找到完整的短语，就被标记为包含于文摘。如果两个短语，根在同一个节点，不同但是很相似，则假设它们其中一个是另一个的别称。一旦决定了摘要的内容（表示为谓词参数结构），则通过 FUF/SURGE 语言生成系统把这些结构转换成需要的参数，从而生成符合语法的文本摘要。

Mani 和 Bloedorn 于 1997 年描述了一个自动文摘的信息抽取框架[30]，使用基于图的方法来发现文档对之间的相似性和非相似性。虽然并没有最终生成文本形式的摘要，但是摘要的主要内容通过实体概念和它们之间的关系表示出来了。实体概念和它们之间的关系是通过图的点和边呈现的。与以往不同，他们并不是抽取句子，而是通过“传播激活”[31]技术来探测图的显著区域。

这个方法与前面提到的主题驱动的方法类似，有一个额外的输入来代表生成文摘的主题。主题是通过图中的入口节点集来表示的。一篇文档被表示成图的方式如下：一个节点表示一个单词的出现（也就是单词本身以及这个单词在文档中的位置），每个节点可以有几类链：邻接链（ADJ）连向文中向邻接的单词，相同链（SAME）连向同一个单词的另一次出现，阿尔法链（ALPHA）连向从 Wordnet 中抽出的意义关系编码，短语链（PHRASE）把属于同一短语的节点序列连在一起，名称链（NAME）连向互指的名称。

图建立起来之后，通过词根比较鉴别出主题节点，并将其设置为入口节点。接下来使用一个称为“传播激活”的过程对语义相关的文本进行搜索。入口节点的单词和短语根据它们的 TF-IDF 值进行初始化。邻接节点的权重依赖于节点的到达路径，随

着路径的距离呈指数级递减函数。在一个句子内部传播要比跨越句子边界便宜，跨越句子边界要比跨越段落边界便宜。给定一对文档的图，通过相同的词干或同义词来鉴定出一般节点，那些不同的节点就是非一般节点。对两篇文档中的每一个句子，要计算两个分值：一个反映了一般节点的出现，即计算这些节点的平均权重；另一个分值则是计算非一般节点的平均权重。两个分值都是在传播激活之后计算。最后，找出那些一般和非一般评分都高的句子，可以通过指定一般和非一般句子的最大数量来控制输出结果。在下一步的研究中，作者计划使用这些结构实际合成一篇理解型的摘要，而不仅仅是把找出文本中的片段。

Carbonell 和 Goldstein[26]于 1998 年介绍了最大边际关系（maximal marginal relevance，MMR），从而为主题驱动的自动文摘做出了重要贡献。其主要思想是把请求相关性和信息独特性结合起来。MMR 通过两种度量的线性组合，在奖励关联句子的同时也惩罚冗余。MMR 算法的一个吸引人之处在于其面向主题的特点，通过加入一个查询 Q，使得这一算法可以根据用户的需要生成文摘。正如作者所说，“信息需求不同的用户，即使面对同一篇文档，也会需要完全不同的文摘”。这一主张在之前的多文档自动文摘的研究中从未提及。

聚类技术在之前的方法中曾经使用过[24,25]，只是用于来划分主题。Radev 于 2000 年提出了以聚类质心为核心的自动文摘方法，该方法成为了 MEAD 系统的基础，作者在 2004 年的论文[32]中对该方法做了详细的论述。在这种方法中，它并没有使用任何的语言生成模块，所有的文档都被表示成词袋（bags of words）的形式。这样的系统非常容易扩展，与领域无关。

基于质心的方法的第一项任务是主题探测，目标是把描述同一事件的新闻分组。为了完成这一任务，使用了在文档的 TF-IDF 向量表示基础上的凝结式聚类算法，把文档成功加入聚类后重新根据下面的公式计算聚类质心：

$$c_j = \frac{\sum_{d \in C_j} \tilde{d}}{\left| C_j \right|} \qquad (1\text{-}2)$$

这里 c_j 是第 j 个聚类的质心，C_j 是属于第 j 个聚类的文档的集合，其基数是$|C_j|$。$\tilde{d}$ 是文档 d 的缩减版，去掉了 TF-IDF 值低于阈值的单词。质心可以被看做是假想的文档，仅包含组成聚类的文档中的 TF-IDF 值大于阈值的单词。每一个事件聚类都是一个新闻文档的集合，从不同来源得到的新闻，按时间顺序排列，描述一个事件随时间的变化。

基于质心的方法的第二项任务是使用质心来鉴别核心句，即每个聚类中的更为接近整个聚类主题中心的句子。作者定义了两个矩阵，与前面提到的 MMR 类似：基于聚类的相关效用（cluster-based relative utility，CBRU）和交叉句信息类别（cross-sentence informational subsumption，CSIS）。第一个矩阵负责衡量一个句子与整个聚类主题的相似程度；第二个则是句子中冗余程度的度量。与 MMR 不同的是，这些矩阵是不依赖于请求的。给定一个聚类 C，其中的文档被分割为 n 个句子。有压缩率 R，从源文档中抽取出 nR 个句子，按时间顺序组成一个序列。句子的选择就是根据 CBRU 和 CSIS 两个矩阵进行。对每一个句子 S_i，使用三个不同的特征：

（1）质心值（C_i），定义为句子中所有单词的质心值之和。

（2）位置值（P_i），令 Cmax 为文档中评分最高的句子的质心值，则

$$P_i = \frac{n-i+1}{n} C_{\max} \tag{1-3}$$

（3）首句覆盖率（F_i），定义为句子 i 的单词向量与文档中首句的单词向量之间的内积。

最终每个句子的评分是上面三个分值的组合再减去冗余惩罚（Rs），即每个句子与评分最高的句子之间的重复程度。

2011 年，William 等人提出了名为 PathSum 自动文摘处理框架[29]。该框架采用了潜在狄利克雷分配（Latent Dirichlet Allocation，LDA）和中国餐馆问题（Chinese Restaurant Process，CRP）相结合的层次潜在狄利克雷分配模型 nLDA。这是 LDA 在自动文摘领域的成功应用，也证实了 LDA 模型在自动文摘中应用的可行性。因此本文也将采用 LDA 模型对自动文摘技术进行研究。

1.2 文本模型

要研究文本摘要，首先要用恰当的数学模型来表示文本。目前对文本的研究中，最主要模型是向量空间模型，以其简单有效的特点成为文本处理中最流行的模型。而近些年来，概率主题模型以其良好的统计特性，越来越受到研究者的青睐。

1.2.1 向量空间模型

向量空间模型（Vector Space Model，VSM）[50]是由 Gerard Salton 提出的，是目

前为止最为常见的一种用于文本表示的数学模型。该模型在很多智能信息处理领域，如文本自动分类、信息检索、自动摘要等，都有着十分广泛的应用，是目前的文本表示模型中最简便有效的方法之一[51][52]。向量空间模型，就是采用多维向量来表示文本单元，通过计算向量间的相似度实现文本单元之间相似度的计算。

向量空间模型中的基本单位是特征词（term）。特征词是指具有独立意义且不可再分的语言单位。特征词可以是一般的单词，也可以是词组、短语等。特征词是使用向量空间模型来分析文本的基础，通常一个文本可以表示成特征词的集合，即 $D=\{t_1, t_2, \ldots, t_n\}$。

不同的特征词在文档中的地位是不一样的，有的特征词重要，有的则相对次要一些。给每一个特征词 t_k 都赋予一个权值 w_k，以表示其在文档中的重要程度。这样文档就可以表示成为 $D=\{t_1,w_1;t_2,w_2;\ldots;t_n,w_n\}$。

对于给定文档 $D=\{t_1, t_2, \ldots, t_n\}$，$t_i \neq t_j$，不考虑每个特征词 t_k 在文档中的顺序，这时可以把 t_1，t_2，……，t_n 看成一个 n 维的坐标系，而 $w_1,w_2,\ldots,w_n$ 则为相应维度上的取值，因而可以将 $D=\{t_1,w_1;t_2,w_2;\ldots;t_n,w_n\}$，简记为 $D=\{w_1,w_2,\ldots,w_n\}$。在此形式下，一个文档就可以看成是 n 维空间中的一个向量，称 $D=(w_1,w_2,\cdots\cdots,w_n)$为文档 D 的向量表示或向量空间模型。相应的，句子、段落、文档集等文本单元也都可以用相同的方式表示成空间中的向量形式。

文档向量的三维空间模型表示如图 1.1。

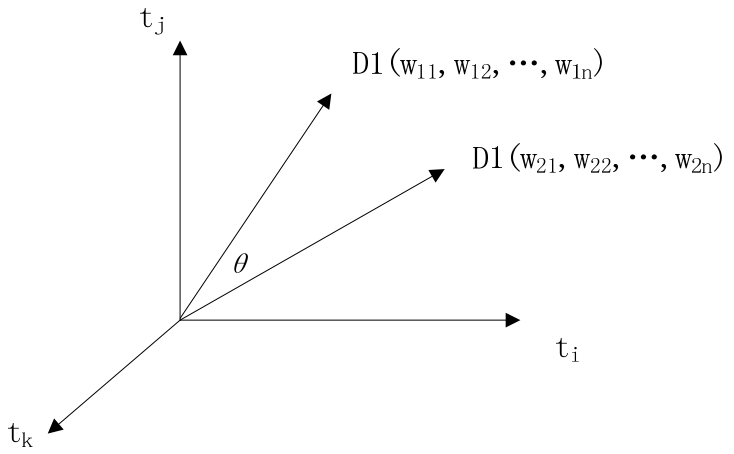

图 1.1 文档的向量空间模型表示

用向量空间模型来处理文本，需要考虑两个关键技术：特征词的选取和权值的计算。此外，由于句子中包含的特征词太少，在高维度的向量空间中，会造成表示句子

的向量极其稀疏，这会给句子之间的相似度度量造成很大的困难。因此，如何降维也是向量空间模型中的研究热点。

1.2.1.1 特征词的选取粒度

向量空间模型是以特征词为单位来表示文本的，所以特征词的选取对向量空间模型是极为重要的。特征词要能够表示出文本的内容特征。最简单的方法，也是最常见的方法，就是直接用文本中出现的单词作为特征词[54][55]。然而对于不同的书写语言而言，单词的确定方式也不同[56]。有些语言，比如英语，单词于单词之间有空格，这样很容易确定单词；但是有些语言，比如汉语，词与词之间是没有空格，这就需要进行词法分析（morphological analysis）来确定单词[61]。

直接使用单词作为特征词的方法显然过于简单，也忽略了单词与单词之间的相互关系，因此有很多研究工作都致力于寻找更加有效的文本表示方法。其中有相当一部分研究都是通过语法分析技术生成的索引短语来表示文本的[57][58]。Lewis 根据他自己的研究认为，采用这种短语来表示文本的方法虽然包含了很好的语义特性，但由于具有较差的统计特性，在一定程度上抵消了其优点[53]。Lewis 又试图通过对短语特征进行聚类的办法来解决这个问题[59]，但并没有什么明显的效果。他认为主要原因是数据集合不够大，许多短语出现的次数过少。Scott 也做了一些试验来分析各种表示方法的优缺点[60]。他的试验结果表明，那些基于语言学的方法、利用了语义和语法知识的方法，虽不比最基本的单词表示方法差，但也没有好多少。人们经过分析认为，造成这种现象的原因，很大程度上是由于，在自然语言中，一个词的语义不是固定不变的，而是根据环境的不同而不同，在不同的上下文语境中有着不同的含义。因此到目前为止，直接使用单词作为特征词，仍旧是最有效的特征词选择方式。

1.2.1.2 权值的确定

同一篇文档中的多个特征词，其重要程度一般都不是均匀的。也就是说有的特征词相对重要，与文档主题的相关程度高，而有些特征词的重要性则不高，与文档主题的相关程度低。因此，对不同的特征词赋予不同的权重以反映其重要性，这是很有必要的。给特征词赋予权重的技术，成为特征词加权（term weighting）[73]技术。

假设有文本集 C 含有 n 个文本，即 C={D_1,D_2,…,Dn}，从文本集 C 中共提取了 m

个特征词，记为 $t_1,t_2,\cdots,t_m$。设在文本 D_j 中特征词 t_i 的权重为 w_{ij}，则 w_{ij} 可以按以下三项指标进行加权：

特征词 t_i 相对于一篇文本 D_j 的权值，称为局部权重（local weight），记为 l_{ij}。通常按照特征词 t_i 在文本 D_j 中出现的次数来计算。也就是说，在如果特征词 t_i 在文本 D_j 中出现的频率越高，则获得的权值越大，表明该特征词对该文本越重要。

特征词 t_i 相对于整个文档集 C 的权值，称为全局权重（global weight），记为 g_i。全局权重是按照特征词 t_i 在整个文档集中的分布来确定的，又称为特征词的逆文档频率（inverse document frequency，idf）。特征词 t_i 在全局中出现的越均匀，则获得的权值越小，说明 t_i 对不同文档的区分度越小。

不同的文本长度不同，越长的文本包含的特征词也越多，每个特征词出现的次数也越多。因此，在较长的文本中特征词会获得较高的分数。为了消除文本长度的影响，引入文本规范化系数（document normalization factor），记为 n_j。

这样，特征词的权重就可以用上述三个指标来计算，即：

$$w_{ij} = \frac{l_{ij} g_i}{n_j} \tag{1-4}$$

设文档集 C 中的文档总数为 n，n_i 为文档集中包含特征词 t_i 的文档数量，特征词 t_i 在文档 D_j 中出现的频率记为 $freq_{ij}$，文档 D_j 中所有的特征词出现频率的最大值记为 $\max tf_j$。则局部权重和全局权重可以计算为：

$$l_{ij} = \frac{freq_{ij}}{\max tf_j} \tag{1-5}$$

$$g_i = \log(\frac{N}{n_i}) \tag{1-6}$$

将公式（1-5）（1-6）代入到公式（1-4）中，得

$$w_{ij} = \frac{freq_{ij}}{\max tf_j} \times \log(\frac{N}{n_i}) \tag{1-7}$$

公式（1-7）是目前受到广泛认可的的权值计算公式，称为“tf-idf（词频－逆文档频率）”[62]加权模式。在这一加权模式的基础上，许多研究人员根据实际情况进行了改进，并做了实验分析，取得了一定的成果。

1.2.1.3 降维的方法

在自动文摘问题中，面对的主要对象是句子。而用向量空间模型来表示句子，其结果一般是非常稀疏的。很难有效的判断出句子与句子之间的关系。因此人们在采用向量空间模型来研究自动文摘问题时，首先要考虑的是降维问题。

降维有多种不同的方法，主要有随机映射（Random Projection）[66][67][68][69]、非负矩阵分解（Non-negative Matrix Factorization）[70][71]、概念索引（Concept Indexing）[72]、潜在语义索引（Latent Semantic Index，LSI）[66][67][73]方法等。

（1）随机映射

设 X 为 m*n 的矩阵，通过一组随机向量，可以将其映射到一个低维（r 维，r<<n）的子空间，从而降低其维度数据维度：

$$S_{m\times r} = X_{m\times n} \bullet R_{n\times r} \tag{1-8}$$

Johnson-Lindenstrauss 引理[68]是 RP 思想的主要依据：

对于任意 0<ε<1 与整数 n，设 r 为正整数，且使得：

$$r \geq 4\left(\frac{\varepsilon^2}{2} - \frac{\varepsilon^3}{3}\right)^{-r} \ln n \tag{1-9}$$

则对于 R^d 中 n 个点的集合 W，存在一个映射 $f:R^d \rightarrow R^r$，使得对所有的 u，v∈W，有：

$$(1-\varepsilon)\|u-v\|^2 \leq \|f(u)-f(v)\|^2 \leq (1+\varepsilon)\|u-v\|^2 \tag{1-10}$$

引理说明高维欧式空间可以映射到一个 $O(\log\frac{n}{\varepsilon^2})$ 维子空间，使得点间距离对于任意 0<ε<1 能近似保留（即偏差不超过（$1\pm\varepsilon$）的因素）。而且这个映射可以在多项式时间内找到。

（2）非负矩阵分解

文本特征矩阵中各个数据元素的值总是非负的，对于这样的 m×n 非负矩阵 $V=(v_{ij})_{m\times n}$，可以近似地分解为两个非负矩阵 $W=(w_{ij})_{m\times r}$ 与 $H=(h_{ij})_{r\times n}$ 的乘积[54,55]，使得：

$$V \approx WH \tag{1-11}$$

其中 r 通常比 m 和 n 都要小得多，也即满足(m+n) r<mn，使得 W 和 H 都要比原矩阵 V 小得多。取式（1-11）的第 i 列表示，即 $v_i=Wh_i$，其中 v_i 和 h_i 是 V 和 H 的

第 i 列，则每一数据 v_i 都是 W 的列的正线性组合，组合的系数为 h 的元素值。这样，$W=[w_1,w_2,\cdots,w_r]$就可看作是对 V 进行线性估计而优化了的基向量。用相对较少的（r 个）基表示许多（m 个）观测数据（r<m）。如果这些基$(w_1, w_2,\cdots, w_r)$能揭示出隐藏在$(v_1, v_2, \ldots, v_m)$中的数据结构，就可获得对观测数据 v_i 好的估计 Wh_i。

非负矩阵分解可以转化为优化问题，用迭代方法交替求解 W 和 H，先固定其中的一个矩阵来计算另一个矩阵，再固定另一个来计算这一个矩阵，交替进行。

（3）概念索引

设 W 为 m×n 文本词条矩阵（其中，m 为文本集中文本的数目，n 为文本集中不同的词条数目），W 的第 i 行为第 i 个文档的向量空间表示（即 $W[i, *]=w_i$），W 的第 j 列为第 j 个词条在各个文档中出现的频率，再设 r 为要降至的维数。概念索引先采用某种简单的聚类算法（k- means 或层次算法等）对文本集作 r 路聚类，将文本集分成 r 个互不相交的子集 $S_1, S_2, \cdots, S_r$，然后，对每个集合 S_i 分别计算质心点向量 C_i，并将它们规格化为单位向量 C_i’，将每一个单位质心向量 C_i’作为降维空间的一个坐标轴，形成一个 r 维子空间，每个文本的 r 维向量表示可通过将其映射到这个 r 维子空间得到，因此，降维后的文本向量空间 W’可通过下式的矩阵运算得到：

$$W_{m\times r}' = W_{m\times n} \bullet C_{n\times r} \tag{1-12}$$

其中，$C_{n\times r}=(C_1', C_2', \cdots, C_r')$ 。

（4）潜在语义索引（LSI）

隐含语义索引利用奇异值分解来实现降维，并凸现矩阵向量间隐含的语义特征。对文本词条矩阵 $W_{m\times n}$，经奇异值分解[57]，矩阵 W 可表示为三个矩阵的乘积：

$$W=UAV^T \tag{1-13}$$

其中，U 和 V 分别为与矩阵 W 对应的左、右奇异向量矩阵，A 为由矩阵 W 的奇异值按递减顺序排列构成的对角矩阵。取 U 和 V 最前面的列构建 W 的 r 秩近似矩阵 W_r:

$$W_r=U_rA_rV_r^T \tag{1-14}$$

U_r 和 V_r 的列向量为正交向量，分别作为文本向量和词向量，用 W_r 来近似表示文本词条矩阵 W，从而实现降维。

1.2.2 概率主题模型

除了向量空间模型以外，另一种重要的文本模型就是概率主题模型。概率主题模型[70-85]是基于这样一种思想：文档是主题的混合物，主题是关于单词的概率分布函数。概率主题模型是一种生成模型，该模型描述了一个生成文档的随机过程。创建新文档时需要确定主题的分布，然后根据这个分布随机选择一个主题，再从主题中生成单词。不同内容的文档可以通过选择主题上的不同分布来生成。

以主题的概率来表示单词和文档的内容，比向量空间模型的表示有显著的优点。每个主题都是一组单词上的概率分布，这实际上是对内容相关的单词的一种聚合。概率主题模型在很多应用中有着极好的效果[78][81][82]。

1.2.2.1 概率生成过程和统计推断过程

一个文档的生成模型，是基于一个简单的概率采样规则，来描述文档中的单词是如何基于潜在变量而生成的。调试一个生成模型，其目标是找出一个潜在变量集，它能最好的解释观察到的数据，假设模型的真正的生成了这些数据。图 1.2 从两个方面解释了生成过程。主题 1 和 2 分别是与金钱和河流相关的两个主题，表示为包含不同分布的单词的集合。依据主题的权重从中选取单词，可以生成不同的文档。文档 1 是仅从主题 1 中取样生成的，文档 3 是仅从主题 2 中取样生成的，而文档 2 则是两个主题的均匀混合而生成的。注意文档中单词的上角标，表示这个单词是从哪个主题采样得到的。从模型定义的方式可以看出，单词与主题之间并没有严格的排他性，并不是一对一的，一个单词可以从不同的主题中产生。这样主题模型可以很好的反映一词多义性。举例来说，在金钱和河流这两个主题中，都有一定的概率产生单词 bank，这就说明 bank 这个单词含有两种不同的含义。

这里描述的生成过程，对单词出现在文档中的顺序没有任何假设。与这个模型唯一相关的信息是单词产生的次数。这就是著名的词袋（bag-of-words）假设，这一假设在统计语言模型中非常普遍。当然单词的顺序信息也许含有文档内容的重要线索，但是这些信息在模型中没有得到利用。Griffiths 等人曾提出了一个主题模型的扩展[83]，利用了单词的顺序信息作为语义因子来引导单词的选择。

一个文档的概率主题模型，可以看做是文档的生成过程，也可以反过来看做是一个统计推断过程。

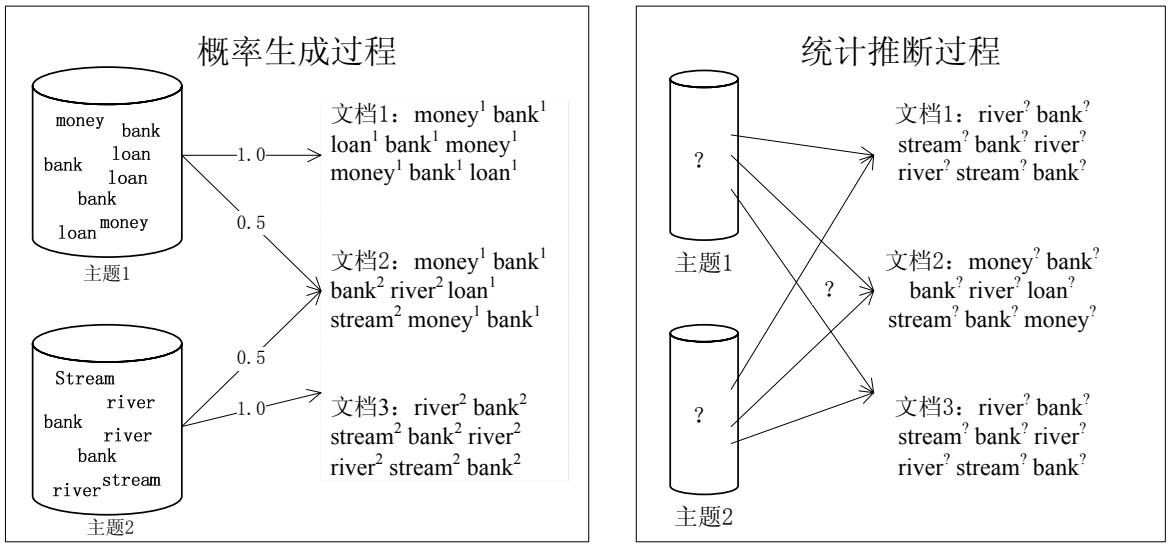

图 1.2 概率主题模型的生成过程和统计推断过程

图 1.2 的右半部分则说明了统计推断问题。给定文档中观察到的单词集合，希望知道生成这些数据的模型是怎么样的。这包括推断单词与每个主题相关联的概率分布、每个文档的概率分布等等。

1.2.2.2 概率潜在语义索引（pLSI）

Hofmann 于 1999 年提出了概率潜在语义索引（Probabilistic Latent Semantic Indexing，pLSI）[79]，是一种最典型的概率主题模型。

pLSI 的核心是一个基于潜在变量的统计模型[84][85]。对一般的共现数据，每一个观察关联一个未观察类变量。也就是说，对于文档 $d\in D=\{d_1,\cdots,d_N\}$ 中的单词 $w\in W=\{w_1,\cdots,w_M\}$ 的每一次出现，都关联一个潜在的主题变量 $z\in Z=\{z_1,\ldots,z_k\}$。作为一个生成模型，pLSI 可以定义如下：

1、通过概率 P(d)选择一个文档 d

2、通过概率 P(z|d)选择一个潜在主题 z

3、通过概率 P(w|z)生成一个单词 w

在结果中只包含文档单词对(d,w)，而潜在主题变量 z 被舍掉了。

把这一过程转换成一个联合概率模型，则结果如下：

$$P(d,w)=P(d)P(w|d) \tag{1-15}$$

其中

$$P(w|d)=\sum P(w|z)P(z|d) \tag{1-16}$$

本质上，要获得上述公式，必须将可能产生观察的 z 的所有选择进行加和。pLSI 模型是一个统计混合模型[86]，基于两个独立性假设：（1）观察到的文档单词对(d,w)是独立生成的，这与词袋方法是相对应的；（2）对潜在主题 z 的条件独立性假设。

与文档聚类模型相比，这里的文档单词分布 P(w|d)是由 P(w|z)的组合而得到。文档没有关联到聚类，而是由权重为 P(z|d)因子的特殊混合来刻画的。这种混合提供了更强的建模能力，在概念上与聚类模型中的后验概率完全不同。

P(d)、P(z|d)、P(w|z)可以通过极大似然估计得到，似然函数如下：

$$\ell = \sum_{d \in D} \sum_{w \in W} n(d,w) \log P(d,w) \qquad (1\text{-}17)$$

其中 n(d,w)代表词频，即单词 w 在文档 d 中出现的次数。值得注意的是，根据贝叶斯公式，通过倒置条件概率 P(z|d)，可以得到一个等价对称的模型，即：

$$P(d,w) = \sum_{z \in Z} P(z)P(w|z)P(d|z) \qquad (1\text{-}18)$$

这只是模型的一个重新参数化的版本。

1.3 多文档自动文摘研究的重点问题

从前面的分析可知，当前多文档文摘的技术研究主要是抽取式的文摘。对于抽取式的多文档文摘，尽管不同的系统采用的方法不同，但按照总体的研究路线来看，都有三个必不可少的重点问题：（1）句子相似度的计算；（2）文摘句的抽取算法；（3）文摘句的排序。这三项问题是多文档文摘的关键技术[124]。任何一个多文档文摘系统都要面临这三个问题。而解决好这三个问题，也成为一个自动文摘系统成败的关键。此外，对于自动文摘系统如何评价，也是与自动文摘相关的研究热点问题。

1.3.1 句子相似度的计算

如何计算两个句子之间的相似程度，这是多文档文摘研究的基础，也可以说是最关键的一步。通过相似度的计算，可以了解两个句子之间在内容上的关系。如果两个句子的关系过于紧密，说明这两个句子中含有冗余的信息，在抽取文摘句时可以挑选冗余性最小的句子。如果某句子与主题句或文档标题相似度很高，说明该句子很大程度上可以代表文档内容，在抽取时可以着重考虑。可以看出，句子相似度的计算，是

自动文摘研究的基础，是决定句子抽取策略的关键性技术。而且句子相似度计算不仅在多文档文摘中，在其他相关的文本智能处理问题中，如问答系统、机器翻译等领域中也发挥着重要作用。很多国内外的专家学者在句子相似度计算方面做了许多研究工作，大致归纳为以下几种方法：

基于单词匹配的句子相似度计算[125]。该方法是仅依靠句子中单词的匹配来确定句子之间的关系的。这种方法简单直观，也便于实现，但是对句子的深层含义没有挖掘，无法真正理解句子的意义。同时对于同义词或一词多义的现象也无法识别，应用上有相当的局限性。

基于向量空间模型的句子相似度计算。把句子表示成向量空间中的向量，通过向量之间的距离来计算句子之间的相似程度。这种方法是目前应用较为普遍的一种方式，但是由于向量空间的维度过大，如何降维是研究的主要问题。通过潜在语义索引的办法来进行降维[126]，是效果不错的一种方法。但是潜在语义索引进行矩阵运算计算代价较大，对实时性会有一定影响。

基于语义辞典的句子相似度计算[19]。单词与单词之间，存在着千丝万缕的联系。借助语义词典，可以找出这些联系，对单词进行深层理解。把两个句子中所有单词的语义相似度汇总后得到就可以得到这两个句子之间的相似度。但是目前这种方法对于一词多义的现象、同一单词在不同上下文中的语义差别现象的处理并不好，往往会使本来无关的单词产生联系。

基于句法分析的句子相似度计算[49][111]。这种方法对句子含义的研究则更加深入，不但研究构成句子的单词，还研究句子的结构。句子的结构通过单词之间的修饰关系来表现。在计算句子的相似度时，句子结构和单词本身的信息都要考虑。从理论上来说，这种方法会较为准确的把握句子之间的相似程度。但是句法分析技术目前还不是很成熟，因此这种方法只能停留在初步探索阶段。

句子相似度的计算在自然语言处理的很多领域中有着广泛的应用，但是不同领域的侧重点也不同。在自动文本摘要技术的研究中，不但要考虑句子与句子之间的相关性，还要考虑一个句子与整篇文档、整个文档集之间的关系。这一点在之前已有的方法中并没有得到应有的重视。

1.3.2 文摘句的抽取

句子抽取技术，是自动文摘技术中的核心。抽取技术解决的是如何从原文档集中抽取句子、抽取哪些句子、抽取句子应该有什么样的标准等问题。

在当前多数的多文档文摘技术研究中，抽取句子时重点考虑的是两个问题：一是该句子是否能够表现原文档集的内容；二是是否有冗余的信息。人们总是希望抽取出的句子既包含丰富的语义信息，又能够含有较少的冗余。

为了判断一个句子是否能够表现原文档集的内容，多数研究工作中的做法是将文档集合中所有的句子按照某一个或多个特征的值进行排序，按照排序的结果从高到低进行文摘句抽取。排序所依据的特征，主要有其所包含单词在文档集中出现的频率、该句子的位置、句子与标题或首句的相似程度、句子的长度等等。通常排序越靠前的句子被认为是越能体现原文档集内容的句子。

为了消除句子之间的冗余信息，典型的做法是考察当前句子与已抽取出的句子之间的相关性，如果相关性较高，则认为新句子的冗余度较高。也有人在抽取过程中动态的调整句子的排序。或者采用聚类的办法，将意义重复的句子聚成一类，从每个类别中选择一个句子作为代表。

许多专家学者对上述两点做了多年的深入的研究，也取得了丰硕的成果。然而，Daniel 于 2011 年完成的博士论文[87]中，对现有的自动文摘的抽取方法提出了质疑。他认为此前人们对于自动文摘的研究，大多集中于过程——用什么样的过程来进行抽取。而今后的研究工作则应该从过程转移到目标——究竟什么样的文摘是好的，是否存在好文摘的数学模型。

1.3.3 文摘句的排序

抽取出文摘句以后，需要将这些文摘句按照一定的顺序排列起来，才能形成最终的文摘。一个好的句子排列顺序，可以使生成文摘阅读起来更加流畅，也能够帮助人们更好的理解原文的意思。因此，文摘句的排序方法，也是自动文摘技术研究中的一个很重要的问题。

句子排序问题在单文档文摘中并不突出，因为单文档文摘中可以将抽取出的句子按照原文的顺序排列。但是在多文档文摘中，句子都来源于不同的文档，其排列的方

法就有待研究了。

文摘句的排序方法，需要考虑句子的内容和时间信息，哥伦比亚大学 Regina Barzilay[103]已经在这方面做了一些工作，他们提出的 Majority Ordering（MO）[104]算法依据原文的顺序来确定文摘句的顺序关系。Zhuli Xie[114]通过进化算法（Gene Expression Programming，GEP）作为学习机制，从手工文摘和原文档的关系中学习到排序的规律，对句子进行排序，这种方法的不足之处在于对手工文摘的依赖，从一个文档集中学习到的规律很难应用于新的文档集，因此需要找到更客观的特征来刻画句子顺序的规律。Bollegala 等人于 2010 年提出了 AGL 算法[110]，这种方法中运用了多种标准来衡量句子间的关系，取得了较好的效果。

1.3.4 自动文摘的评价

评价自动生成的文摘的好坏，是一个非常困难的任务。因为对于任何给定的文档或文档集，都不存在一个理想的完美文摘可以做为评价参考。目前的做法，通常都是以人手工编制的文摘作为参考标准。但是就目前的研究情况来看，不同的人手工编制的文摘之间差距非常大，不同的人对于同一篇文摘的评价之间差距也非常大。另外，度量的标准不统一也是一个自动文摘评价中的重要问题。这使得比较不同系统和建立基线都变得非常困难。这一问题在自然语言理解中的其他领域并不存在。另外，手工文摘过于昂贵也是很重要的问题。正如 Lin[37]所指出的，像 DUC 会议这种大规模的手工文摘评价，需要超过 3000 小时的人工。

目前存在的自动文摘评价方法，主要有内部评价和外部评价两种。内部评价是指直接对文摘的质量进行评估，根据摘要的组织是否流畅、是否能够很好的理解、是否有效的反映了原文的内容、长度是否符合要求等几个方面来考察文摘。具体的方法主要有直接阅读文摘并给与评价、将自动生成的文摘与人们手工编写的文摘进行对比、将自动生成的文摘与原文进行对比、将不同系统生成的文摘相互对比等。外部评价则是指，将文摘应用于某种具体的任务中，根据文摘与原文在执行该任务时的差别程度来评价文摘的好坏。

外部评价方法的基本思想是借助于自动文摘系统完成一些任务，通过文摘系统在任务完成过程中所起作用的大小来对其进行评价[42][43]。比如为某些操作说明书生成摘要，看摘要能否有效地指导操作。还可以通过摘要来判断原文内容是不是人们所关心

的，如果摘要能够帮人们做出正确的判断，那么这个摘要也是有效的。或者将自动文摘系统应用到自动问答系统中，看会起到什么作用。也有人对自动文摘系统输出的文摘进行编辑，使之成为可以接受的文摘，看这个编辑的过程是否很费力气。

内部评价方法主要考察两种指标，一是文摘可读性，也就是文摘在表达方面的质量（Quality），二是文摘的信息概括性（Informativeness），也就是文摘是否有效的反映了原文的内容。

在文摘的表达质量方面，有很多不同的具体评价准则。譬如，J-L.Minel 等[39]根据是否有不明的指代、是否有同义反复、是否完整的保留原文中图表信息来给文摘的可读性进行人工评分。还有 H.Saggion 和 G.Lapalme[40]采用 J.Rowley[41]的建议，根据拼写、语法、是否清楚的标引原文的话题、是否受个人风格影响、简明、可读以及可理解标准对文摘进行打分。上面的例子都是人工参与评价的，都属于在线评价。也有人尝试了对文摘的表达质量以离线的方式进行评价。如 Inderjeet Mani[43][5]基于单词的长度、句子的长度提出了一个自动评价文摘表达质量的方案，但这种方案比较粗糙，不能很好地反映文摘的可读性和可接受程度。总而言之，到目前为止，有效地对文摘的表达质量进行评价还只局限于在线方式。

对于文摘的信息概括性，学者们有两种理解：一种理解是文摘保留了多少源文本所含的信息，还有一种是文摘中包含多少“理想文摘”(Idea Summary)所含的信息。相应地，对文摘信息概括性的评价也分成两种：一种将生成的文摘和源文本对比，另一种是生成的文摘与“理想文摘”比较。在自动文摘研究的早期，研究人员多采用前一种方法对文摘进行评价，而目前的一些方法多属于第二种。由于文摘本身的主观性，这里的“理想文摘”是假想的，一般情况下都以人工编制的摘要代替，人工编制的文摘又被称为“参考文摘”(Reference Summary)。到目前为止已经提出了多种基于参考文摘进行的内部评价办法[44][45]，同时也遇到了一些困难。

在 DUC、TAC 等国际会议中，对自动文摘的评价大多采用内部评价的方式。前三届 DUC 会议采用的是由南加州大学信息科学研究所的 Chin-Yew Lin[46]开发的人工评价平台（Summary Evaluation Environment，SEE）。从 2004 开始又采用了 ROUGH 评价工具包[48]，这个工具包是 Chin-Yew Lin 受机器翻译评价方法 BLEU[47]的启发而开发的，主要是通过生成的文摘与参考文摘的对比来进行评价的，包括 n 元组的重复程度、最大公共子序列等。2005 年开发的 Pyramids 方法[122]，是在文摘内容单元（Summarization Content Unit，SCU）的基础上对文摘进行评价，取得了不错的成绩。

1.4 论文的研究内容

前面介绍了自动文摘技术中的四个重点问题：句子相似度的计算、文摘句的抽取、文摘句的排序、自动文摘的评价。而本文正式针对这四个问题，对自动文本摘要技术进行了深入的研究探讨。

在相似度计算方面，针对传统的向量空间模型中句子表示过于稀疏的问题，引入潜在狄利克雷分配的生成模型，用潜在主题代替原来的单词构成潜在主题空间。将单词、句子、文档、文档集等不同粒度的文本单元映射到相同的潜在主题空间中，在相同的平台上进行统一的度量。这样的度量方式不但克服了之前技术的缺点，而且为后面句子的抽取和排列打下了良好的基础。

在文摘句的抽取问题上，本文将研究重点发在了生成文摘的目标上。紧扣文摘要充分反映原文档集的内容这一目标，综合了聚类法与排序法的优点，提出了逐步从原文档集中去除无意义句子的算法。这种方法与传统的句子排序及句子聚类的方法，从出发点上存在着本质的不同。不再关注每个句子是否反映了原文的内容、句子间的信息是否冗余这些传统问题，而是直接关注所生成文摘的整体质量。因此用该方法生成的文摘，在整体质量的评估中能够获得较高的分数。

在文摘句的排序方面，本文研究了句子与句子之间相互关系的特点，从潜层表层、本位语境、前置后置等多个方面对句子之间的关系进行了深入的剖析。并在此基础上提出了基于相关距离的、基于原文顺序概率的、基于概率主题模型的多种文摘句排序算法。

在自动文摘的评价方面，本文利用潜在主题空间中的 LDASim 度量，提出了 LS 自动评价方法。在文档相关评测、句子相关评测、句子对应评测的基础上，定义了可读性、无冗余性、全面性评分。综合多种评分，得到文摘的最终评价结果。

1.5 论文的组织结构

本文组织结构如下：

第一章绪论。主要对选题的研究背景、意义和当前的发展状况进行了阐述， 着重介绍了自动文本摘要的相关概念、自动文本摘要技术的发展历史及现状、主要的文本模型、自动文本摘要技术的研究重点和难点、自动文摘评价的主要方法和思想等，并给出了本文的主要研究内容和论文的组织结构。

第二章基于概率主题模型的文本相似性度量。介绍了相似性度量研究的现状。接着通过了潜在狄利克雷分配对文本进行建模，利用潜在主题构建潜在主题空间，并将不同粒度的文本单元表示成潜在主题向量的形式。提出了 LDASim 相似性度量方法，并通过实验证明该方法的有效性。

第三章基于差分进化和概率主题模型的句子抽取算法。介绍了句子抽取技术的研究现状。针对现有技术中的不足，提出了基于差分进化的句子聚类方法和基于概率主题模型的逐句递减规则。将二者相结合，提出了差分进化和概率主题模型相结合的句子抽取算法。通过具体的句子抽取实验，将本文提出的算法与现有的算法进行比较，以证明本文提出算法的有效性。

第四章基于概率主题模型的文摘句排序算法。首先介绍了句子排序问题的重要性和复杂性，以及当前主要的句子排序方法。从潜层表层、本位语境多个层面来定义句子间的相似度，提出了基于相关距离的文摘句排序算法；从原文档集中学习句子的前置概率和后置概率，提出了基于原文顺序概率的文摘句排序算法；综合相关距离、前置后置概率、时间顺序等多种标准，提出了基于概率主题模型的层次性文摘句排序算法。最后通过实验验证本文提出的三种算法的有效性。

第五章自动文本摘要的评价方法研究。首先介绍了自动文摘评价方法的总体思想和分类，接着详细介绍了目前国际会议中使用的主流的评价方法。提出了基于概率主题模型的 LS 文摘自动评价方法。在文档相关评测、句子相关评测、句子对应评测的基础上，定义了 LS 可读性、LS 无冗余性、LS 全面性评分。综合多种评分，得到文摘的最终评价结果。在实验中，将 LS 与现有的自动评测方法以及人工评测方法相对比，表明 LS 评价方法的结果更为接近人工评价的结果。

结论。对全文的工作进行了总结，说明了本文的主要创新之处，并对下一步的研究工作做了展望。

第2章 基于概率主题模型的文本相似性度量

2.1 引言

相似性度量技术，在智能信息处理中有着广泛的应用，是多文档自动文摘技术的基础。

所谓相似性度量，就是度量两个事物之间的相似程度。具体到文本处理中，相似性度量就是指度量两个文本单元的相似程度。这里的文本单元，可以是单词、短语、句子、段落、文档、文档集等等。而这其中，句子与句子之间的相似性度量是自动文摘技术中最常用到的。

要衡量两个句子之间的相似性，需要解决两个问题：一是句子的建模，即用什么样的数学模型来表示句子；二是度量的具体方法。度量的具体方法，往往跟句子的数学模型密切相关。不同的模型就有不同的度量方法。因此，这两个问题实质上可以归结为一个问题，就是建模问题。

句子本身是由单词构成的，因此最直观最基础的句子表示方法，就是将句子看成是单词的集合，即 $S=\{word_1, word_2, \ldots, word_n\}$。在这种建模方式下，衡量两个句子之间的关系，就是看这两个集合的交集有多大，也就是两个句子包含多少相同的单词。这种方法虽然直观简单，但是过于粗糙，无法表现出两个句子之间深层的语义关系。

人们经过多年的研究和探索，已经找到了很多方法，可以较好的表示文本单元，其中最主要的有向量空间模型。

向量空间模型（Vector Space Model，VSM）[50]是由 Gerard Salton 提出的，是目前为止最为常见的一种用于文本表示的数学模型。该模型在很多智能信息处理领域，如文本自动分类、信息检索、自动摘要等，都有着十分广泛的应用，是目前的文本表示模型中最简便有效的方法之一[51,52]。向量空间模型，就是采用多维向量来表示文本单元，通过计算向量间的相似度实现文本单元之间相似度的计算。

当文本单元被表示成空间中的向量以后，就可以采用向量之间的某种距离来表示文本单元之间的相似度。文本单元间的相似度计算常用向量之间的内积来表示：

$$Sim(D_i, D_j) = \sum_{k=1}^{n} w_{ik} \times w_{jk} \qquad (2\text{-}1)$$

为了向量的归一化，人们更常使用的则是利用向量之间夹角的余弦值来计算文档间的相似度：

$$Sim(D_i, D_j) = \cos\theta = \frac{\sum_{k=1}^{n} w_{ik} \times w_{jk}}{\sqrt{\left(\sum_{k=1}^{n} w_{ik}^2\right) \times \left(\sum_{k=1}^{n} w_{jk}^2\right)}} \quad (2\text{-}2)$$

在自动文摘问题中，面对的主要对象是句子。而用向量空间模型来表示句子，其结果一般是非常稀疏的。很难有效的判断出句子与句子之间的关系。因此人们在采用向量空间模型来研究自动文摘问题时，首先要考虑的是降维问题。降维有多种不同的方法，主要有随机映射（Random Projection）[66,67,68,69]、非负矩阵分解（Non-negative Matrix Factorization）[70,71]、概念索引（Concept Indexing）[72]、隐含语义索引（Latent Semantic Index，LSI）[66,67,73]方法等。

除了向量空间模型以外，另一种重要的文本模型就是概率主题模型。概率主题模型[70-85]是基于这样一种思想：文档是主题的混合物，主题是关于单词的概率分布函数。主题模型是一种生成模型，该模型描述了一个生成文档的随机过程。创建新文档时需要确定主题的分布，然后根据这个分布随机选择一个主题，再从主题中生成单词。不同内容的文档可以通过选择主题上的不同分布来生成。以主题的概率来表示单词和文档的内容，比向量空间模型的表示有显著的优点。每个主题都是一组单词上的概率分布，这实际上是对内容相关的单词的一种聚合。概率主题模型在很多应用中有着极好的效果[78,81,82]。

本章将采用概率主题模型对文本进行建模，并结合向量空间模型的良好特性，提出一种基于概率主题模型下的潜在主题空间的文本表示和度量方法。

2.2 基于概率主题模型的相似性度量 LDASim

在向量空间模型下，文本单元的表示简单而明确，可以很容易的通过向量间夹角的余弦值来计算两个文本单元的距离。但是，由于句子与文档相比，其包含的特征词的数量差距过大，导致表示句子的向量极其稀疏。尽管可以有很多方法进行降维，但是这些降维方法都是以特征词总数为基础的。特征词的总数过于庞大，且其数量取决于文档集而不是人为可控。用向量空间模型来表示句子，效果很差。在自动文本摘要相关的研究中，主要的处理对象就是句子，因此要想办法既保留向量空间模型的良好

特性，又避免以数量庞大的特征词作为维度。

在概率主题模型中，文档集被看做是潜在主题的混合物，是以潜在主题的概率来表示文本单元的，而潜在主题的数量是人为可控的。这就提供了一个思路：用潜在主题为维度的空间代替特征词为维度的向量空间。

本节中，首先采用潜在狄利克雷分配对文档集进行概率主题建模，接着将文本单元表示成潜在主题空间中的向量形式，采用余弦聚类对向量间的相似程度进行度量。

2.2.1 潜在狄利克雷分配

概率潜在语义索引（probabilistic Latent Semantic Indexing，pLSI）[79]是一种典型的概率主题模型。但是在 pLSI 模型中，对于混合权重 θ=P(z|d)如何产生没有任何的假设。这使得验证模型对新文档集的一般性非常困难。Blei 扩展了这一模型，在 θ 上引入了一个狄利克雷先验，这个模型被称为潜在狄利克雷分配（Latent Dirichlet Allocation, LDA）[120]。

LDA 模型假设对于文档集 C 中的每一个文档 D，都有如下的生成过程：

（1）选择 N，服从泊松分布 Poisson(ξ)

（2）选择 θ，服从狄利克雷分布 Dir(α)

（3）对每一个单词 w_n，有：

（a）选择一个主题 z_n，服从多项式分布 Multinomial(θ)

（b）选择一个单词 w_n，依据概率 $p(w_n|z_n,\beta)$，以主题 z_n 为条件的多项式概率。

在这个基本的模型中，有一些简单的假设。首先，狄利克雷分布的维度 k 假设是已知的且固定不变的；第二，单词概率由 k×V 的矩阵 β 进行参数化，其中 $\beta_{ij}=p(w_j=1|z_i=1)$，这是需要估计的一个固定值；最后，泊松分布与其后的模型没有必然的联系，通常直接使用文档长度即可。也就是说，其中的 N 与其他的数据生成变量（θ 和 z）无关，一般在实践中忽略其随机性。

一个 k 维的狄利克雷随机变量 θ 可以从 k-1 维单纯形上取值，其概率密度为：

$$p(\theta|\alpha)=\frac{\Gamma(\sum_{i=1}^{k}\alpha_i)}{\prod_{i=1}^{k}\Gamma(\alpha_i)}\theta_1^{\alpha_1-1}\cdots\theta_k^{\alpha_k-1} \tag{2-3}$$

其中参数 α 是一个 k 维向量，每一维的值 $\alpha_i>0$。Γ(x)是 Gamma 函数。

给定参数 α 和 β，则主题 z、文档 D 和混合系数 θ 的联合分布为：

$$p(\theta, z, D \mid \alpha, \beta) = p(\theta \mid \alpha) \prod_{n=1}^{N} p(z_n \mid \theta) p(w_n \mid z_n, \beta) \tag{2-4}$$

这里 $p(z_n|\theta)$是每一维 θ_i 使得 $z_n^i = 1$的概率。对 z 求和并对 θ 求积分，就得到了一个文档 D 的边缘分布：

$$p(D \mid \alpha, \beta) = \int p(\theta \mid \alpha) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n \mid \theta) p(w_n \mid z_n, \beta) \right) d\theta \tag{2-5}$$

最后，将单个文档的边缘概率累乘，就得到了一个文档集的概率：

$$p(C \mid \alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d \mid \alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn} \mid \theta_d) p(w_{dn} \mid z_{dn}, \beta) \right) d\theta_d \tag{2-6}$$

图 2.3 给出了 LDA 模型的图形表示。从图中可以看出，LDA 模型分为三个层次：α 和 β 是文档集层的参数，在生成一个文档集时只采样一次；θ_d 是文档层的变量，每个文档采样一次；z_{dn} 和 w_{dn} 是单词层的变量，对每个单词都要采样一次。

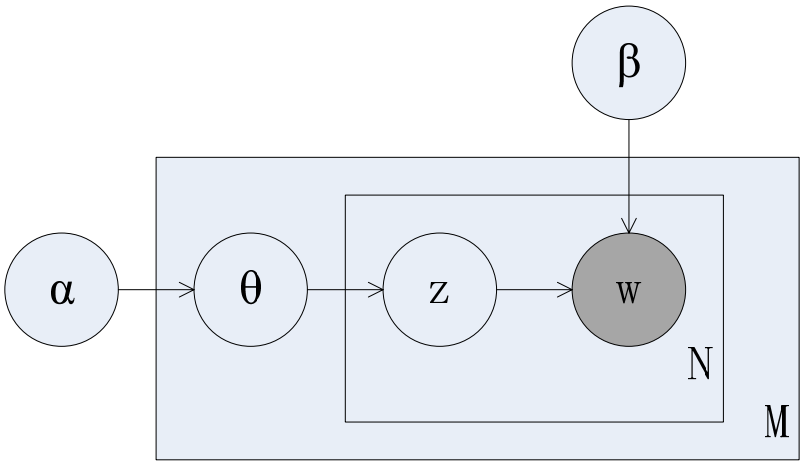

图 2.1 潜在狄利克雷模型的几何表示

从前面的模型介绍可以看出，对于一个文档集而言，LDA 模型只有两个参数 α 和 β。这两个参数可以通过 EM 算法[36]估计得到，不断重复 E 步和 M 步迭代，直到下界函数收敛，即可得到参数 α 和 β 的值。

极大似然函数为：

$$\ell(\alpha, \beta) = \sum_{d=1}^{N} \log p(D_d \mid \alpha, \beta) \tag{2-7}$$

由于 $p(D_d \mid \alpha,\beta)$ 不好计算，又有 $\log p(D_d \mid \alpha,\beta) \geq L(\gamma,\phi \mid D,\alpha,\beta)$，所以可以用 $L(\gamma,\phi \mid D,\alpha,\beta)$ 做为下界函数，这里

$$L(\gamma,\phi \mid D,\alpha,\beta) = E_{q(\theta,z)}(\log P(\theta,z,D \mid \alpha,\beta)) - E_{q(\theta,z)}(\log q(\theta,z)) \qquad (2\text{-}8)$$

EM 算法如下：

（1）E 步。对每篇文档 D，计算出变分参数的最优值 $\{\gamma_D^*,\phi_D^* : D \in C\}$。

（2）M 步。最大化对数似然函数的下界 $L(\gamma,\phi \mid D,\alpha,\beta)$，得到 α 和 β。

通过上面两步的反复迭代，直到下界函数收敛。这样，就得到了 LDA 模型的参数 α 和 β。

2.2.2 潜在主题空间和潜在主题向量

对于文档集 C，经过上一节介绍的 LDA 建模后，通过变分推理和 EM 算法估计出参数 α 和 β 的值。通过一次狄利克雷采样，可以得到文档集 C 的 k 个潜在主题 $z_1, z_2, \ldots, z_k$。

定义 2-1 潜在主题空间

给定文档集 C 和正整数 k，对文档集 C 进行维度为 k 的 LDA 建模，得到潜在主题 $z_1, z_2, \ldots, z_k$。以 $z_1, z_2, \ldots, z_k$ 为 k 维坐标系的实数空间 R^k 称之为文档集 C 的 k 维潜在主题空间，记作 $\mathbb{Z}_C^k$。

与传统的向量空间相比，潜在主题空间的一大优势是对维数的有效控制。潜在主题空间的维数 k，是人为确定的。而传统向量空间的维数，无法人为确定，是由文档集包含特征词的情况而决定的。尽管人们使用很多的降维技术，传统向量空间的维数依旧与文档集相关。维数人为确定的好处是，人们可以根据不同的需要（比如面临的任务、计算的复杂度、要求的精度等等）来确定不同的维数，而这在传统向量空间中是无法实现的。

潜在主题空间是由潜在主题作为维度的，而传统向量空间是以单词作为维度的。单词是自然语言的基本单位，但是就其含义来说又是复杂的。在使用单词作为维度时，人们往往需要语义消岐等技术，但是依旧无法完全消除一词多义、多词同义等现象的

干扰，而这些现象往往对文本的处理造成很大的困难。在潜在主题空间中，潜在主题就成为表意的基本单位。根据概率主题模型的描述，同一个潜在主题，可以依不同的概率生成不同的单词，这就表达了单词的多词同义性；多个不同的潜在主题，可以按不同的概率生成同一个单词，这说明这个单词包含了多个潜在主题的信息，也就表达出了单词的一词多义性。每个潜在主题的概率不同、每个潜在主题生成单词的概率也不同，这就表达了单词含义随语境的变化而变化的特点。

从上面的分析可以看出，潜在主题空间的良好特性是通过潜在主题与单词之间的概率关系表达出来的，因此使用潜在主题关于单词的条件概率来构成单词在潜在主题空间中的向量。

定义 2-2 单词的潜在主题向量

文档集 C 的 k 维潜在主题空间为 $\mathbb{Z}_C^k$，对于文档集 C 中的一个单词 w_j，以潜在主题 z_i 关于单词 w_j 的条件概率 $p(z_i|w_j)$作为相应坐标值，由此形成的向量称之为单词 w_j 在 $\mathbb{Z}_C^k$ 中的潜在主题向量，记作 $V_z(w_j)$。即

$$V_z(w_j) = (p(z_1 \mid w_j), p(z_2 \mid w_j), \cdots, p(z_k \mid w_j)) \tag{2-9}$$

以上定义的是一个单词的潜在主题向量。对于多个单词构成的单词集，可以用单词的向量求均值的形式来定义单词集的潜在主题向量。

定义 2-3 单词集的潜在主题向量

文档集 C 的 k 维潜在主题空间为 $\mathbb{Z}_C^k$，对于一个单词集 $W = \{w_1, w_2, ..., w_n\}$，$w_i \in C$，把 W 包含的所有单词的潜在主题向量的均值，称为单词集 W 的潜在主题向量，记作 $V_z(W)$。即，

$$\begin{aligned} V_z(W) &= \frac{1}{n}\sum_{i=1}^{n} V_z(w_i) \\ &= (\frac{1}{n}\sum_{i=1}^{n} p(z_1 \mid w_i), \frac{1}{n}\sum_{i=1}^{n} p(z_2 \mid w_i), ..., \frac{1}{n}\sum_{i=1}^{n} p(z_k \mid w_i)) \end{aligned} \tag{2-10}$$

单词是自然语言中最小的语义单位。其他粒度的文本单元，如短语、句子、段落、文档、文档集等，均是由单词组成，都可以看做是不同规模的单词的集合。因此，由定义 2-3 可以得到不同粒度的文本单元的潜在主题向量。

将句子看做一个包含 n 个单词的集合，即 S={w_1,w_2,…,w_n}。那么句子 S 的潜在主题向量可以表示为：

$$\begin{aligned} V_z(S) &= \frac{1}{n}\sum_{i=1}^{n} V_z(w_i) \\ &= (\frac{1}{n}\sum_{i=1}^{n} p(z_1 \mid w_i), \frac{1}{n}\sum_{i=1}^{n} p(z_2 \mid w_i), ..., \frac{1}{n}\sum_{i=1}^{n} p(z_k \mid w_i)) \end{aligned} \tag{2-11}$$

再把文档 D 看做是一个包含 m 个句子的集合，即 D={S_1,S_2,…,S_m}。其中的每个句子，都是单词的集合，都可以根据式（2-11）表示成潜在主题向量的形式。那么文档 D 的潜在主题向量可以表示为：

$$\begin{aligned} & V_z(D) \\ &= V_z(\{S_1, S_2, \cdots, S_m\}) \\ &= V_z(\{\{w_{11}, \cdots, w_{1n_1}\}, \cdots \{w_{m1}, \cdots, w_{mn_m}\}\}) \\ &= (\frac{1}{m}\sum_{j=1}^{m}(\frac{1}{n_j}\sum_{i=1}^{n_j} p(z_1 \mid w_{ji})), \cdots, \frac{1}{m}\sum_{j=1}^{m}(\frac{1}{n_j}\sum_{i=1}^{n_j} p(z_k \mid w_{ji}))) \end{aligned} \tag{2-12}$$

再把文档集 C 看做是一个包含 r 个文档的集合，即 C={D_1,D_2,…,D_r}。其中的每个文档，都是单词的集合，都可以根据式（2-12）表示成潜在主题向量的形式。那么文档集 C 的潜在主题向量可以表示为：

$$\begin{aligned} & V_z(C) \\ &= V_z(\{D_1, D_2, \cdots, D_r\}) \\ &= V_z(\{S_{11}, \cdots, S_{1m_1}\}, \cdots, \{S_{r1}, \cdots, S_{rm_r}\}) \\ &= (\frac{1}{r}\sum_{s=1}^{r}\frac{1}{m_s}\sum_{j=1}^{m_s}\frac{1}{n_j}\sum_{i=1}^{n_j} p(z_1 \mid w_{sji}), \cdots, \frac{1}{r}\sum_{s=1}^{r}\frac{1}{m_s}\sum_{j=1}^{m_s}\frac{1}{n_j}\sum_{i=1}^{n_j} p(z_k \mid w_{sji})) \end{aligned} \tag{2-13}$$

2.2.3 潜在主题向量的计算

在上一节中，已经将各种不同粒度的文本单元都表示成了潜在主题向量的形式。所有的潜在主题向量，都是以单词的潜在主题向量为基础的。而单词的潜在主题向量，其核心是潜在主题关于单词的条件概率 p(z|w)。本节来讨论 p(z|w)的计算方法。

对文档集 C 进行 LDA 建模后，通过变分推理和 EM 算法，可以得到参数 α 和 β

的值。

参数 α 是一个 k 维的正向量，即 $\alpha=(\alpha_1,\alpha_2,\cdots,\alpha_k)$，且 $\alpha_i>0$。以 α 为参数的狄利克雷分布决定了 θ 的值。通过 LDA 模型的第二步，进行一次狄利克雷采样，即可得到 θ 的值，即：

$$p(\theta\mid\alpha)=\frac{\Gamma(\sum_{i=1}^{k}\alpha_i)}{\prod_{i=1}^{k}\Gamma(\alpha_i)}\theta_1^{\alpha_1-1}\cdots\theta_k^{\alpha_k-1} \tag{2-14}$$

潜在主题是服从以 θ 为参数的多项式分布的。将上式采样得到的 θ 代入多项式分布中，即可得到潜在主题的概率，即：

$$p(z\mid\theta)=\frac{(\sum_{i=1}^{k}z_i)!}{z_1!\cdots z_k!}\theta_1^{z_1}\cdots\theta_k^{z_k} \tag{2-15}$$

参数 β 是一个 k×V 的矩阵，它决定了潜在主题 z 与单词 w 之间的概率关系。由 EM 算法已经估计出 β 的值，由此可以得到：

$$p(w_j\mid z_i)=\beta_{ij} \tag{2-16}$$

对于每一个单词 w_j，统计其在文档集 C 中出现的单词总数，记为 $count(w_j)$。设文档集 C 的总词数为 N，则可求得单词 w_j 的概率为：

$$p(w_j)=\frac{count(w_j)}{N} \tag{2-17}$$

根据贝叶斯公式，有

$$p(z_i\mid w_j)=\frac{p(z_i)p(w_j\mid z_i)}{p(w_j)} \tag{2-18}$$

将公式（2-15）（2-16）（2-17）代入到公式（2-18）中，即可计算得出潜在主 z_i 关于单词 w_j 的条件概率。

2.2.4 基于概率主题模型的相似度

通过对文档集进行 LDA 建模，将单词集表示成了潜在主题向量的形式。在潜在主题空间中，可以使用余弦距离来计算两个向量之间的相似程度。

定义 2-4 基于概率主题模型的相似度 LDASim

文档集 C 的 k 维潜在主题空间为 $\mathbb{Z}_C^k$，对于 C 中的任意两个单词子集 $X=\{x_1,x_2,...,x_n\}$，$x_i \in C$ 和 $Y=\{y_1,y_2,...,y_m\}$，$y_i \in C$，表示成 $\mathbb{Z}_C^k$ 中的潜在主题向量 $V_z(X)$和 $V_z(Y)$，把 $V_z(X)$和 $V_z(Y)$在 $\mathbb{Z}_C^k$ 下的余弦相似度称为 X 和 Y 基于概率主题模型的相似度，记作 LDASim(X,Y)，即，

$$LDASim(X,Y)=\cos(V_z(X),V_z(Y)) \tag{2-19}$$

由定义 2-3 可知，$V_z(X)$和 $V_z(Y)$均为正向量，因此 LDASim(X,Y)∈[0,1]。当 X=Y 时，LDASim(X,Y)=1，即一个单词集与其自身的相似度最大。当 X 和 Y 正交时，也就是 X 和 Y 之间没有任何共同关联的潜在主题，则 LDASim(X,Y)=0。

定义 2-4 所定义的 LDASim，其度量对象是任意两个单词集。而在自然语言中，单词是最基本的语言单位，各种粒度的文本单元都可以表示成不同规模的单词集。这就说明 LDASim 的一个最重要的特点，其度量的对象可以是不同粒度的文本单元。

最基本的，LDASim 可以度量两个单词之间的相似度。对于文档集 C 中的任意两个单词 w_i 和 w_j，有

$$\begin{aligned} &LDASim(w_i,w_j) \\ &=\cos(V_z(w_i),V_z(w_j)) \\ &=\frac{P(z_1\mid w_i)P(z_1\mid w_j)+\cdots+P(z_k\mid w_i)P(z_k\mid w_j)}{\sqrt{P(z_1\mid w_i)^2+\cdots P(z_k\mid w_i)^2}\cdot\sqrt{P(z_1\mid w_j)^2+\cdots+P(z_k\mid w_j)^2}} \end{aligned} \tag{2-20}$$

句子的相似度在自动文本摘要技术中非常重要，可应用于句子的聚类、冗余判断等问题中。LDASim 可以用来度量两个句子之间的相似度。对于文档集 C 中的任意两个句子 S_1 和 S_2，有

$$LDASim(S_1,S_2)=\cos(V_z(S_1),V_z(S_2)) \tag{2-21}$$

文档与文档之间的相似度，也是比较常见的问题，在自动文本分类等领域中有着广泛的应用。LDASim 可以用来度量两个文档之间的相似度。对于文档集 C 中的任意两个文档 D_1 和 D_2，有

$$LDASim(D_1,D_2)=\cos(V_z(D_1),V_z(D_2)) \tag{2-22}$$

如果要从一篇文档中找出一个最能代表该文档含义的句子，则需要计算句子与文

档之间的相似度。LDASim 可以度量两个不同粒度的文本单元。对于文档集 C 中的一个句子 S 和一篇文档 D，有

$$LDASim(S,D)=\cos(V_z(S),V_z(D)) \tag{2-23}$$

甚至 LDASim 可以将最小粒度的单词与最大粒度的文档集直接进行运算，找出最能代表该文档集的特征单词。对于文档集 C 中的任意单词 w，有

$$LDASim(C,w)=\cos(V_z(C),V_z(w)) \tag{2-24}$$

可见，LDASim 是一种通用的度量手段，将单词、句子、文档等不同粒度的文本单元表示在同一个潜在主题空间中，可以用统一的方式衡量其中任意两者的相似程度。这也是 LDASim 的意义所在。

2.3 实验及结果分析

上一节定义了潜在主题空间中的通用相似性度量 LDASim，本节将通过具体的实验来验证 LDASim 的效果。

2.3.1 实验数据与实验目的

使用的实验数据，是文本分析会议（Text Analysis Conference，TAC）及其前身文档理解会议（Document Understanding Conference，DUC）中所使用的数据集。DUC2006 和 DUC2007 会议数据集中，共有 50 个类别的数据，每个类别包含 25 篇文档，要求从这 25 篇文档中生成一篇文本摘要，该摘要不能多于 250 个单词。TAC2008 和 TAC2009 会议数据集中，则要求从 10 篇文档中生成不多于 100 个单词的文本摘要。

实验分为两个部分：第一部分是测定潜在主题空间 $\mathbb{Z}_C^k$ 的维度，也就是潜在主题的数量，称之为 k 值实验；第二部分是将 LDASim 与传统的相似性度量方法相比较，称之为度量比较实验。

2.3.2 k 值实验的方法、结果及分析

k 值，也就是潜在主题空间的维度，其大小直接影响到 LDASim 的效果。由于相似性的度量目前尚没有自动的评价方法，采用人工评价的方法。

表 2.1 五名专家对 C1 进行 k=10 的 LDASim 的评价结果

	专家 1	专家 2	专家 3	专家 4	专家 5
句子 1	可接受	可接受	可接受	可接受	可接受
句子 2	不可接受	可接受	不可接受	不可接受	可接受
句子 3	非常好	非常好	非常好	非常好	可接受
句子 4	不可接受	不可接受	不可接受	不可接受	不可接受
句子 5	不可接受	不可接受	可接受	不可接受	不可接受
句子 6	不可接受	不可接受	不可接受	不可接受	不可接受
句子 7	非常好	可接受	非常好	可接受	可接受
句子 8	可接受	可接受	可接受	不可接受	不可接受
句子 9	可接受	不可接受	不可接受	不可接受	不可接受
句子 10	可接受	不可接受	不可接受	不可接受	不可接受

共有五名专家参与评价工作。从 DUC2006 数据中选出 20 个类别，分别记为 C1，C2……C20。令 k=10，对 C1 中的 25 篇文档，进行 LDA 建模训练。之后用 LDASim 评价 C1 中每篇文档的每个句子与文档集 C1 之间的相似程度，选出前十名的句子。由五名专家分别对这十个句子进行评估，评估分为 3 个等级：非常好、可接受、不可接受。“非常好”表示该句子确实能够很好的代表原文档集的内容，不论是生成文摘，或是抽取主题句，选择该句子都有很强的说服力；“可接受”表示该句子与原文档集的内容有一定的关联度，勉强可以用来代表原文档集；“不可接受”表示该句子与原文档集的核心内容关联不大，不能用它作为原文档集内容的代表。当 k=10 时，对文档集 C1 中的句子进行 LDASim 度量，取前十名的句子由五名专家进行评分，其结果如表 2.1 所示。

将评价结果转换成对应的分值，不可接受=0 分，可接受=1 分，非常好=3 分，则可以将上述结果进行加和平均，得到的结果如表 2.2 所示。

最后总的平均得分为 0.7 分，低于可接受的 1 分，可见对文档集 C1 来说，k=10 的 LDASim 效果是很差的。将同样的实验对全部 20 个文档集进行，计算其平均分，得到的结果如表 2.3 所示。

最终评分是 0.737 分，小于可接受的 1 分。可见，k=1 时，LDASim 的总体评价能力较差。逐步扩大 k 值，分别取 k=100、300、500、800、1000 重复上述实验，得

到结果如表 2.4 所示。

表 2.2 五名专家对 C1 进行 k=10 的 LDASim 的评分

	专家 1	专家 2	专家 3	专家 4	专家 5	平均
句子 1	1	1	1	1	1	1
句子 2	0	1	0	0	1	0.4
句子 3	3	3	3	3	1	2.6
句子 4	0	0	0	0	0	0
句子 5	0	0	1	0	0	0.2
句子 6	0	0	0	0	0	0
句子 7	3	1	3	1	1	1.8
句子 8	1	1	1	0	0	0.6
句子 9	1	0	0	0	0	0.2
句子 10	1	0	0	0	0	0.2
平均	1	0.7	0.9	0.5	0.4	0.7

将上述实验结果的最后平均值，转换成曲线图的形式，得到如图 2.2 所示的结果。

从图 2.2 的结果中可以看出，初期随着 k 值的增大，LDASim 度量的效果有着显著的提升。这是因为 k 值所代表的是潜在主题的数量。如果潜在主题数过少，在将句子或文档转换到潜在主题空间时，其区分度过小，造成很多潜在语义上的差别无法很好的表现出来。当 k 逐渐增大到 500 以上时，LDASim 的效果逐渐趋于稳定，均值能保持在 2 以上。可见只要有足够多的潜在主题，LDASim 能够在很大程度上反映出人们的主观判断结果的。随着 k 值的进一步增大，达到 1000 时，LDASim 的效果反而略有下降。可见潜在主题数并不是越多越好，过多的主题数会使意义分散，反而不利于相似度的度量。在后面的实验中，都取 k 值为 800。

2.3.3 度量比较实验的方法、结果及分析

要考察 LDASim 在实际应用中的效果如何，需要将 LDASim 与传统的度量方法相比较。由于相似度不方便进行直接的比较，于是采用外部评价的办法，即把 LDASim

应用到一个任务中，根据任务执行的效果来评价 LDASim。

表 2.3 五名专家对 20 个文档集进行 k=10 的 LDASim 的评分

	专家 1	专家 2	专家 3	专家 4	专家 5	平均
C1	1	0.7	0.9	0.5	0.4	0.7
C2	0.9	0.4	1.3	0.1	0.3	0.6
C3	0.7	0.7	1.1	0.4	0.8	0.74
C4	0.9	1.2	1.3	0.9	0.7	1
C5	0.9	0.8	0.6	0.8	0.5	0.72
C6	1.5	0.8	0.9	1	0.1	0.86
C7	1.4	0.9	1.4	0.4	0.2	0.86
C8	0.7	0.8	1.3	0.7	0.9	0.88
C9	1.4	0.4	0.5	0.6	0.5	0.68
C10	1.5	0.8	0.7	0.7	0.4	0.82
C11	1.1	0.4	0.5	0.1	0.7	0.56
C12	0.7	0.6	1	0.1	0.2	0.52
C13	1.4	0.3	0.8	0.3	0.1	0.58
C14	0.6	0.5	1.4	1	0.5	0.8
C15	1	0.8	0.5	0.9	0.7	0.78
C16	0.7	0.8	1	0.5	0.2	0.64
C17	1.1	0.3	1.4	0.5	0.2	0.7
C18	0.8	0.7	0.8	1	0.5	0.76
C19	0.6	1.1	0.9	0.5	0.7	0.76
C20	0.7	1.1	1.4	0.2	0.5	0.78
平均	0.98	0.705	0.985	0.56	0.455	0.737

在本文的实验中具体采用的任务是用 MMR 算法提取自动文摘。先对实验方法做一简要介绍，然后给出实验结果。

2.3.3.1 度量比较实验的方法介绍

假设面临的任务是，根据一个指定的主题，从一篇文档中选择若干句子生成文摘。这里的指定主题，通常是一个查询句子，与之前所说的潜在主题不是同一个概念，用字母 Q 表示。生成摘要时，首先遍历整个文档，从中选择一个与 Q 最为相关的句子，将此句子加入到文摘中。然后重新遍历文档，寻找下一个句子。当然寻找的第二个句子也要与 Q 相关。但是由于已经有了第一个句子，那么第二个句子的内容就要尽可能的与第一个句子不同，以避免重复。要生成的是一个长度有限的文摘，在有限的长度中，要尽可能容纳更多的内容，而且这些内容还要与 Q 相关。如何解决好这一矛盾，则是选择句子的关键。

表 2.4 不同 k 值下 LDASim 的评分效果表

	k=10	k=100	k=300	k=500	k=800	k=1000
C1	0.7	1.17	1.9	2.22	1.77	1.71
C2	0.6	1.24	2.13	1.95	2.24	2.05
C3	0.74	1.33	1.9	1.88	1.68	1.95
C4	1	1.08	1.33	1.71	1.49	1.63
C5	0.72	1.54	2.15	2.33	2.15	1.65
C6	0.86	1.77	1.94	2.4	2.8	2.05
C7	0.86	1.27	1.94	1.92	1.99	1.85
C8	0.88	1.11	2.07	2.54	2.49	2.92
C9	0.68	0.85	1.51	1.63	1.49	1.12
C10	0.82	1.53	2.48	2.96	2.86	2.57
C11	0.56	0.83	1.79	2.07	2.31	2.67
C12	0.52	0.52	0.61	0.95	0.92	1.31
C13	0.58	1.17	1.3	1.74	2.11	1.92
C14	0.8	1.57	2.23	2.47	2.46	2.32
C15	0.78	1.03	2.01	2.34	2.71	2.74
C16	0.64	1.55	1.97	1.93	2.05	2.41
C17	0.7	0.7	1.39	1.91	1.94	1.6
C18	0.76	1.03	1.62	1.6	2.08	2.08
C19	0.76	1.62	1.7	1.76	2.11	2.14
C20	0.78	1.67	2.15	2.35	2.72	2.3
平均	0.737	1.229	1.806	2.033	2.1185	2.0495

最大边际相关性（Maximum Marginal Relevance，MMR）算法就是基于这样一种思想：尽可能多的相关，且尽可能少的冗余。MMR 算法主要是依据下面的公式来选择下一个句子：

$$MMR = \arg\max_{S_i \in D \setminus T} \left(\lambda(Sim_1(S_i, Q)) - (1-\lambda)\max(Sim_2(S_i, S_j)) \right) \qquad (2\text{-}25)$$

其中 D 表示输入文档，它是一系列句子构成的集合，即 D={S_1,S_2,…,S_n}；S 是已选出的文摘句的集合；Q 如上文所说，表示一个查询主题，所生成的文摘将围绕这个主题；Sim_1 和 Sim_2 是相似性度量，衡量像个句子之间的相似程度的函数。λ 为平衡系数，属于[0,1]。

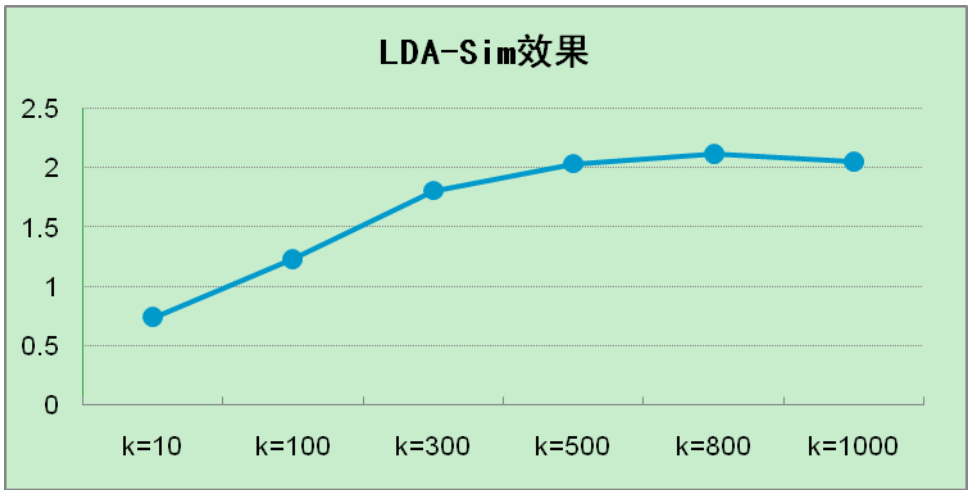

图 2.2 LDASim 在不同 k 值下的效果曲线图

开始设 T 为空集，即文摘中没有任何句子。抽取句子的过程，就是不断地将新的句子加入到 S 中。在每一次抽取时，都选择 MMR 值最大的句子。从 MMR 公式本身即可看出：Sim1 越大，表示句子 Si 与主题 Q 的关联性越大；Sim2 越大，表示句子 Si 与已选文摘句的冗余越大。MMR 就是遵循着前面提到的“尽可能多的相关，尽可能少的冗余”的原则。平衡系数 λ 调节着相关与冗余的比重关系。当 λ=1 时，该算法选择句子时只考虑句子的相关性，忽略冗余性，其结果是所选择的文摘句都与预设主题 Q 高度相关，但是内容重复率很高；当 λ=0 时，则不考虑相关性，尽可能选择不同内容的句子，其结果是所选的文摘句与预设主题 Q 没有关联，却极大限度的覆盖源文档 D 的内容。

要实施 MMR 算法，必须选择恰当的相似性度量方法，即公式中（2.37）中的 Sim_1 和 Sim_2。目前广泛采用的相似性度量方法，就是向量空间模型中的余弦距离。将每个句子表示成单词为维度的空间中的向量，每个维度的值是该单词的词频-反向文档频率值（Term Frequency Inverse Document Frequency，TFIDF）[62]。

这里就可以使用本文定义的 LDASim 度量，来替换原有的度量方式。若将 Sim_1 和 Sim_2 均替换成 LDASim，就得到如下 MMR 公式：

$$LMMR_1 = \arg\max_{S_i \in D \setminus T} \left(\lambda(LDASim(S_i, Q)) - (1-\lambda)\max(LDASim(S_i, S_j)) \right) \quad (2\text{-}26)$$

也可以只替换其中的一个度量，比如替换 Sim_1，则得到以下公式：

$$LMMR_2 = \arg\max_{S_i \in D \setminus T} \left(\lambda(LDASim(S_i, Q)) - (1-\lambda)\max(Sim_2(S_i, S_j)) \right) \quad (2\text{-}27)$$

2.3.3.2 度量比较实验的结果及分析

表 2.5 对 DUC2006 数据集生成摘要的 ROUGH2 评价结果

	MMR	LMMR1	LMMR2
C1	0.0082	0.0088	0.0085
C2	0.0080	0.0090	0.0090
C3	0.0085	0.0084	0.0084
C4	0.0081	0.0089	0.0093
C5	0.0087	0.0083	0.0085
C6	0.0090	0.0088	0.0094
C7	0.0081	0.0085	0.0088
C8	0.0086	0.0083	0.0086
C9	0.0090	0.0085	0.0088
C10	0.0089	0.0085	0.0092
C11	0.0084	0.0086	0.0093
C12	0.0084	0.0087	0.0085
C13	0.0089	0.0084	0.0091
C14	0.0086	0.0087	0.0093
C15	0.0083	0.0090	0.0090
C16	0.0085	0.0089	0.0087
C17	0.0082	0.0091	0.0085
C18	0.0086	0.0082	0.0088
C19	0.0080	0.0089	0.0091
C20	0.0088	0.0090	0.0088
平均	0.0085	0.0087	0.0089

实验中采用三种方式生成文摘：

（1）采用公式（2-25），Sim_1 和 Sim_2 均使用传统的向量空间模型的余弦距离，记为 MMR；

（2）采用公式（2-26），两个度量均使用 LDASim，记为 LMMR1；

（3）采用公式（2-27），第一个度量使用 LDASim，第二个度量使用向量空间模型的余弦距离，记为 LMMR2。

评价的方法采用目前 DUC 会议中通用的 ROUGH 评价工具集中的 ROUGH2 和 ROUGH-SU，参考的标准文摘由 DUC 会议提供。首先对 DUC2006 数据集进行实验，从全部数据中选择 20 个类别进行实验，分别用上述三种方法生成文本摘要，并对生成的摘要进行评价。表 2.5 给出了其 ROUGH2 评价结果，表 2.6 给出了 ROUGH-SU 评价结果。

从以上两个表的结果中可以看出，采用 LDASim 以后，生成文摘的质量有少许提升。这说明 LDASim 度量是有积极效果的。再用同样的方法，对 2007、2008、2009 的数据集做实验，得到的平均结果如表 2.7 和表 2.8 所示。

表 2.6 对 DUC2006 数据集生成摘要的 ROUGH-SU 评价结果

	MMR	LMMR1	LMMR2
C1	0.0136	0.0143	0.0135
C2	0.0142	0.0138	0.0144
C3	0.0137	0.0138	0.0139
C4	0.0139	0.0136	0.0136
C5	0.0140	0.0137	0.0144
C6	0.0134	0.0138	0.0140
C7	0.0137	0.0134	0.0136
C8	0.0136	0.0138	0.0134
C9	0.0141	0.0141	0.0143
C10	0.0142	0.0142	0.0144
C11	0.0142	0.0137	0.0142
C12	0.0134	0.0136	0.0136
C13	0.0141	0.0139	0.0144
C14	0.0136	0.0137	0.0135
C15	0.0143	0.0141	0.0140
C16	0.0134	0.0138	0.0135
C17	0.0135	0.0138	0.0139
C18	0.0139	0.0139	0.0135
C19	0.0139	0.0143	0.0140
C20	0.0135	0.0137	0.0138
平均	0.0138	0.0138	0.0139

表 2.7 三种方法对四个数据集的 ROUGH2 评价结果

Dataset	MMR	LMMR1	LMMR2
2006	0.0085	0.0087	0.0089
2007	0.0100	0.0100	0.0102
2008	0.0076	0.0078	0.0085
2009	0.0084	0.0088	0.0098

从这四个数据集的综合实验结果可以看出，LMMR 比原始的 MMR 效果有明显的提升，这就证明了 LDASim 度量的效果。另外，LMMR2 的效果要好于 LMMR1。LDASim 度量更注重度量对象在内涵上的联系，即使词形不同，但含义相近，会在

LDASim 上有着较好的体现。而传统的向量空间模型的度量，则关注的是表层的关联性，即是否包含相同的单词。在 MMR 公式中的 Sim2 度量，是用于排除冗余。对于冗余，人们多从感官上出发。含有较多的相同单词，则视为冗余；若不含有相同单词，则往往忽略其含义上的冗余性。因此，LMMR2 的测试效果要好于 LMMR1。

表 2.8 三种方法对四个数据集的 ROUGH-SU 评价结果

Dataset	MMR	LMMR1	LMMR2
2006	0.0138	0.0138	0.0139
2007	0.0150	0.0153	0.0165
2008	0.0113	0.0121	0.0128
2009	0.0119	0.0122	0.0127

2.4 本章小结

本章重点讨论了如何度量两个文本单元之间的相似度问题。针对向量空间模型的一些不足，提出了基于概率主题模型的文本相似性度量方法。在对文档集进行 LDA 建模的基础上，定义了潜在主题空间。通过计算单词与潜在主题之间的概率关系，给出了和潜在主题向量的定义。任何一个单词集合均可以表示成潜在主题向量的形式。在此基础上，又提出了 LDASim 相似性度量方法。该方法可以将不同粒度的文本单元（单词、句子、文档、文档集等），映射到同一个欧式空间（即潜在主题空间）中，可以用相同的标准度量不同粒度的文本单元之间的相似程度。由于潜在主题的数量可以人为控制，且每个单词与每个潜在主题之间都有一定的概率关系，这就使得 LDASim 避免了传统的向量空间模型中常见的稀疏问题，又能够在较深的层面上挖掘语义关系。

最后的实验分为两部分进行。第一部分是测定潜在主题空间的维度 k 的取值，验证不同的 k 值对模型效果的影响。第二部分是验证本文提出的 LDASim 的效果。采用 MMR 算法进行自动文摘抽取，在算法中使用不同的相似性度量。通过对 DUC2006、2007 和 TAC2008、2009 四个数据集上进行了自动文摘抽取实验，其结果证明 LDASim 比传统的向量空间模型的余弦度量效果更好。

第3章 基于差分进化和概率主题模型的句子抽取算法

3.1 引言

句子抽取技术，是自动文本摘要技术的核心。如何从原文档集中抽取句子、抽取哪些句子、抽取句子应该有什么样的标准，这是句子抽取技术需要回答的问题。抽取出的句子是要组成文摘的，人们希望自动生成的文摘能够全面的反映原文的内容，又能够突出原文的重点，同时含有较少的冗余。而实现这些都要依赖于句子抽取技术。在当前的多文档文摘的研究中，句子的抽取一般有两种方法：聚类法和排序法。

基于聚类法的自动文摘系统，是对相似的句子进行聚类，每个聚类代表了原文档集的一个逻辑主题。然后在各个逻辑主题中抽取句子生成文摘[97]。这种方法生成的文摘可以降低冗余度，提高文摘的覆盖率。由于该方法对文档集合理解不仅停留于浅层，而是从深层的逻辑结构理解文档集合，从而使文摘的质量更高，成为目前多文档文摘研究的主流。但是因为要照顾到所有的逻辑主题，往往使生成的文摘不能很好的突出原文的重点。而且这种方法需要事先确定多文档集的逻辑主题数量。多文档集合的逻辑主题数量是根据内容的紧凑程度确定，一般是未知的。常见的聚类方法，都有一定的不足，基于划分聚类方法须要事先知道类别数，基于层次的方法可以不需要事先知道类别数，终止条件可以通过阈值判断，但该方法不足是聚类过程不能回溯，一旦确定对象的所属类别就不能更改。根据处理对象的特点选择合适的经过改进的聚类方法，该问题可以在一定程度上得到解决。

目前绝大多数的聚类法自动文摘系统中，文档都是使用向量空间模型中的向量来表示的，把文档都看做是一个词袋（bag of words）。每一个文档都被表示成 m 维空间中的一个向量。这样的表示，通常维度都很高，这对聚类算法的表现是个巨大的挑战。对很高维度下的稀疏向量，聚类算法很难有好的表现[100]。如果把这种算法应用到句子聚类中，有明显的缺点：句子的表示非常低效，向量维度 m 相对于句子中的单词数而言太大了。

Aliguliyev 于 2009 年提出的基于 NGD[102]的聚类抽取算法[101]，有效的避免了向量空间模型中的稀疏问题。但是，该方法也有着自身的缺点，那就是对谷歌搜索引擎的依赖。另外，该方法也有着聚类法所固有的缺点，由于要从每个聚类里抽取句子，保

证内容的全面性，却造成了原文的重点在文摘中并不突出的情形。

在排序类的多文档文摘系统中，一般是按照某种方式，计算句子与原文档集的主题之间的相关程度，按相关程度的大小对句子进行排序，直接根据排序的结果进行句子抽取。例如，比较著名的方法是卡耐基梅隆大学的 Jade Goldstein 等人提出的基于最大边际相关性（Maximal Marginal Relevance，MMR）的多文档自动文摘方法[26]，通过 MMR 方法做文摘，将与主题相关、而句子之间不相似的句子保存在文摘中，从而达到去除冗余信息的目的，这种方法主要适用于问题相关的多文档文摘。另一个比较典型的例子是密西根大学 Redev 提出基于质心的多文档自动文摘方法[100]，首先以词为研究单元，以原文档中的高频词组成伪句子，以其为质心，按照句子与质心的相似程度、句子的位置、句子与首句的相关程度对句子进行打分，根据分数对句子排序，按照排序结果抽取出适当的句子，组成文摘。最著名的排序类文摘系统，要数 SumBasic 系统[91]。该系统是 Nenkova 等人深入研究了词频与人手工文摘之间的关系[93][94][95][96][97]后研制了的，是当时学术界公认的效果最好的自动文摘系统。

SumBasic 的一个最突出的特点是对单词的权重进行动态调整，以使冗余最小。其权值调整公式如（3.1）所示。

$$p_{new}(w_i) = p_{old}(w_i) \cdot p_{old}(w_i) \qquad (3\text{-}1)$$

权重调整发生在抽取过程之中。其中 w_i 为当前抽取的句子中的单词，$p_{old}(w_i)$ 为每个单词在抽取之前的权重，$p_{new}(w_i)$ 为抽取之后更新的权重。

这种动态的权值调整，使得“什么才是最重要的要包含到文摘中的信息？”这一问题的答案随着文摘抽取的进行而不断发生变化。可以这样理解：$P_{old}(w_i)$可以看做是单词 wi 应该包含在文摘中的概率，而 $P_{new}(w_i)$则是单词 w_i 应该在文摘中出现两次的概率。通过这样一种方式来调整概率，使得初始概率较低的一些词也有机会对句子选择产生较大影响。而且，这种单词概率的更新提供了一种很自然的处理冗余的方式，没有必要再去检查单词的重复了。

在实践中，有人对 SumBasic 模型进行了改进，采用二元组（Bigrams）代替单词，称为 SumBasic+。再对权重更新增加一个启发式算法，称为 SumBasic++。这两个系统，至少在 ROUGH 评价上，有着非常出色的表现。

但是 SumBasic 系统也有其自身的缺点，也是排序类抽取方法所固有的缺点，那就是陷入句子的细节评价之中，而忽略了文摘的整体性。

Daniel 于 2011 完成的博士论文[87]中，对现有的自动文摘的抽取方法提出了质疑——“目标始终没有定义，SumBasic 只是一个过程。它运作的很好，但是它并没有定义一个好文摘的模型”。此前人们对于自动文摘的研究，大多集中于过程——用什么样的过程来进行抽取。而现在，人们则更希望从过程转移到目标——究竟什么样的文摘是好的，是否存在好文摘的数学模型。

在 SumBasic 中，一篇文摘有一个最终的值 S，它简单的表示为其所包含的单词的权重和。在冗余处理上，SumBasic 用一种隐式的约束条件取代了之前的相似性度量方式（如 MMR 方法中公式 2-25 的 Sim2）。这其实就是对文摘的一种简单的模型描述，记为：

$$S = \sum_i w_i c_i \tag{3-2}$$

其中 c_i 表示第 i 个单词（或者更一般的，称为语言单元或 n 元组）在文摘中的出现；w_i 则为 c_i 的权重。

现在，假设给定一个权重集，将要在长度约束 L 的条件下，寻找一个文摘，使得目标函数最大化。因为要做的是抽取式的文摘，长度约束则可以根据选择的句子来描述。句子由下标 j 来索引，sj 表示句子 j 在文摘中的出现情况，lj 表示该句子的长度（单词数）。则自动文摘的目标可以定义为：

$$\begin{aligned} \text{Maximize} \quad &: \quad \sum_i w_i c_i \\ \text{Subject to} \quad &: \quad \sum_j \mathrm{l}_j s_j \le L \end{aligned} \tag{3-3}$$

这一目标可以理解为带权重的最大覆盖问题。输入空间包含权重化的单位，划分成句子的重叠单元。要找到一个句子的集合，在满足句子总长度这一约束条件下，尽可能的覆盖更大的空间。这一最大覆盖公式以前曾有过介绍[88]，但是其关注的焦点不是实用的目标函数，而是实体抽取过程。

对这一问题寻求最好的解决方案，似乎需要一个输入句子数的指数级算法，也就是著名的集合覆盖问题[89]。或者可以看做是 NP 完全的背包问题：使背包中的物品价值最大化，每个物品有一个价值和一个重量，而背包有重量限制。如果假设句子之间没有重复的单词，那么这就是完全是一个背包问题（句子是物品，摘要是背包）。更一般的情况，句子之间有共享的单词，可以简化为背包问题。如果多项式时间内可以解决文摘的问题，那么在多项式时间内也可以解决背包问题，因此文摘的问题也是

NP 完全问题。

按照上面的目标，生成文摘的最简单的方法，就是每次选择使目标值最大的句子。具体有两种方式：（1）确实选择单词权重之和最高的句子；（2）用句子长度对单词权值进行泛化。在实验证明，第二种方式能产生总体目标价值更高的文摘，因为第一种方式总是倾向于选择较长的句子。

如果把目标公式看做一个整数线性规划（Integer Linear Program，ILP）问题[90]，那么就可能得到一个精确的解决方法。虽然整数线性规划的解决方法是输入规模的指数级，不过人们已经对很多问题进行了优化，给出了更快速的解决方法。这里给出目标的完整 ILP 描述：

$$\begin{aligned} &\text{Maximize} \quad : \quad \sum_i w_i c_i \\ &\text{Subject to} \quad : \quad \begin{cases} (1) \quad \sum_j s_j l_j \le L \\ (2) \quad s_j Occ_{ij} \le c_i, \forall i, j \\ (3) \quad \sum_j s_j Occ_{ij} \ge c_i, \forall i \\ (4) \quad c_i, c_j \in \{0,1\}, \forall i, j \end{cases} \end{aligned} \tag{3-4}$$

公式（3-4）中的（1）此前已给出，（2）和（3）是确保句子和其单词保持一致的结构性约束。Occ 是一个矩阵，Occ_{ij} 表示单词 i 在句子 j 中的出现。其中（2）保证了选择一个句子就选择它包含的所有单词；（3）保证了一个单词被选择当且仅当它出现在至少一个被选择的句子当中。

3.2 基于差分进化的句子聚类方法

基于聚类的句子抽取算法，其基本思想是先将句子聚类形成主题，然后从不同的主题中抽取句子。这种方法有效的保证了抽取出的句子对原文内容的全面覆盖，因此成为目前句子抽取算法的主流。

本节提出了一种基于差分进化的句子聚类方法。首先采用表层距离和语义距离来定义句子之间的距离，接着通过差分进化算法对句子进行聚类。

3.2.1 表层距离与语义距离

句子之间的距离，是根据句子中包含的单词来定义的。首先来定义单词之间的距

离。

给定文档集 C，其中任意两个单词 w_k 和 w_l 之间的表层距离定义为：

$$diss_{surf}\left(w_k,w_l\right)=\frac{\max\left\{\log\left(f_k^{local}+1\right),\log\left(f_l^{local}+1\right)\right\}-\log\left(f_{kl}^{local}+1\right)}{\log(n+1)-\min\left\{\log\left(f_k^{local}+1\right),\log\left(f_l^{local}+1\right)\right\}} \tag{3-5}$$

其中 f_k^{local} 表示文档集 C 中包含 w_k 的句子的数量，f_l^{local} 表示文档集 C 中包含 w_l 的句子的数量。f_{kl}^{local} 表示文档 C 中同时包含 w_k 和 w_l 的句子的数量。n 表示文档集 C 中包含的句子总数。

如果两个单词同时包含在一个句子中的次数越多，那么这两个单词的距离就越近。反之则越远。当 k=l 时，也就是计算一个单词与其自身的表层距离，这时有 $f_k^{local}=f_l^{local}=f_{kl}^{local}$，因此 $diss_{surf}\left(w_k,w_l\right)$=0，单词与其自身的距离为 0。若句子总数 n=1，则会出现 $diss_{surf}\left(w_k,w_l\right)=\frac{0}{0}$ 的情况，此时定义该值为 0。

将两个句子 S_i 和 S_j 所包含的所有单词之间的表层距离做求和平均，可以得到两个句子的表层距离，即两个句子 S_i 和 S_j 之间的表层距离定义为：

$$diss_{surf}\left(S_i,S_j\right)=\frac{\sum_{t_k\in S_i}\sum_{t_l\in S_j}diss_{surf}\left(w_k,w_l\right)}{m_i m_j} \tag{3-6}$$

其中 m_i 和 m_j 分别表示句子 S_i 和 S_j 包含的单词数。

表层距离仅仅是根据单词在句子中的包含关系来定义的一种距离，需要更进一步的挖掘句子之间的语义关系。因此，采用第二章介绍的 LDASim 来计算两个句子之间的语义距离。

首先对文档集 C 进行 Dirichlet 建模，有

$$p(w\mid\alpha,\beta)=\int_\theta p(w\mid\theta,\beta)p(\theta\mid\alpha)d\theta \tag{3-7}$$

其中，

$$p(\theta\mid\alpha)=\frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\prod_{k=1}^{K}\theta_k^{\alpha_k-1} \tag{3-8}$$

$$p(w\mid\theta,\beta)=\prod_{t=1}^{V}\left(\sum_{k=1}^{K}p(z_t=k\mid\theta)p(t\mid z_t=k,\beta)\right) \tag{3-9}$$

因而公式为：

$$P(w|\alpha,\beta)=\frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\int_{\theta}\prod_{k=1}^{K}\theta_k^{\alpha_k-1}\prod_{t=1}^{V}\left(\sum_{k=1}^{K}P(z_t=k|\theta)P(t|z_t=k,\beta)\right)d\theta \quad (3\text{-}10)$$

记 $P(z_t=k|\theta)=\theta_k$ 和 $P(t|z_t=k,\beta)=\beta_{kt}$，则上式可以改写为：

$$P(w|\alpha,\beta)=\frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\int_{\theta}\prod_{k=1}^{K}\theta_k^{\alpha_k-1}\prod_{t=1}^{V}\left(\sum_{k=1}^{K}\beta_{kt}\theta_k\right)d\theta \quad (3\text{-}11)$$

这里的积分是定义在狄利克雷分布上，可以采用变分推理和吉布斯采样的方法来进行参数估计，得到 α 和 β。

则对文档集 C 中的任意两个单词 w_i 和 w_j，可得

$$\begin{aligned}&LDASim(w_i,w_j)\\&=\cos(\overrightarrow{V(w_i)},\overrightarrow{V(w_j)})\\&=\frac{P(z_1|w_i)P(z_1|w_j)+\cdots+P(z_k|w_i)P(z_k|w_j)}{\sqrt{P(z_1|w_i)^2+\cdots P(z_k|w_i)^2}\cdot\sqrt{P(z_1|w_j)^2+\cdots+P(z_k|w_j)^2}}\end{aligned} \quad (3\text{-}12)$$

于是定义两个单词 w_i 和 w_j 之间的语义距离为：

$$diss_{sema}(w_i,w_j)=1-LDASim(w_i,w_j) \quad (3\text{-}13)$$

再将两个句子 S_i 和 S_j 中包含的单词的语义距离进行加和平均，可以定义句子 S_i 和 S_j 之间的语义距离为：

$$diss_{sema}\left(S_i,S_j\right)=\frac{\sum_{w_k\in S_i}\sum_{w_l\in S_j}diss_{sema}\left(w_k,w_l\right)}{m_i m_j} \quad (3\text{-}14)$$

同样 m_i 和 m_j 表示句子 S_i 和 S_j 中包含的单词数。

最后，定义两个句子 S_i 和 S_j 之间的距离为表层距离和局部距离的乘积，即：

$$diss\left(S_i,S_j\right)=diss_{surf}(S_i,S_j)\cdot diss_{sema}(S_i,S_j) \quad (3\text{-}15)$$

3.2.2 基于差分进化的句子聚类

本节采用一个改进的离散差分进化算法对文档集中的句子进行聚类。在进化算法

中，需要对染色体进行编码，确定适应函数和交叉变异的规则，设置好初始条件和终止条件，进过多次的迭代得到最终的聚类结果。

3.2.2.1 染色体编码

种群中的每个染色体，代表一种可能的聚类方案。每个染色体中包含 n 个基因，n 是文档集中的句子总数。每个基因的值表示该句子被划分到的类别编号。设有 k 个聚类，则每个基因的取值范围为[1,k]。例如，若将 10 个句子划分成 3 类，即 n=10 且 k=3，染色体的编码为[2,3,1,1,2,1,2,3,3,2]，表示第三、四、六号句子属于聚类 1，第一、五、七、十号句子属于聚类 2，第二、八、九号句子属于聚类 3。

假设当前为第 t 代，则第 a 个染色体可以表示为：

$$X_a(t)=\left[x_{a,1}(t),x_{a,2}(t),\cdots,x_{a,n}(t)\right] \tag{3-16}$$

其中 $x_{r,s}(t)$ 是正整数且 $x_{r,s}(t)\in\{1,2,\cdots,k\}$。$a=1,...,\text{Pop}$，Pop 表示整个种群的规模。

3.2.2.2 适应函数

为了评估一个染色体所代表的聚类结果的质量，必须要有适应函数。适应函数是要保证两点：（1）同一聚类中的句子之间的距离尽可能小；（2）不同聚类之间句子的距离尽可能大。

公式（3-15）已经给出了句子之间距离的计算方法，则判定同一聚类中的句子之间的距离尽可能小的函数为

$$F_1=\sum_{p=1}^{k}\left|C_p\right|\sum_{S_i,S_j\in C_p}diss\left(S_i,S_j\right)\to\min \tag{3-17}$$

和

$$F_2=\sum_{p=1}^{k-1}\sum_{q=p+1}^{k}\sum_{S_i\in C_p}\sum_{S_j\in C_q}\left|C_p\right|\left|C_q\right|diss\left(S_i,S_j\right)\to\max \tag{3-18}$$

其中$|C_p|$是加入到聚类 C_p 中的句子数量。

则最终的适应函数定义如下：

$$F=\frac{\sum_{p=1}^{k}\left|C_p\right|\sum_{S_i,S_j\in C_p}diss\left(S_i,S_j\right)}{\sum_{p=1}^{k-1}\sum_{q=p+1}^{k}\sum_{S_i\in C_p}\sum_{S_j\in C_q}\left|C_p\right|\left|C_q\right|diss\left(S_i,S_j\right)}\to\min \tag{3-19}$$

对于每一代新产生的染色体，都使用公式（3-19）进行度量。如果新染色体使 F 值更小，则保留新染色体；否则淘汰新染色体，保留其父辈。

3.2.2.3 交叉和变异

设 $X_b(t)$为当前种群（第 t 代）中最好的染色体。从当前种群中随机选择两个其他的染色体 $X_u(t)$和 $X_v(t)$（$b,u,v \in \{1,2,\cdots,\mathrm{Pop}\}$且$b \neq u \neq v$）。则对 t+1 代染色体 $Y_b(t+1)$中的第 s 个基因，有

$$y_{b,s}(t+1)=\begin{cases} x_{b,s}(t)+\pi_s x_{u,s}(t)-(1-\pi_s)x_{v,s}(t) & \text{if } rnd_s < CR \\ x_{b,s}(t) & otherwise \end{cases} \tag{3-20}$$

其中 rnd_s 和 π_s 都是[0,1]区间上的均匀分布的随机值，对于每一个基因 s 会分别取值一次。这样得到的 y 值是实数，通过下式将其转换为整数：

$$y_{b,s}(t+1)=\begin{cases} INT(k\cdot rnd_s+1) & \text{if } INT(y_{b,s}(t+1))<1 \text{或} INT(y_{b,s}(t+1))>k \\ INT(y_{b,s}(t+1)) & otherwise \end{cases} \tag{3-21}$$

其中 INT 为去尾法取整函数。

为了保证种群数量的恒定，在新生儿 Y 和其父辈 X 中，只能保留一个。根据适者生存的原则，使用公式（3-19）进行衡量，淘汰适应能力较差的染色体，即：

$$X_b(t+1)=\begin{cases} Y_b(t+1) & \text{if } F(\mathrm{Y}_b(\mathrm{t+1}))>F(\mathrm{X}_b(\mathrm{t})) \\ X_b(t) & otherwise \end{cases} \tag{3-22}$$

根据适应函数的评价结果，如果新生儿的适应能力更强，则在新一代种群中，用新生儿代替其父辈；如果相反，则在新一代种群中保留父辈。

对染色体 $X_b(t)$的变异，则有下面公式给出：

$$y_{b,q}=\begin{cases} x_{b,q}(t)+\pi_q x_{b,r}(t)-(1-\pi_q)x_{b,s}(t) & \text{if } rnd_q < MR \\ x_{b,q}(t) & otherwise \end{cases} \tag{3-23}$$

其中，q,r,s 是染色体基因的随机下标，$q,r,s \in \{1,2,\cdots,n\}$且$q \neq r \neq s$。$MR \in [0,1]$为预定义好的变异概率。$rnd_q$ 和 π_q 都是[0,1]区间上的均匀分布的随机值，对于每一个基因 q 会分别取值一次。

同公式（3-21）类似，公式（3-23）也要进行整数化，即：

$$y_{b,q}(t+1)=\begin{cases} INT(k\cdot rnd_q+1) & \text{if } INT(y_{b,q}(t+1))<1 \text{或} INT(y_{b,q}(t+1))>k \\ INT(y_{b,q}(t+1)) & otherwise \end{cases} \quad (3\text{-}24)$$

同样的，根据适应函数的结果进行选择。如果经过变异的染色体的适应能力更强，则在新一代种群中，用变异结果代替其父辈；如果相反，则在新一代种群中保留父辈。

3.2.2.4 初始和终止条件

首先要确定聚类的个数 k、种群的规模 Pop、交叉概率 CR、变异概率 MR 等常量，这些数值的确定则根据具体的数据有关。

接下对种群进行初始化，即确定第 0 代种群中每个染色体中每个基因的值。采用随机的方式对第 0 代种群赋值。设第 0 代种群中第 a 个染色体为 $X_a(0)$，则

$$X_a(0)=[x_{a,1}(0),x_{a,2}(0),\cdots,x_{a,n}(0)] \quad (3\text{-}25)$$

对其中第 r 个基因 $x_{a,r}(0)$，有

$$x_{a,r}(0)=k_r\cdot sigm(k_r)+1 \quad (3\text{-}26)$$

这里 k_r 是区间[1,k]中均匀分布的随机值，对每一个不同的 r 取值一次。sigm()是 s 形函数，它可以将实数映射到[0,1]，其数学公式如下：

$$sigm(z)=\frac{1}{1+\exp(-z)} \quad (3\text{-}27)$$

初始化做好之后，就可以进行进化迭代了。每一代的种群由上一代种群生成，根据公式（3-21）（3-24）进行交叉和变异，采用适应函数（3-19）对新生染色体进行评价，根据公式（3-22）进行个体淘汰。

使用最大迭代数作为算法的终止条件，也就是说，当种群繁衍到 t_{max} 代时，算法终止。如果在此之前，种群已趋于稳定，也就是连续很多代没有变化，也可以作为算法终止的条件。算法终止时，选择适应能力最强的染色体，即为想要的聚类结果。

3.3 基于概率主题模型的逐句递减规则

在概率主题模型下，文档被看做是主题的混合物，文档可以表示为主题的概率分布。通过这些主题，建立潜在主题空间。把句子集合表示成潜在主题空间中的向量，

使用潜在主题空间中的距离来计算句子的权重。在进行句子抽取时，将关注的目标锁定在文摘与原文档集的相似程度上。随着抽取的进行，动态调整向量表示，从而动态调整句子的权重。

3.3.1 概率主题模型下句子权值的计算

首先对文档集 C 进行 Dirichlet 建模，通过变分推理估计出参数 α 和 β。对于文档集 C 中的每一个单词 w，有

$$p(w|\theta,\beta)=\sum_{z}p(w|z,\beta)p(z|\theta) \tag{3-28}$$

其中 $\theta \sim Dirichlet(\alpha)$，即

$$p(\theta|\alpha)=\frac{\Gamma(\sum_{i=1}^{k}\alpha_i)}{\prod_{i=1}^{k}\Gamma(\alpha_i)}\theta_1^{\alpha_1-1}\cdots\theta_k^{\alpha_k-1} \tag{3-29}$$

在确定了参数 α 以后，通过一次 Dirichlet 采样，可以得到主题的概率分布参数 θ，由 θ 和 β 可以确定一个单词的概率 p(w)。

对于一个句子 S_j，其静态权重可以定义为：

$$weight_{static}(S_j)=\sum_{w_i\in S_j}\frac{p(w_i)}{\left|\{w_i\,|\,w_i\in S_j\}\right|} \tag{3-30}$$

之所以称之为静态权重，是因为该权重所涉及的参数 α 和 β 是文档集级的。对一个文档集进行建模之后，α 和 β 就会被估计出来。进行 Dirichlet 采样，确定主题的分布后，每个句子的权值也随之确定了。在对句子进行抽取操作时，该权值不会改变。

这样的权值确定方式，与传统的文摘算法中的权值类似。其缺点是容易造成摘要句之间的同质性，使冗余过大。因此希望能够在抽取的过程中动态调整权值，也就是让句子权值随着已抽取的句子的变化而变化。

句子的权值是抽取而设定的，其基本思想是“权值越大的句子，在文摘中出现的概率越高”，与这个命题等价的逆否命题是“在文摘中出现的概率越低的句子，其权值越小”。这就给出了一个启示，可以从另外一个方向来思考。

对于文档集 C 中的一个句子 S_i，若将 S_i 从文档集 C 中去掉，对原文档集 C 的含义几乎没有影响，那么 S_i 应该不能表现出文档集 C 的主旨，其在文摘中出现的概率也应该较低。也就是说，从 C 中去掉 S_i 后剩余的部分，与 C 的相似程度越高，则说明

S_i的权值应该越低。所以，动态赋予一个句子权值的方法就是，看去掉这个句子之后，保留下来的句子是否依旧能反映原文内容。

在第二章，提出了相关性度量 LDASim。LDASim 是一个通用的文本度量方法，适用于各种粒度的文本单元。因此可以在这里使用 LDASim 来度量保留下来的句子与原文档集的相关程度。

对于文档集 C 一篇文摘 T 中的一个句子 S_i，其动态权重定义为

$$weight_{dyna}(S_i) = 1 - LDASim(C, T - \{S_i\}) \tag{3-31}$$

最后定义一个句子 S_i 在概率主题模型下的权重为其静态权重和动态权重的乘积，即

$$weight(S_i) = weight_{static}(S_i) \cdot weight_{dyna}(S_i) \tag{3-32}$$

有了句子权重的计算方法，就可以依据句子的权重大小，从原文档集中抽取句子组成文摘。

3.3.2 逐句递减规则

本节来研究抽取句子的具体规则，原则是从整体上把握文摘与原文档集的关系。将一篇文档看做是一个句子的集合，即 $D=\{S_1,S_2,\ldots,S_n\}$。而一个文档集是文档的集合，则包含 m 个文档的文档集 C 可以表示为：

$$\begin{aligned} C &= \{D_1, D_2, \ldots, D_m\} \\ &= \{S_{11}, S_{12}, \ldots, S_{1n_1}, S_{21}, S_{22}, \ldots S_{2n_2}, \ldots, S_{m1}, S_{m2}, \ldots, S_{mn_m}\} \end{aligned} \tag{3-33}$$

可见 C 依旧是个句子的集合。

假设 T 为目标文摘，也就是系统生成的自动文摘。那么 T 本身也是一篇文档。因此，文摘 T 也可以按照上述方式表示成句子集合的形式，即 $T=\{S_1,S_2,\ldots,S_k\}$。由于采取抽取的方法生成文摘，所有的文摘句均来源于原文档集，因此，文摘 T 是文档集 C 的一个子集，即 $T \subseteq C$。

对于集合 C 来说，C 的非空子集一共有 $2^{|C|}$-1 个。那么这 $2^{|C|}$-1 个子集均可以作为文摘 T 的候选。在第一章的 1.2.1 节，给出了文摘的定义。从这个定义中可以看出，文摘具有三个特点：可读性、概括性、压缩性。由于文摘 T 中所有句子均来自原文，

因此可以认为这些句子都是符合语法的，可以被人们阅读理解的，即采用句子抽取方法生成的文摘 T 本身就具有可读性（就可读性而言，除了每个句子本身符合语法以外，还有一个重要的问题是句子的组织顺序，这个问题将在下一章着重讨论）。那么在抽取算法中重点要考虑的就是两个方面：概括性和压缩性。

所谓概括性，就是文摘尽可能多的反映原文的内容，即文摘 T 与原文档集 C 的相似性较高。所谓压缩性，就是文摘的长度有一定限制。那么，自动文摘抽取问题就可以描述为：

求文档集 C 的一个非空子集 T，使得

$$\begin{aligned} &\text{Maximize} \quad : \quad SIM\left(C,T\right) \\ &\text{Subject to} \quad : \quad |T| \le \psi \end{aligned} \tag{3-34}$$

其中 SIM(C,T)表示文档集 C 与文摘 T 的相似度。|T|表示文摘 T 的长度，ψ 为最大长度阈值。

传统的方法中，没有对式（3-34）中的 *SIM*(*C,T*)进行直接的度量，而是通过对具体句子的评分来实现的。用某种方式评价句子与原文档集的相关程度，并进行打分，抽取分值高的句子组成文摘。然而事实上，抽取分值高的句子这一方法，并不能很好的模拟 SIM(C,T)。这一点可以通过下面的实验的验证。

取一个文档集 C，用包含单词的词频权重的方法对 C 中的句子进行评分，得到评分最高的句子 S_m。然后用 SumBasic 方法对 C 生成自动文摘 T_1，当然，其中包含句子 S_m。再找两位专家，甲和乙。要求两位专家用抽取句子的方式手工生成 C 的摘要。其中，甲生成的摘要中，必须含有句子 Sm，记为 T_2；乙生成的摘要中，必须不含有句子 S_m，记为 T_3。另找五位专家，对这三篇摘要进行评价，评价分为三个等级：非常好、可接受、不可接受（参见 2.4.2 节）。得到实验结果如表 3.1 所示。

表 3.1 五名专家对三篇文摘的评价结果

	专家 1	专家 2	专家 3	专家 4	专家 5
T1	可接受	不可接受	可接受	不可接受	可接受
T2	非常好	可接受	非常好	非常好	可接受
T3	非常好	非常好	非常好	非常好	可接受

从实验结果可以看出，尽管文摘 T_1 包含了句子 S_m，但其文摘效果并不理想。而 T_2 和 T_3 均为人工生成，效果较好。尽管 T_3 不包含句子 S_m，T_3 的评分效果还是略好于 T_2。这说明是否包含评分最高的句子 S_m 并不是一个好文摘的充分必要条件。即使不包含句子 S_m，也可能生成比包含 S_m 更好的文摘。上面的实验说明，句子评分的方式有其局限性。若过分关注于句子这一细节单位，对整体的文摘表现则有所忽略。

本文摒弃了抽取分值高的句子这种传统的抽取思路，而改为从文摘全文的角度进行把握的方法。如果将文档集 C 的所有非空子集，都作为文摘 T 的候选。从中选出一个与原文内容最相关的子集，那么毫无疑问，这个子集就是 C 本身。不考虑长度因素，仅从相关性上看，文档集本身就是最好的摘要。当然，长度因素是必须要考虑的。因此本文提出的逐句递规则的基本思想就是，从文档集本身出发，逐步去掉其中权值较低的句子，直到其长度达到指定的要求为止。

3.4 差分进化和概率主题模型相结合的句子抽取算法

前面介绍了基于差分进化的句子聚类方法和基于概率主题模型的逐句递减规则。基于差分进化的句子聚类方法，把句子划分成了不同的主题，这可以保证在进行句子抽取时对原文内容有良好的覆盖。基于概率主题模型的句子递减规则，将关注的目标锁定在文摘与原文档集的相似程度上，使得生成的文摘会突出原文的重点。

将以上两者结合起来，采取先聚类再逐句递减的方式，提出差分进化和概率主题模型相结合的句子抽取算法（Differential Evolution and Probabilistic Topic Models，DEPTM）。

算法 3-1 差分进化和概率主题模型相结合的 DEPTM 句子抽取算法

步骤 1. 文本预处理。对给定文档集 C 做预处理，包括去除停用词、词根还原等工作。之后将文档集表示成文档的集合，每篇文档表示成句子的集合，每个句子表示成单词的集合。

步骤 2. 概率主题建模。采用第二章的方法，对文档集进行建模。经过变分推理，估计出 α 和 β。进行狄利克雷采样，以确定潜在主题的分布概率。将不同粒度的文本单元都表示潜在主题空间中的向量形式。计算出 C 中每个句子的静态权重 $weight_{static}(S_i)$。

步骤 3. 确定种群规模 Pop、聚类个数 k、交叉概率 CR、变异概率 MR 等常量。

生成初代种群。

步骤 4. 进行新一代的种群繁衍。根据公式（3-21）（3-24）进行交叉和变异，采用适应函数（3-19）对新生个体进行评价，根据公式（3-22）进行个体淘汰。

步骤 5. 重复步骤 4，直到达到最大迭代数 t_{max} 或种群趋于稳定。取最终代种群中适应性最强的个体作为句子聚类结果。

步骤 6. 设 T 为目标文摘，令 T=C。

步骤 7. 根据公式（3-31）计算 T 中每个句子的动态权重 $weight_{dyna}(S_i)$，并根据公式（3-32）计算出每个句子的权重 $weight(S_i)$。

步骤 8. 对所有句子个数大于 1 的聚类，将其中的句子按 $weight(S_i)$ 进行统一排序，找出 $weight(S_i)$ 最小的句子 S_{min}。若所有聚类中的句子个数均小于或等于 1，则将所有剩余句子排序，找出权值最小的句子 S_{min}。

步骤 9. 将句子 S_{min} 从 T 中去除，即令 $T=T-\{S_{min}\}$。同时将 S_{min} 从其所在聚类中去除。

步骤 10. 若 $|T| \leq \psi$，则算法结束。否则，跳转到步骤 7。

从算法的步骤 8 可以看出，在去掉句子的过程中，从句子和文档两个层面来保证生成文摘的质量。在句子层面，要始终保证，保留在文摘中的句子要对原文的内容有充分的覆盖，不能让所有的文摘句都只集中于原文的部分内容。为了做到这一点采用聚类的方法。将原文档集中的句子进行聚类，在去掉句子时要保证每个聚类都不为空。在文档层面，要始终保证，保留下来的文摘在整体上与原文档内容最为接近。为了做到这一点，采用动态权重来约束文摘与原文档集的相关程度。

这其中有一个例外，就是剩余句子数已经等于聚类个数，每类恰好剩一个句子，但是总长度还达不到阈值 ψ 的要求。这种情况的出现，通常是因为聚类数量设置不当引起的，应该重新设置聚类数量后重新执行算法。若要维持现有的聚类数量，则到算法最后出现该情况时，忽略掉聚类判断条件，以便算法可以继续执行下去。

3.5 实验及结果分析

在实验中使用准确率、召回率、ROUGH-2、ROUGH-SU 四种指标来测试 DEPTM 算法的有效性。实验数据采用 DUC2006 数据集。

要测试准确率和召回率，首先请三位专家进行手工抽取，生成三篇文摘 Tm1、

Tm2、Tm3。令 Tm 表示这三篇文摘中所有句子的集合，即 Tm=Tm1∪Tm2∪Tm2。设 Ta 是系统生成的文摘，则 Ta 的准确率可以表示为：

$$P = \frac{|T_m \cap T_a|}{|T_a|} \tag{3-35}$$

召回率可以表示为：

$$R = \frac{|T_m \cap T_a|}{|T_m|} \tag{3-36}$$

实验中使用三种方法生成自动文摘：2009 年提出的基于 NGD 的方法[101]、公认的效果最好的自动文摘系统 SumBasic 系统[91]、本文提出的 DEPTM 句子抽取算法。分别对 DUC2006 数据集中的 10 个主题进行自动文摘抽取（D0601 到 D0610）。图 3.1 给出了三种方法的准确率，图 3.2 给出了三种方法的召回率。

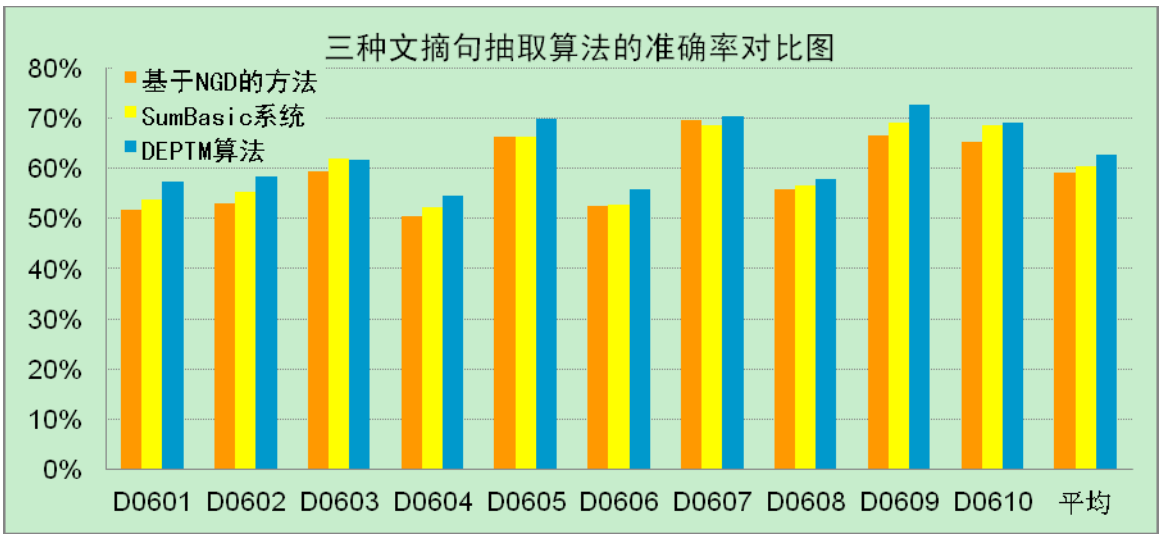

图 3.1 三种句子抽取算法的准确率对比

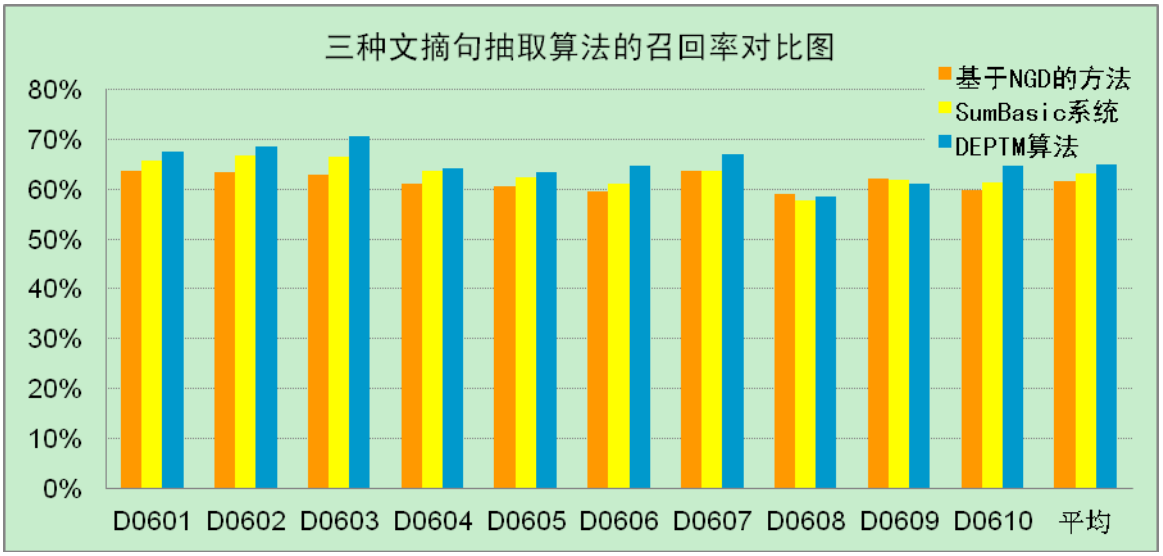

图 3.2 三种句子抽取算法的召回率对比

从以上的对比可以看出，本文提出的 DEPTM 算法相比基于 NGD 的方法和 SumBasic 系统，在准确率和召回率上都有明显的优势。这是因为人们在手工生成文摘时，考虑的是文摘整体是否能够概括原文的意思，而不是考虑的具体某个句子含有多少信息量。DEPTM 算法正是从这一角度提出的，其生成文摘的过程始终保证了文摘整体与原文的相关性，因此其效果与手工文摘更加接近。

表 3.2 DEPTM 与其他系统的 ROUGH-2 结果对比

	ROUGH-2 均分	评分范围
DEPTM	0.09573	0.08532 - 0.10037
系统 24	0.09558	0.09144 - 0.09977
系统 15	0.09097	0.08671 - 0.09478
系统 12	0.08987	0.08583 - 0.09385
系统 8	0.08954	0.08540 - 0.09338
系统 23	0.08792	0.08371 - 0.09204
系统 28	0.08700	0.08332 - 0.09096
系统 31	0.08576	0.08186 - 0.08956
系统 2	0.08536	0.08148 - 0.08922
系统 33	0.08444	0.08057 - 0.08845
系统 5	0.08264	0.07911 - 0.08597

参与 DUC2006 会议的自动文摘系统一共有 35 个。会议对这 35 个系统生成的文摘进行了自动评价，采用的是 ROUGH 评价包。其评价时使用的参考标准是专家手工生成的文摘。采用 ROUGH-2 和 ROUGH-SU 对 DEPTM 算法生成的文摘进行评价。将评价结果与参与 DUC 会议的其他系统结果进行对比，表 3.2 和表 3.3 分别给出了 ROUGH-2 和 ROUGH-SU 的结果。

表 3.3 DEPTM 算法与其他系统的 ROUGH-SU 结果对比

	ROUGH-SU 均分	评分范围
DEPTM	0.15618	0.14233 - 0.16724
系统 24	0.15529	0.15126 - 0.15906
系统 12	0.14755	0.14360 - 0.15142
系统 15	0.14733	0.14373 - 0.15069
系统 8	0.14607	0.14252 - 0.14943
系统 28	0.14522	0.14157 - 0.14880
系统 23	0.14486	0.14096 - 0.14846
系统 33	0.14483	0.14118 - 0.14826
系统 31	0.14381	0.14004 - 0.14735
系统 2	0.14094	0.13732 - 0.14454
系统 5	0.14012	0.13682 - 0.14321

从实验结果对比中可以看出，DEPTM 算法是行之有效的。其平均表现好于其他系统。但是从评分范围上看，该算法的范围较大，说明该算法的稳定性略差。经过分析认为，这种现象是由狄利克雷分布的随机性造成的。如何减少这种随机性的影响，使 DEPTM 算法更加趋于稳定成熟，这是今后的一个重要的研究方向。

3.6 本章小结

本章的研究对象是句子抽取技术，讨论了概率主题模型下文摘句的抽取算法。分析了抽取技术的现状和抽取的目标。提出了基于差分进化的句子聚类方法，该方法是通过一个进化过程对句子进行聚类，根据句子的内容将句子划分到不同的主题聚类。又提出了基于概率主题模型的逐句递减规则，将句子的权值分为静态权值和动态权值

两部分，通过逐步去除无关句子的方法完成句子抽取。结合以上两者，提出了差分进化和概率主题模型相结合的句子抽取算法 DEPTM。最后，通过实验验证了本文算法的可行性。通过对比证明在 DUC2006 的 10 个数据集上，DEPTM 算法平均效果好于基于 NGD 的方法和 SumBasic 系统。

第4章 基于概率主题模型的文摘句排序算法

4.1 引言

上一章研究的，是句子的抽取方法，即选择哪些句子作为文摘句。要把选择出来的文摘句组成文摘，就涉及到文摘句的排序问题。

文摘句排列的顺序，对文摘的可读性有着非常重要的影响。Barzilay 等人做了如下实验[104]：使用 Columbia Summarization system[106]生成自动文摘，并提交至 Document Understanding Workshop（DUC）会议。然后从所有被人工评判为 poor 的文摘中选择 10 篇。以输入的源文档作为参考，对这 10 篇文摘由人工进行句子顺序调整。在进行顺序调整的过程中，保持文摘的内容不变，即调整后的文摘中所有的句子与原始生成的自动文摘中所有的句子完全一致，仅仅是排列顺序先后不同。通过这一过程，产生了 10 篇新的文摘。加上原始的 10 篇文摘，这样总共有 20 篇文摘。把这 20 篇文摘交给两位专家进行评测，这两位专家均未参与人工重排序，也没有阅读过源文档。将每篇文档的原始文摘和重排序文摘分别交给两位专家，专家阅读后给予评分。评分分为三档：无法理解、部分理解、完全理解。实验结果如表 4.1 所示。

表 4.1 句子顺序重要性的实验结果

id	原始文摘	重排序后的文摘
1	无法理解	无法理解
2	部分理解	完全理解
3	无法理解	完全理解
4	部分理解	完全理解
5	无法理解	部分理解
6	无法理解	无法理解
7	无法理解	无法理解
8	无法理解	完全理解
9	无法理解	部分理解
10	完全理解	完全理解

从表 4.1 可以看出，原始文摘中有 7 篇是无法理解的，有 2 篇部分理解，只有 1 篇是完全理解的。而经过调整之后的结果则有明显进步：3 篇无法理解，2 篇部分理解，5 篇完全理解。实验者还进一步进行了 Fisher 抽取，以证明上述实验的统计意义。

上述实验表明，句子顺序对于文章的理解是非常重要的。尽管句子的顺序并不存在唯一的最佳答案，但是顺序问题却是万万不能忽略的。对于给定的句子集合，并不是所有的顺序都可以让读者有效了解文章内容，只有少数顺序对读者来说是容易理解的。

对于单文档的抽取型文摘，文摘句的排序问题相对简单。因为所有的文摘句都来源于同一篇文档，源文档即可提供句子顺序的参考。源文档是人工编写的自然语言文档，其句子的顺序是按照人的思维和理解习惯排列的。因此有理由认为，如果把抽取出的文摘句按照其在原文中的先后顺序排列，将会给读者带来较好的阅读体验。尽管有学者研究证明原文顺序未必是句子的最佳顺序[105]，但是目前仍旧是最好的排序选择，也是单文档文摘主要采用的方法[2]。

多文档文摘则与单文档文摘不同，因为组成文摘的句子是没有一个唯一的原文顺序可供参考。因此，多文档文摘面临着更加复杂的排序问题。多文档抽取型文摘的句子排序方法，是目前自动文摘领域的一个研究热点。针对句子排序问题，人们提出许多算法。目前主要有三个方向：按时间顺序排列、按照原文顺序排列、按相似度排列。

按时间顺序排列的方法简称为时序方法（Chronological Ordering，CO）。由于自动文摘实验所采用的的数据，多数为新闻报道。每篇新闻报道都有其发布时间，通常能精确到分钟。把先发生的事件排在前面，后发生的事件排在后面，符合一般人的思维习惯。因此，把抽取出的文摘句按照其所在原文的发布时间的先后顺序来排列，可以取得较好的排序效果。时序方法对描述具体事件的文档集很适用，特别是对事件的持续追踪报道的新闻文档集。

按原文顺序的算法是将单文档文摘的句子排序方法直接应用于多文档文摘，比较典型的是 MO 算法（Majority Ordering）[104]。由于多文档文摘的句子来自于不同的文档，并不存在一个简单的唯一顺序。假设有两个句子，S1 和 S2。在文档 D1 中，可以找出与 S1 相似度最高的句子 S1’，与 S2 相似度最高的句子 S2’。若在 D1 中，S1’排列在 S2’之前，则认为 S1 应排列在 S2 之前。然而在不同的文档中，有可能得出完全相反的结果。在 MO 算法中是采用前置次序图的方法来解决这种冲突。MO 算法对原文档集的要求较高，如果所有的原文本都具有较规则的组织结构，表达方式、句子顺

序大体一致，那么 MO 算法能够得到较高的结果。反之，则效果很差。

前面两种方法都要依赖于原文档集，而基于相关性的方法则是不考虑原文，仅依赖抽取出的文摘句本身的方法。根据文摘句之间的相关性来决定文摘句的顺序。其基本思想是，如果文摘句的排序结果，可以使得任意两个相邻的句子之间存在比较强的相关性，那么这一顺序将产生较强的可读性。基于相关性的方法通常采用图算法来求解：把每个句子看做节点，句子之间的相关性作为节点间的距离，通过适当的算法找到一条包含所有点的路径，使得相邻点之间的距离之和最小。

本章的研究将结合这几种方法的优势，并根据概率主题模型的特点，提出基于概率主题模型的句子排序算法。

4.2 基于相关距离的文摘句排序算法

通过两种途径来衡量文摘句之间的相似程度。一种是不依赖于上下文环境，仅从两个文摘句本身得到的相似度，我们称之为本位相似度；另一种是向原文档集学习得到的，根据文摘句在原文档集的上下文语境环境而确定的相似度，我们称之为语境相似度。本节给出结合这两种相似度的基于相关距离的句子排序算法。

4.2.1 本位相似度和语境相似度

所谓本位相似度，就是完全从文摘句本身计算而得出的两个句子之间的相似程度。这里将本位相似度也分为两个部分。一部分是表层本位相似度，也就是根据两个句子是否包含相同单词来计算的相似度；另一部分是潜层相似度，则是计算两个句子之间的潜层语义间的联系。

定义 4-1 表层本位相似度

设句子 S_i 中包含 l_i 个单词，句子 S_j 中包含 l_j 个单词，S_i 和 S_j 中共同包含的单词有 l_{ij} 个。则句子 S_i 和 S_j 的表层本位相似度 $\mathrm{Sim}_{\text{inter-word}}$ 定义为：

$$Sim_{inter\text{-}word}(S_i, S_j) = \frac{l_{ij}}{\sqrt{l_i \cdot l_j}} \tag{4-1}$$

由表层本位相似度的定义可以看出，如果两个句子包含的共同的单词数越多，则其相似度越大。用两个句子的长度进行归一泛化，这样可以消除句子长度对相似度的

影响，并将表层本位相似度映射到[0,1]。一个句子与其本身的相似度最大，即 $Sim_{inter\text{-}word}(S_i,S_i)=1$。如果两个句子不包含任何相同的单词，则 $Sim_{inter\text{-}word}(S_i,S_i)=0$。

潜层相似度则要挖掘两个句子之间的语义关系，采用第二章提出的基于概率主题模型的相似度 LDASim 来度量。

定义 4-2 潜层本位相似度

文档集 C 的 k 维潜在主题空间为 $\mathbb{Z}_C^k$，对 C 中的任意两个句子 S_i 和 S_j，将其表示成 $\mathbb{Z}_C^k$ 中的潜在主题向量 $V_z(S_i)$和 $V_z(S_j)$。它们之间的基于概率主题模型的相似度称为潜层本位相似度，记作 $Sim_{inter\text{-}latent}$，即

$$Sim_{inter\text{-}latent}(S_i,S_j)=LDASim(S_i,S_j)=\cos(V_z(S_i),V_z(S_j)) \quad (4\text{-}2)$$

潜层本位相似度与表层本位相似度具有相同的性质，都属于[0,1]。一个句子与其自身的相似度最大，即 $Sim_{inter\text{-}latent}(S_i,S_i)=1$。如果两个句子不对应任何相同的潜在主题，则其相似度最小，即 $Sim_{inter\text{-}latent}(S_i,S_i)=0$。

通过对大量手工摘要的统计研究表明，如果两个文摘句中含有相同的实词，则这两个句子在文摘中排列在一起的概率大大增加[104]。如果文摘中两个相邻的句子包含有相同的单词，在阅读时会产生承接感，从而增强文摘的阅读体验。这说明表层本位相似度是衡量文摘中句子顺序关系的重要指标。同时，排列在一起的两个句子，应该在语义上有较强的关联性，这一点可以通过潜层本位相似度来衡量。因此，可以用表层本位相似度和潜层本位相似度的加权平均来定义本位相似度。

定义 4-3 本位相似度

文档集 C 中的任意两个句子 S_i 和 S_j，其表层本位相似度与潜层本位相似度的加权平均值称为本位相似度，记作 Sim_{inter}，即

$$Sim_{inter}(S_i,S_j)=\lambda\cdot Sim_{inter\text{-}word}(S_i,S_j)+(1-\lambda)\cdot Sim_{inter\text{-}latent}(S_i,S_j) \quad (4\text{-}3)$$

其中 $\lambda\in[0,1]$ 为权重平衡系数，调整表层本位相似度与潜层本位相似度的比重。显然本位相似度属于[0,1]。一个句子与其自身的本位相似度最大，即 $Sim_{inter}(S_i,S_i)=1$；如果两个句子不包含任何相同的单词，并且不对应任何相同的潜在主题，则其本位相似度最小，即 $Sim_{inter}(S_i,S_i)=0$。

之所以将本位相似度分成两个部分，就是希望从两个层面对句子的相似度有全面

的把握，避免采用一种方式的单一性。如果句子的含义类似，含有较多相同单词的句子会获得更高的相似度评分；如果句子包含的共同单词数相等，那么对应更多相同潜在主题的句子会得到较高的相似度评分。可见这里定义的本位相似度，从两个层面把握文摘句之间的相互关系，适合作为句子排序的标准。

由于要解决的任务是对文摘句的排列，而文摘句都是从原文抽取的，其排列顺序应该与原文息息相关。因此仅从文摘句本身来考察其相似度是不全面的。还应该根据原文的上下文语境来判断句子之间的相关程度。

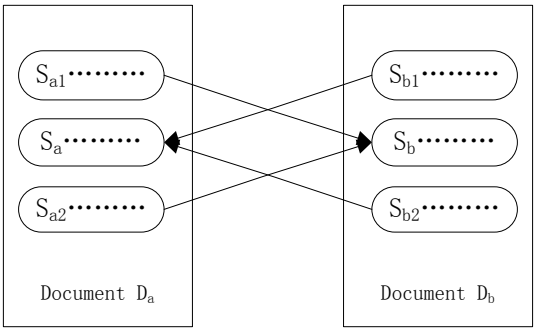

图 4.1 句子间的语境相似度

假设有 D_a 和 D_b 两篇文档。分别抽取文档 D_a 中的句子 S_a 和文档 D_b 中的句子 S_b，作为文摘句。现在要衡量 S_a 与 S_b 之间的相似度。假设在文档 D_a 中，S_a 的前一个句子是 S_{a1}，后一个句子是 S_{a2}；在文档 D_b 中，S_b 的前一个句子是 S_{b1}，后一个句子是 S_{b2}。可以通过衡量 S_a 与 S_{b1} 和 S_{b2} 的相关性，以及 S_b 与 S_{a1} 和 S_{a2} 的相关性，来考察 S_a 与 S_b 的相似度。

语境相似度的计算方法如图 4.1 所示。计算句子 S_{a1} 与句子 S_b 的相似度，如果 S_{a1} 与 S_b 非常相似，甚至完全相同，那么由于 S_{a1} 与 S_a 是同一原始文档中排列在一起的句子，二者必然具有较高的相似度，因此就有理由认为 S_b 与 S_a 有着极高的相似度。同样，计算 S_{a2} 与 S_b、S_a 与 S_{b1}、S_a 与 S_{b2} 的相似度，取这四者之中的最大值，就可以代表句子 S_a 与 S_b 在原文档集里面的相似度。

与本位相似度类似，语境相似度同样通过表层和潜层两个方面来衡量。

定义 4-4 表层语境相似度

对文档 D_a 中的句子 S_a 和文档 D_b 中的句子 S_b，设在文档 D_a 中，S_a 的前一个句子是 S_{a1}，后一个句子是 S_{a2}；在文档 D_b 中，S_b 的前一个句子是 S_{b1}，后一个句子是 S_{b2}。计算 S_{a1} 与 S_b、S_{a2} 与 S_b、S_a 与 S_{b1}、S_a 与 S_{b2} 的表层本位相似度，取这四者之中的最大值，称

为 S_a 与 S_b 表层语境相似度，记作 $Sim_{exter\text{-}word}(S_a, S_b)$，即

$$Sim_{exter\text{-}word}(S_a, S_b) = \max(Sim_{inter\text{-}word}(S_{a1}, S_b), Sim_{inter\text{-}word}(S_{a2}, S_b), Sim_{inter\text{-}word}(S_a, S_{b1}), Sim_{inter\text{-}word}(S_a, S_{b2})) \quad (4\text{-}4)$$

定义 4-5 潜层语境相似度

对文档 D_a 中的句子 S_a 和文档 D_b 中的句子 S_b，设在文档 D_a 中，S_a 的前一个句子是 S_{a1}，后一个句子是 S_{a2}；在文档 D_b 中，S_b 的前一个句子是 S_{b1}，后一个句子是 S_{b2}。计算 S_{a1} 与 S_b、S_{a2} 与 S_b、S_a 与 S_{b1}、S_a 与 S_{b2} 的潜层本位相似度，取这四者之中的最大值，称为 S_a 与 S_b 潜层语境相似度，记作 $Sim_{exter\text{-}latent}(S_a, S_b)$，即

$$Sim_{exter\text{-}latent}(S_a, S_b) = \max(Sim_{inter\text{-}latent}(S_{a1}, S_b), Sim_{inter\text{-}latent}(S_{a2}, S_b), Sim_{inter\text{-}latent}(S_a, S_{b1}), Sim_{inter\text{-}latent}(S_a, S_{b2})) \quad (4\text{-}5)$$

最后，与本位相似度类似，可以用表层语境相似度和潜层语境相似度的加权平均来定义语境相似度。

定义 4-6 语境相似度

文档集 C 中的任意两个句子 S_i 和 S_j，其表层语境相似度与潜层语境相似度的加权平均值称为语境相似度，记作 Sim_{exter}，即

$$Sim_{exter}(S_i, S_j) = \lambda \cdot Sim_{exter\text{-}word}(S_i, S_j) + (1-\lambda) \cdot Sim_{exter\text{-}latent}(S_i, S_j) \quad (4\text{-}6)$$

其中 $\lambda \in [0,1]$ 为权重平衡系数，调整表层语境相似度与潜层语境相似度的比重。根据语境相似度的定义，显然有语境相似度 $Sim_{exter}(S_i, S_j) \in [0,1]$。语境相似度的值取决于一个句子在原文中相邻的前后两个句子。所以一个句子与其本身的语境相似度并不等于 1，而是与其在原文中相邻的句子的语境相似度的值为 1。这说明语境相似度很好的刻画了文摘句在原文档集中的邻接特性，适合作为句子排序任务的判断准则。

4.2.2 基于相关距离的文摘句排序算法

首先来定义两个句子之间的相关距离

定义 4-7 相关距离

文档集 C 中的任意两个句子 S_i 和 S_j，其本位相似度与语境相似度的乘积称为这两个句子之间的相关距离，记作 $Dist(S_i,S_j)$，即

$$Dist(S_i,S_j)=Sim_{inter}(S_i,S_j)\cdot Sim_{exter}(S_i,S_j) \quad (4\text{-}7)$$

有了相关距离，就可以通过构造句子关系图来对句子进行排序。把每个句子看做节点，通过句子之间的相关距离作为节点与节点之间的弧的权重，由此构成一个句子关系图。这样，通过适当的图算法，找到一条包含所有点的路径，使得相邻点之间的距离之和最大。那么这条路径就是想要的排序结果。

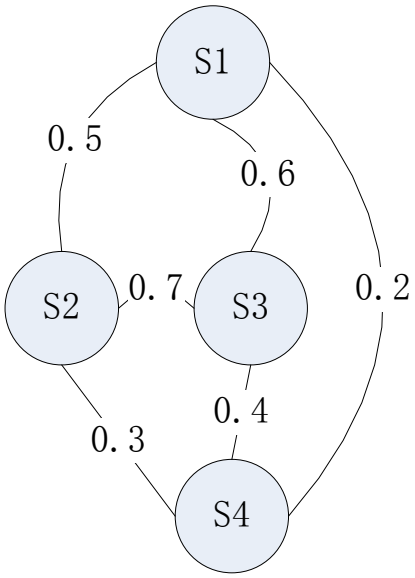

图 4.2 基于相关距离的句子关系图实例

图 4.2 给出了一个四个句子的排序实例，任意两个句子之间标出了它们的相关距离。在图 4.2 中，可以得到最佳顺序为 S1-S2-S3-S4 或 S4-S3-S2-S1。

虽然求解这样一个图是一个 NP 完全问题，不过很多数学家已经提出了求近似解的方法。即使要求得最佳解，从问题的规模上看也不是非常困难的事。因为文摘本身的一个属性就是远远短于原文。DUC 会议通常要求一篇文摘在 100 或 250 个单词以内。假设要进行排序的文摘包含 10 个句子，那么就有 10!=3628800 种顺序。这样的规模，进行完全搜索求解也是可以接受的。所以，对于相关性方法来说，重要的就是如何构造句子关系图，也就是计算句子之间的相关距离。

设原文档集为 C，从中抽取出 n 个句子组成摘要 T，即 $T=\{S_1,\cdots,S_n\}$。下面给出基

于相关距离的文摘句排序算法（Sentence Ordering algorithm based on Correlative Distance,SOCD）。

算法 4-1 基于相关距离的文摘句排序算法

步骤 1. 对 T 中的 n 个句子，根据公式（4-1），计算出任意两句子间的表层本位相似度。再找出 T 中句子在 C 中对应的原文档，根据公式（4-4）计算出任意两句子之间的表层语境相似度。

步骤 2. 对文档集 C 进行概率主题模型建模，用变分推理估计出参数 α 和 β。

步骤 3. 根据公式（4-2），计算出 T 中任意两个句子的潜层语境相似度。再按照 T 中句子在 C 中对应的原文档，根据公式（4-5）计算出 T 中任意两个句子的潜层语境相似度。

步骤 4. 根据公式（4-3）（4-6）和（4-7），计算出 T 中任意两个句子的相关距离。

步骤 5. 将句子视为节点，句子间的相关距离视为弧的权重，构造句子关系图。

步骤 6. 去掉权重小于阈值 δ 的弧。

步骤 7. 对句子关系图进行完全搜索，找到包含所有节点且使得相邻点之间的距离之和最大的路径。

算法中的 δ 是个人为规定的阈值，用于去掉权值过小的弧，这样可以简化句子关系图，提高图搜索的效率。

算法 4-1 最终求解的结果是图中的句子关系图最大路径，这条最大路径包含着正反两种句子排序结果。所以还需要用其他手段来选择其中的一个。一种方式是时序原则，看这条路径上两端的句子，哪一个所在的原文的时间较早，就以哪一个句子为开头。另一种方式是由文摘句在原文中排列顺序决定。找出每个文摘句在原文中的次序标号，用正反两种标号的排列结果，分别与序列{1..n}计算 spearman 相关系数，取相关系数高的排列结果作为最终的文摘句排列顺序。

4.3 基于原文顺序概率的文摘句排序算法

上一节算法中的语境相似度，是从原文档集中求解句子关系的。但是上一节所关心的只是句子间的相关性，因此对原文中句子位置的比较仅限于相邻的句子。事实上原文中所有的句子都可能提供有关句子相对位置的信息。本节将探讨如何从原文中学

习更多的位置信息并利用这些信息对文摘句进行排序。

4.3.1 前置概率和后置概率

两个句子的顺序，不外乎两种：S_a 在 S_b 之前、S_a 在 S_b 之后。所以要调整文摘句的顺序，需要向原文档集学习两种次序关系：前置概率和后置概率。

前置概率考察的是两个句子之间的位置关系，句子 S_a 是否应该排列在句子 S_b 之前，或者说句子 S_a 应该排列在句子 S_b 之前的概率。前置概率的基本思想是，如果句子 S_a 与 D_b 中排列在句子 S_b 之前的某一句子非常相似，那么就认为句子 S_a 排列在句子 S_b 之前是有一定合理性的。其相似程度越大，则如此排列的概率越大。图 4.3 给出了前置概率度量的基本思想。

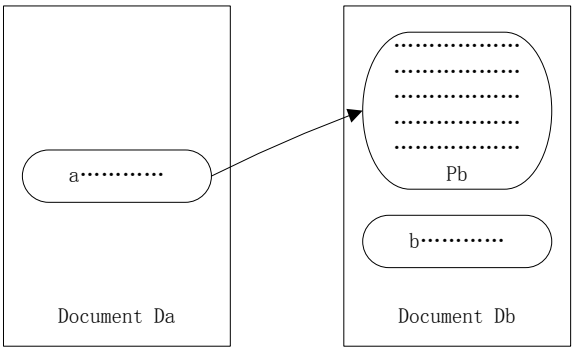

图 4.3 前置概率度量的基本思想

同上一节中的相似度一样，定义前置概率也分为表层和潜层两部分。

定义 4-8 表层前置概率

对文档集 C 中的两篇文档 D_a 和 D_b，句子 $S_a \in D_a$，句子 $S_b \in D_b$。设 P_b 为文档 D_b 中排列在句子 S_b 之前的所有句子的集合。则将句子 S_a 与 P_b 中所有句子的表层本位相似度中的最大值称为 Sa 与 Sb 的表层前置概率，记作 $p_{\text{prev-word}}$，即

$$p_{prev\text{-}word}(S_a, S_b) = \max_{b' \in P_b} Sim_{inter\text{-}word}(S_a, b') \tag{4-8}$$

定义 4-9 潜层前置概率

对文档集 C 中的两篇文档 D_a 和 D_b，句子 $S_a \in D_a$，句子 $S_b \in D_b$。设 P_b 为文档 D_b 中排列在句子 S_b 之前的所有句子的集合。则将句子 S_a 与 P_b 中所有句子的潜层本位相似度中的最大值称为 Sa 与 Sb 的潜层前置概率，记作 $p_{\text{prev-latent}}$，即

$$p_{prev\text{-}latent}(S_a, S_b) = \max_{b' \in P_b} Sim_{inter\text{-}latent}(S_a, b') \quad (4\text{-}9)$$

则由表层前置概率和潜层前置概率，可以得出前置概率的定义。

定义 4-10 前置概率

对文档集 C 中的两个句子 S_a 和 S_b，将 S_a 和 S_b 的表层前置概率与潜层前置概率的平方平均值成为 S_a 与 S_b 的前置概率，记作 p_{prev}，即

$$p_{prev}(S_a, S_b) = \sqrt{\frac{p^2_{prev-word}(S_a, S_b) + p^2_{prev-latent}(S_a, S_b)}{2}} \quad (4\text{-}10)$$

后置概率考察的是两个句子之间的位置关系，句子 S_b 是否应该排列在句子 S_a 之后，或者说句子 S_b 应该排列在句子 S_a 之后的概率。后置概率的基本思想是，如果句子 S_b 与 D_a 中排列在句子 S_a 之后的某一句子非常相似，那么就认为句子 S_b 排列在句子 S_a 之后是有一定合理性的。其相似程度越大，则如此排列的概率越大。图 4.4 给出了后置概率度量的基本思想。

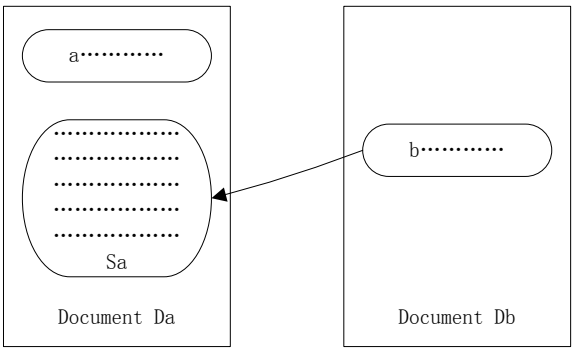

图 4.4 后置概率度量的基本思想

同样，后前置概率也分为表层和潜层两部分。

定义 4-8 表层后置概率

对文档集 C 中的两篇文档 D_a 和 D_b，句子 $S_a \in D_a$，句子 $S_b \in D_b$。设 N_a 为文档 D_a 中排列在句子 S_a 之后的所有句子的集合。则将句子 S_b 与 N_a 中所有句子的表层本位相似度中的最大值称为 S_a 与 S_b 的表层后置概率，记作 $p_{next\text{-}word}$，即

$$p_{\text{next-}word}(S_a, S_b) = \max_{a' \in N_a} Sim_{inter\text{-}word}(a', S_b) \quad (4\text{-}11)$$

定义 4-9 潜层后置概率

对文档集 C 中的两篇文档 D_a 和 D_b，句子 $S_a \in D_a$，句子 $S_b \in D_b$。设 N_a 为文档 D_a 中排列在句子 S_a 之后的所有句子的集合。则将句子 S_b 与 N_a 中所有句子的潜层本位相似度中的最大值称为 S_a 与 S_b 的潜层后置概率，记作 $p_{\text{next-latent}}$，即

$$p_{\text{next-}latent}(S_a, S_b) = \max_{a' \in N_a} Sim_{inter\text{-}latent}(a', S_b) \tag{4-12}$$

则由表层后置概率和潜层后置概率，可以得出后置概率的定义。

定义 4-10 后置概率

对文档集 C 中的两个句子 S_a 和 S_b，将 S_a 和 S_b 的表层后置概率与潜层后置概率的平方平均值成为 S_a 与 S_b 的后置概率，记作 p_{next}，即

$$p_{next}(S_a, S_b) = \sqrt{\frac{p^2_{next-word}(S_a, S_b) + p^2_{next-latent}(S_a, S_b)}{2}} \tag{4-13}$$

S_a 和 S_b 的前置概率描述的是句子 S_a 排列在句子 S_b 之前的概率；S_a 和 S_b 的后置概率描述的是句子 S_b 排列在句子 S_a 之后的概率。前置概率和后置概率描述的其实是同一种顺序关系。由此，可以将前置概率和后置概率综合而成原文顺序概率。

定义 4-11 原文顺序概率

对文档集 C 中两个句子 S_a 和 S_b，将 S_a 和 S_b 的前置概率与后置概率的乘积称为 S_a 排列在 S_b 之前的原文顺序概率，记为 $p(S_a > S_b)$，即

$$p_{order}(S_a > S_b) = p_{prev}(S_a, S_b) \cdot p_{next}(S_a, S_b) \tag{4-14}$$

这里使用符号“>”，并没有比较大小的含义，仅仅标识两个句子的顺序方向。

两个句子的原文顺序概率，就是从原文档集中学习到的这两个句子如此排列的可能性。利用所有文摘句的原文顺序概率信息，就可以找到一种较为合理的文摘句的排列顺序。

4.3.2 基于原文顺序概率的文摘句排序算法

有了前文定义的原文顺序概率，就可以生成一个句子关系的有向图，通过这一有向图来求解句子顺序。

将抽取出的文摘句作为节点，两个节点 S_a 和 S_b 之间存在两条有向弧：S_a 到 S_b 的弧、S_b 到 S_a 的弧。分别将原文顺序概率 $p_{order}(S_a>S_b)$ 和 $p_{order}(S_b>S_a)$ 作为这两条弧的权重，就得到了一个句子关系的有向图。设置阈值 δ，若一个弧的权重值小于 δ，则从图中去掉该弧，以简化计算。有了这样一个有向图，句子排列问题就转化为：在句子关系有向图中求一条路径，经过每个节点恰好一次，且具有最高的权重和。

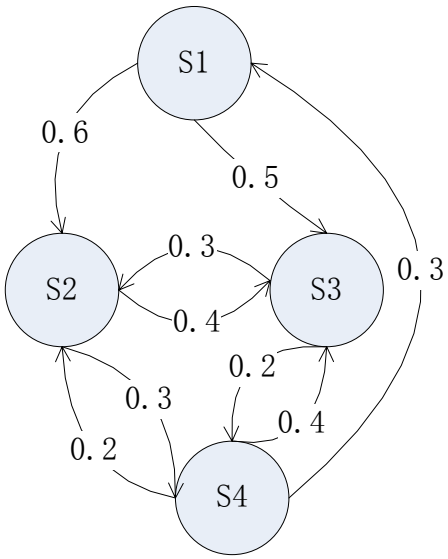

图 4.5 基于原文顺序概率的句子关系有向图实例

图 4.5 给出了句子关系有向图的一个例子。从图中可以看出，句子的最佳排列顺序应该为 S1-S2-S4-S3。

不幸的是，此问题仍旧是一个 NP 完全问题[108]。在节点数不多的情况下，可以采用完全搜索的办法来求解。也可以采用如下算法得到近似最优解：（1）给每个节点赋一个权重值，其值等于该节点的出弧的权重和减去入弧的权重和。（2）找出权重最大的节点，作为起始节点。（3）将起始节点从句子关系有向图中删除，之后更新每条弧的权重。（4）重复上述步骤，直到句子关系有向图中不包含任何节点为止。这样，删除节点的顺序即为最后的排序结果。采用这种算法得到不一定是有向图的最优解，但是效率比完全搜索要高得多。

设原文档集为 C，从中抽取出 n 个句子组成摘要 T，即 $T=\{S_1,\cdots,S_n\}$。下面给出基

于原文顺序概率的文摘句排序算法（Sentence Ordering algorithm based on Source Ordering Probability，SOSOP）。

算法 4-2 基于原文顺序概率的文摘句排序算法

步骤 1. 对 T 中的 n 个句子，找出 T 中句子在 C 中对应的原文档。根据公式(4-8)，计算出任意两句子间的表层前置概率，根据公式（4-11）计算出任意两句子之间的表层后置概率。

步骤 2. 对文档集 C 进行概率主题模型建模，用变分推理估计出参数 α 和 β。

步骤 3. 按照 T 中句子在 C 中对应的原文档，根据公式（4-9），计算出 T 中任意两个句子的潜层前置概率，根据公式（4-12）计算出 T 中任意两个句子的潜层后置概率。

步骤 4. 根据公式（4-10）（4-13）和（4-14），计算出 T 中任意两个句子的原文顺序概率。

步骤 5. 将句子视为节点，两个句子的原文顺序概率视为有向弧的权重，构造句子关系有向图。

步骤 6. 去掉权重小于阈值 δ 的弧。

步骤 7. 给每个节点赋一个权重值，其值等于该节点的出弧的权重和减去入弧的权重和。

步骤 8. 找出权重最大的节点，作为起始节点。

步骤 9. 将起始节点从句子关系有向图中删除，之后更新每条弧的权重。

步骤 10. 重复步骤 9，直到句子关系有向图中不包含任何节点为止。这样，删除节点的顺序即为最后的排序结果。

算法 4-2 与算法 4-1 最大的不同在于，求解的图是有向图。4.2 节中计算的是两个句子之间的相似程度，没有前后次序关系，具有可交换性。而本节中计算的是两个句子的次序关系，不可交换。事实上，相似性和次序关系都是句子排列中需要参考的标准。如果能将这两种方式结合起来，必将提升句子排序的效果。

4.4 基于概率主题模型的层次性文摘句排序算法

前面两节中分别介绍了两种不同的文摘句排序算法。一种是基于相关距离的，一

种是基于顺序关系的。本节将提出一种新的算法，可以将相关距离与顺序关系结合起来，提升句子排序的效果。

4.4.1 句子链和句子链的邻接

首先定义邻接符号和句子链。

定义 4-12 邻接符号“→”

对于两个句子 S_a 和 S_b，$S_a \rightarrow S_b$ 表示 S_a 与 S_b 相邻接。这里的邻接，有两层含义。一是表示这两个句子的排列是连在一起的，中间没有其他句子；二是表示句子 S_a 排序在句子 S_b 之前。

定义 4-13 句子链

一个句子链由一连串的相邻接的句子组成，即 $A=(a_1 \rightarrow a_2 \rightarrow \cdots \rightarrow a_{n-1} \rightarrow a_n)$ 是长度为 n 的句子链。

箭头符号也可用于句子链，表示两句子链相邻接。设有两个句子链 $A=(a_1 \rightarrow a_2 \rightarrow \cdots \rightarrow a_{n-1} \rightarrow a_n)$ 和 $B=(b_1 \rightarrow b_2 \rightarrow \cdots \rightarrow b_{m-1} \rightarrow b_m)$，则 $A \rightarrow B=(a_1 \rightarrow a_2 \rightarrow \cdots \rightarrow a_{n-1} \rightarrow a_n \rightarrow b_1 \rightarrow b_2 \rightarrow \cdots \rightarrow b_{m-1} \rightarrow b_m)$。两个句子链相邻接的结果仍旧是一个句子链。

4.4.2 四种顺序标准

可以使用四种标准来衡量句子链之间的关系：时序标准、相关距离、前置概率、后置概率。

时序标准就是把句子按照其时间先后来排序。在多文档自动文摘任务中，可以把从不同的文档中抽取出的句子，按照该句子所在原文的时间先后来排列。这一标准在很多系统中得到过比较广泛的应用。

对于单个句子，定义时序标准如下：

$$f_{chro}(a \rightarrow b)=\begin{cases}1 & T(a)<T(b) \\ 1 & D(a)=D(b) \text{且} N(a)<N(b) \\ 0.5 & T(a)=T(b) \text{且} D(a)\neq D(b) \\ 0 & otherwise\end{cases} \quad (4\text{-}15)$$

其中 T(a)表示句子 a 所在文档的发布时间。若句子 a 所在文档的发布时间比句子 b 所在文档的发布时间早，那么 a 应当排列在 b 之前。D(a)表示句子 a 所在的文档，N(a)表示句子 a 在原文中的位置。如果句子 a 和 b 在同一个文档中，且在原文中 a 排列在 b 之前，那么在结果文摘中 a 应当排列在 b 之前，这是单文档文摘中句子的一般排序标准。如果句子 a 和 b 属于不同的文档，但这两篇文档的时间是相同的，那么 a 和 b 之间存在有一定的时序关系，取 f_{chro}=0.5。其他情况则无法得知两者之间的时序关系，取值为 0。

将时序标准扩展到句子链 A 和 B，可得如下公式：

$$f_{chro}(A \to B) = \begin{cases} 1 & T(a_m) < T(b_1) \\ 1 & D(a_m) = D(b_1) \text{且} N(a_m) < N(b_1) \\ 0.5 & T(a_m) = T(b_1) \text{且} D(a_m) \neq D(b_1) \\ 0 & otherwise \end{cases} \tag{4-16}$$

其中 a_m表示句子链 A 中的最后一个句子，b_1表示句子链 B 中的第一个句子。当 A 中的最后一个句子和 B 中的第一个句子存在明显的时序关系时，就认为句子链 A 和 B 存在时序关系，否则取值为 0。

在比较句子链的时序关系时，可以进行更复杂的运算（比如依次比较任意句子对之间的时序关系）。但是通过大量的实验，发现更复杂的运算方式并没有给排序结果带来明显地提升，简单的首尾句比较已经足以反映句子链之间的时序关系。因此保留了上述简单形式的公式。

定义 4-7 已经给出了两个句子之间的相关距离。现在把句子的相关距离扩展到句子链。对于两个句子链 A 和 B，它们之间的相关距离定义如下：

$$f_{dist}(A \to B) = \frac{1}{|B|} \sum_{b \in B} \max_{a \in A} Dist(a,b) \tag{4-17}$$

对于每一个句子 b∈B，计算 A 中所有句子与 b 的相关距离，取其最大值作为 b 与 A 的相关距离。再将 B 中每一个句子与 A 的相关距离求均值，得到 A 与 B 的相关距离。

定义 4-10 已经给出了两个句子之间的前置概率。现在把句子的前置扩展到句子链。对于两个句子链 A 和 B，它们之间的前置概率定义如下：

$$f_{prev}(A \to B) = \frac{1}{|B|} \sum_{b \in B} \max_{a \in A} p_{prev}(a,b) \tag{4-18}$$

对于句子链 B 中的任一个句子 b，计算句子链 A 中所有句子与它的前置概率，找出其中的最大值作为句子链 A 与 b 的前置概率。再把所有的 A 与 B 中句子的前置概率的平均值作为这两个句子链的前置概率值。

定义 4-13 已经给出了两个句子之间的后置概率。现在把句子的后置扩展到句子链。对于两个句子链 A 和 B，它们之间的后置概率定义如下：

$$f_{next}(A \to B) = \frac{1}{|A|}\sum_{a \in A} \max_{b \in B} p_{next}(a,b) \tag{4-19}$$

对于句子链 A 中的任一个句子 a，计算句子链 B 中所有句子与它的后置概率，找出其中的最大值作为 a 与句子链 B 的后置概率。把所有的 A 中的句子与 B 的后置概率的平均值作为这两个句子链的后置概率值。

以上将时序标准、相关距离、前置概率、后置概率这四种标准从句子扩展到了句子链上，从而为后面的排序算法打下基础。

4.4.3 四种标准的结合

上一节给出了四种度量标准。其中相关距离是从两个句子的意义角度出发，考察两个句子是否应该相邻；前置概率和后置概率，是从句子排列顺序的角度出发，考察一个句子是否应该排列在另一个句子之前或之后；时序标准是从时间角度出发，考察两个句子的时间顺序。这四个标准分别从不同的角度刻画了两个句子的相邻关系。但是不知道这四个标准在衡量两个句子的相邻关系时占有怎样的比重。因此目标是用这四个标准，构建出一个函数，来衡量两个句子链的相邻关系，即：

$$f(A \to B) = \begin{cases} p & \text{如果A和B邻接} \\ 0 & \text{如果A和B不邻接} \end{cases} \tag{4-20}$$

其中 0≤p≤1，表示 A 与 B 相邻的可能性。

f(A→B)的定义可以看成一个两类的分类器，即将两个句子的关系分成邻接和不邻接两类，因此可以使用支持向量机（SVM）来完成这项工作。

通过 SVM，可以找出一个超平面，将空间中的数据点划分为两个部分。将两个句子链组成的“句子链对”作为空间中的数据点，通过超平面把这些数据点划分为“邻接的句子链对”和“不邻接的句子链对”。训练 SVM 需要正例和反例，可以人工来生成这些数据。取文档集 C，人工从文档集中抽取句子组成一篇文摘 S。假设 S 中含有 m

个句子，即 $S=(a_1\rightarrow a_2\rightarrow\cdots\rightarrow a_m)$。则可以依次抽取 S 中相邻的两个句子链 A 和 B 组成句子链对。具体抽取过程为：

(1.1) 抽取的句子链 A 和 B 中，均含有 1 个句子。可以得到 m-1 个句子链对，即 $\{(a_1),(a_2)\},\{(a_2),(a_3)\},\cdots\{(a_{m-1}),(a_m)\}$。

(1.2) A 中含有 1 个句子，B 中含有 2 个句子。即 $\{(a_1),(a_2\rightarrow a_3)\}$, $\{(a_2),(a_3\rightarrow a_4)\}$, … ,$\{(a_{m-2}),(a_{m-1}\rightarrow a_m)\}$。

……

(m-1.1) A 中含有 m-1 个句子，B 中含有 1 个句子。即$\{(a_1\rightarrow a_2\rightarrow\cdots\rightarrow a_{m-1}),(a_m)\}$。

这样，最终可以得到$(m^3-m)/6$ 个句子链对。将所有的句子链对按正序邻接，即(A→B)。求出前面的四种标准组成向量，即(f_{dist}(A→B), f_{chro}(A→B), f_{prev}(A→B), f_{next}(A→B))，作为训练 SVM 的正例。将所有的句子链对按反向邻接，即(B→A)，计算出向量(f_{dist}(B→A), f_{chro}(B→A), f_{prev}(B→A), f_{next}(B→A))作为反例。有了正例集和反例集，就可以训练出合适的 SVM。

4.4.4 基于概率主题模型的层次性文摘句排序算法

使用训练好的 SVM，就可以判断任意两个句子链是否邻接。如果 A 和 B 不邻接，则令 f(A→B)=0；如果 A 和 B 邻接，则可以计算出向量(f_{dist}(A→B), f_{chro}(A→B), f_{prev}(A→B), f_{next}(A→B))到超平面的距离，将此距离归一泛化到(0,1)，就可以得到 A 与 B 邻接的可能性。这样，通过前面的五种度量标准，找到了公式（4-20）的具体计算方法。

上面得到了两个句子链邻接函数的计算方法，下面来讨论具体的排序算法。给定句子集合 $T=\{a_1,a_2,\cdots,a_n\}$，首先将此集合看做是 n 个句子链的集合，每个句子链只包含一个句子，即 $T=\{A_1,A_2,\cdots,A_n\}=\{(a_1),(a_2),\cdots,(a_n)\}$。利用前面的邻接函数，找出 T 中相邻程度最高的两个句子链。将这两个句子链相连接成为一个句子链，这样 T 就变成了 n-1 个句子链的集合。重复前面的过程，直到 T 中只包含一个句子链，那么这个句子链就是想要的句子最终的排序结果。

设原文档集为 C，从中抽取出 n 个句子组成摘要 T，即 $T=\{S_1,\cdots,S_n\}$。下面给出基于概率主题模型的层次性句子排序算法（Bottom-up Sentence Ordering algorithm based on Probabilistic topic model，BSOP)。

算法 4-3 基于概率主题模型的层次性文摘句排序算法

步骤 1. 采用 4.4.3 节中的方法，利用四种标准训练支持向量机。

步骤 2. 将文摘 T 表示成 n 个句子链的集合，即 $T=\{(a_1),(a_2),\cdots,(a_n)\}$。

步骤 3. 利用训练好的支持向量机，根据公式（4-20），计算 T 中所有句子链之间邻接关系。

步骤 4. 找出邻接程度最高的两个句子链，把这两个句子链连接成一个句子链。

步骤 5. 重复步骤 3 到步骤 4，直到 T 中仅包含一个句子链。

4.5 实验及结果分析

选用了广为流行的 DUC 2006 的数据集作为实验数据集。DUC2006 数据集包括了 50 个不同主题的文档集，每个文档集包含 25 篇新闻。选择其中的 30 个文档集作为训练集，用剩下的 20 个文档集作为测试集。

包括本文方法在内，一共有 8 种方法参与实验：

SOCD，本文提出的基于相关距离的文摘句排序算法

SOSOP，本文提出的基于原文顺序概率的文摘句排序算法

BSOP，基于概率主题模型的层次性文摘句排序算法

AGL，Bollegala 等于 2010 年提出的文摘句排序算法[110]

MO，依据原文顺序的 Majority Ordering 算法[104]

CO，仅依赖时间标准对文摘句进行排序。

RND，对文摘句进行随机排序。

HUM，人类手工排序。

通过人工评价和自动评价两种方式来验证排序算法的效果。所谓人工评价，就是由专家直接针对排序结果进行评分；所谓自动评价，就是由专家给出人工排序结果，用人工排序结果作为参考，计算机器排序结果与人工排序结果之间的相关性，从而得出机器排序结果的好坏。

首先进行人工评价。找五位专家来对每一种排序打分，分值分为四个等级：优、良、可、差。其中“优”表示按此顺序生成的文摘完全可以正常阅读并理解，不再需要调整句子顺序；“良”表示基本可以正常理解文摘内容，但句子顺序仍可以进一步

调优；“可”表示勉强可以理解文摘内容，如果重新调整句子顺序，将会给文摘阅读带来很大改善；“差”表示无法理解文摘内容，完全需要重新组织句子顺序。

表 4.2 八种排序算法的人工评分测试结果

	优	良	可	差
BSOP	28	44	20	8
SOCD	21	31	28	20
SOSOP	22	35	24	19
AGL	24	39	21	16
MO	9	23	6	62
CO	21	25	32	22
RND	0	5	8	87
HUM	82	15	2	1

对于每一个方法，有 20 个测试集，5 位专家进行评分，因此可以得到了 20*5=100 个评分结果。表 4.2 给出了结果数据。

图 4.6 给出了上述数据的条形图，按照效果好坏的顺序排列，更直观的反映各种方法的优劣。

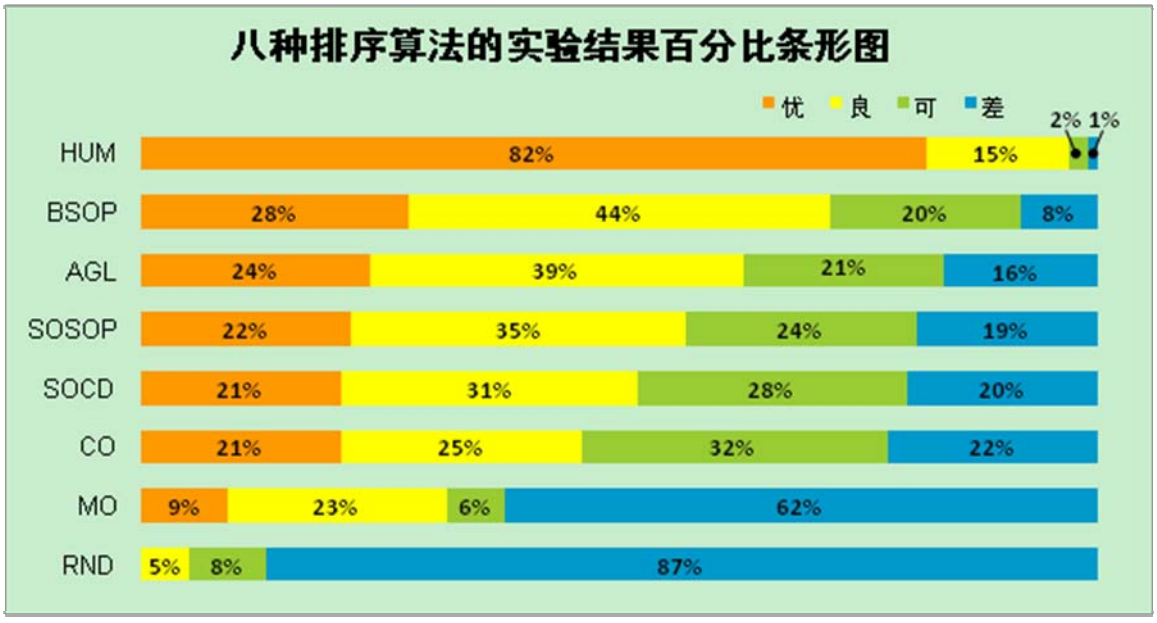

图 4.6 八种排序算法的实验结果百分比条形图

以上是人工评价的结果。下面进行自动评价。采用两种相关系数来进行计算：Kendall’s τ和Spearman秩相关系数，来衡量两个排序之间的关系。

设π和σ是N个元素的两种排序方法。如果两个元素x和y，在π和σ中排列的先后顺序一样，则称（x,y）为顺序相同对；反之，则称为顺序不同对。π和σ的Kendall’s τ定义为：

$$\tau = \frac{\text{（顺序相同对的数量）-（顺序不同对的数量）}}{\text{总对数}} \tag{4-21}$$

Kendall’s τ 的取值范围是[-1,1]。当两个顺序完全相同时，τ=1；当两个顺序完全相反时，τ=-1。

对于N个元素的任意两个排序π和σ，它们之间的 Spearman 秩相关系数定义如下：

$$r_s = 1 - \frac{6}{N(N+1)(N-1)} \sum_{i=1}^{N} (\pi(i) - \sigma(i))^2 \tag{4-22}$$

其中$\pi(i)$和$\sigma(i)$分别代表第i个元素在π和σ中的排列次序。Spearman 秩相关系数r_s的取值范围是[-1,1]。当两个顺序完全相同时，r_s=1；当两个顺序完全相反时，r_s=-1。

将前面的七种排列方法BSOP、SOCD、SOSOP、AGL、MO、CO、RND所得到的结果与人工排序HUM得到的结果进行对比，计算上述两种系数，得到结果如表4.3所示。

表 4.3 七种排序方法与人工方法对比的结果

	Kendall	Spearman
RND	-0.064	-0.124
MO	0.311	0.338
CO	0.407	0.423
SOCD	0.446	0.471
SOSOP	0.587	0.591
AGL	0.613	0.604
BSOP	0.622	0.617

从实验结果可以看出，本文提出的 BSOP 算法的结果要好于 AGL 方法。SOSOP 和

SOCD 算法的效果不及 AGL，但是好于 MO 算法，略好于 CO 算法。BSOP 是综合了多种评价标准的算法，可见本文从本位语境、表层潜层、前置后置等多个角度对句子关系的刻画是卓有成效的，对提成文摘句排序的效果

4.6 本章小结

本节研究的对象是文摘句的排序算法。分析了文摘句排序的重要性以及目前主要的排序算法。从潜层表层、本位语境等不同的角度定义了文摘句之间的相似度，提出了基于相关距离的文摘句排序算法；从原文档集中学习句子的排序规律，定义了前置概率和后置概率，并提出了基于原文顺序概率的文摘句排序算法；定义了句子链及其邻接关系，以时序标准、相关距离、前置概率、后置概率这四项标准为基础，通过训练支持向量机将四项标准结合在一起，提出了基于概率主题模型的层次性的文摘句排序算法。通过实验对比证明，文本提出的算法能够产生较好的排序效果，比现有的部分方法有所提高，但是比人工排序的理想效果还有比较大的差距。

第5章 自动文本摘要的评价方法研究

5.1 引言

自动文摘的评价问题，从自动文摘技术诞生以来，就一直是一个难以解决的、有争议的话题[38]。图 5.1 给出了目前主要文摘评价方法的分类树。

最早期的自动文摘采用人工评价的方式，即直接由专家来判定机器生成文摘的好坏。这种方法费时费力，代价昂贵，且标准不统一，人们渴望一种自动评价的方法。

自动文摘的自动评价方法分为外部评价和内部评价。外部评价是指将生成的文摘代替原文，应用于某项实际的任务，例如文本分类任务、文本检索任务等等。将使用文摘执行任务的结果与使用原文执行任务的结果进行对比，从而对文摘的质量进行评价。内部评价方法则是将自动生成的文摘与人工生成的文摘进行对比，根据二者的相似程度对自动文摘进行评价。内部评价方法也是当前研究的热点领域。

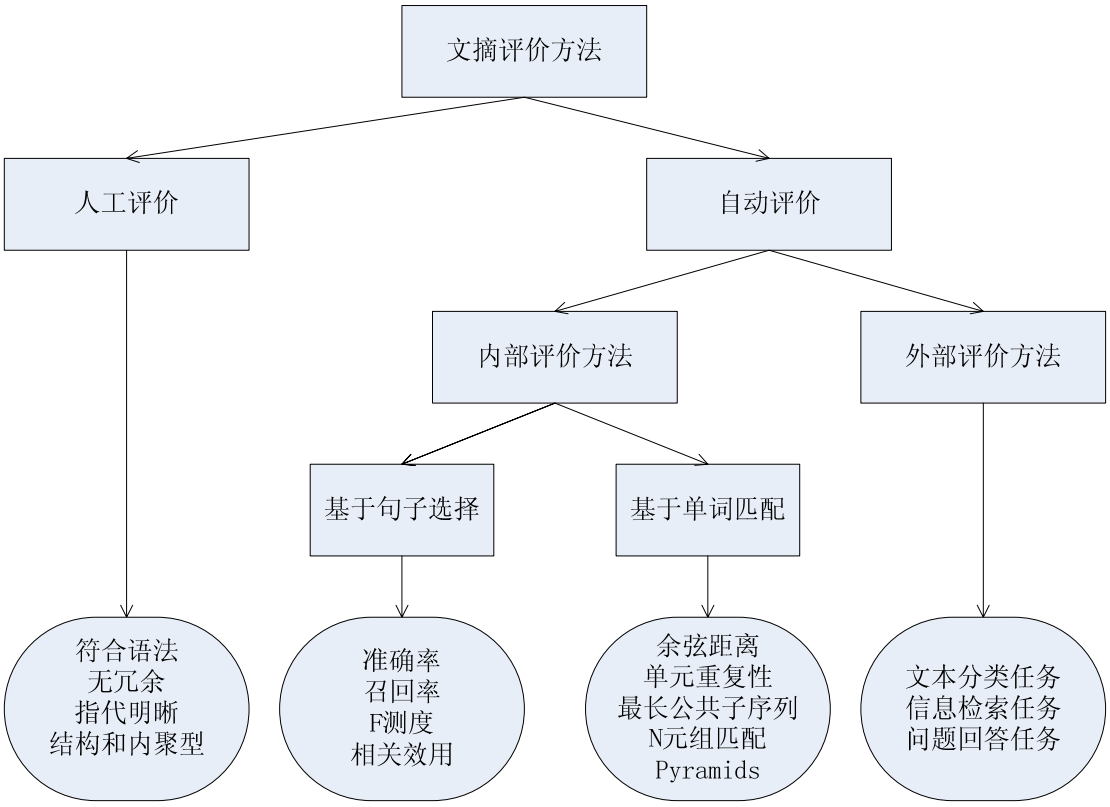

图 5.1 文摘评价方法分类树

最初的人工评价，是由一位或多位专家，对生成文摘的总体质量进行评估打分。最简单的评分是好或不好，可接受或不可接受，也可以从事先制定的一个评分等级中选择，比如 A 到 E 共分 5 级，A 表示非常好，E 表示非常差。

这样对文摘整体质量作出评估，显然过于粗糙。每个评分专家的习惯不同、侧重点不同，给出的评分往往差异很大。这样的评价方法，不但无法推广和比较，就连对一篇文摘的一次评价也不够稳定和令人信服。因此，人们对手工的文摘评价也做了较为细致的划分，从不同的侧面对文摘进行评估。主要有：

（1）符合语法（grammaticality）。文摘首先必须能够被人们阅读和理解，不能存在明显的语法错误，不能有非法字符、标点错误、单词拼写错误等等。

（2）无冗余（non-redundancy）。文摘本身是个精简的压缩的文档，其中不应该含有冗余的信息。如果含有大量的冗余信息，一定不是好文摘。

（3）指代明晰（reference clarity）。文摘中出现的代词，要有明确的指向。任何一个代词，一定要可以从文摘的上下文中找出其指代对象。

（4）意思连贯结构良好（coherence and structure）。文摘应该有良好的结构，叙述清晰，表意连贯，前后一致，行文通常。

一般的人工评价系统中，都是对以上四个侧面分别进行打分。例如在 DUC2005 会议中，每篇文摘的每一个侧面都按照从 A 到 E 的 5 个等级来进行评分。

内部评价方法就是按照根据文摘自身的质量来评价，而不借助于外部任务。通常内部评价方法都要有参考标准，把要评价的文摘和参考的标准文摘进行比较，根据其相似程度对文摘进行评价。限于文摘自身的特点，绝对理想的参考标准并不存在，一般人们采用一篇或多篇手工文摘作为参考的标准。把要评价的文摘称为“候选文摘（candidate summary）”，把作为参考标准的手工文摘称为“参考文摘（reference summary）”或“理想文摘（ideal summary）”。把候选文摘和参考文摘进行对比时，可以采用不同的粒度。根据所采用的粒度不同，可以简单的划分为基于句子的方法和基于词的方法。

基于句子的评价方法中，最常见的就是准确率（precision）、召回率（recall）、F 测度（F-measure）[117][118]。这三个指标，在很多领域中都有着广泛应用，与文本处理相关的文本分类、文本检索等方向中，都可以使用这三个基本的指标作为衡量标准。

在基于句子的自动文摘评价中，准确率是指，候选文摘和参考文摘中都包含的句子个数，与候选文摘中句子个数的商，即：

$$Precision = \frac{Count(Cs \cap Rs)}{Count(Cs)} \tag{5-1}$$

其中 Cs 表示候选文摘，Rs 表示参考文摘。Count 为统计句子个数的函数。

召回率是指，候选文摘的参考文摘都包含的句子个数，与参考文摘中句子个数的商，即：

$$Recall = \frac{Count(Cs \cap Rs)}{Count(Rs)} \tag{5-2}$$

F 测度则是准确率与召回率的一种组合度量，基本的 F 测度为准确率与召回率的调和平均[119]，即：

$$F = \frac{2 \cdot P \cdot R}{P + R} \tag{5-3}$$

而 F 测度更加复杂的形式可以表示为：

$$F = \frac{(\beta^2 + 1) \cdot P \cdot R}{\beta^2 \cdot P + R} \tag{5-4}$$

其中 β 是一个调节因子，β 的大小决定了 F 测度的偏向。当 β=1 时，就是 F 测度的基本形式，准确率和召回率同等考虑。当 β>1 时，F 测度的结果更偏向于准确率；当 β<1 时，则更偏向于召回率。

这种准确率和召回率的方法，存在的一个主要问题就是，人们对同一篇文档当中句子的重要性的判断并不一致。使用准确率和召回率，往往会对两个质量相差无几的文摘产生非常不同的评价结果。例如，做为标准的手工文摘中包含两个句子{S_1,S_2}，有两篇候选文摘 A 和 B，其中 A 包含两个句子{S_1,S_2}，B 包含两个句子{S_1,S_3}。使用准确率和召回率的方法，文摘 A 显然会比文摘 B 得到的分数高许多。然而，事实的情况是很有可能句子 S_2 和句子 S_3 在原文中是同等重要的，两篇文摘本该得到差不多的分数。

为了处理上面提到的问题，引入相对效用（relative utility，RU）[44]度量。通过 RU，可以把输入文档中的句子表示成置信值的形式，置信值根据句子是否包含在文摘中来计算。例如，一个文档包含 5 个句子，{S_1,S_2,S_3,S_4,S_5}，则可以表示成{S_1/5, S_2/4, S_3/4, S_4/1, S_5/2}。这里每个句子对应的数字表示，根据人工的判断，该句子可以成为

摘要的可能性。这个值称为句子的效用，它依赖于输入文档、摘要长度以及人工的判断。在上面这个例子中，文摘中出现{S_1, S_2}并不会比出现{S_1, S_3}获得更好的分数，因为这两种选择会得到同样的效用分数（5+4）。而且上面的例子中没有比这得分更高的两个句子的组合了，所以{S_1, S_2}和{S_1, S_3}都是最佳的文摘。

要计算相对效用，对一篇文档中的所有句子都要给一个效用值，这通常只能由人工来实现。假设有 N 个专家，每人对一篇文档中的全部 n 个句子给出了效用值。那么可以定义一篇候选文摘的相对效用度量标准为：

$$RU = \frac{\sum_{j=1}^{n} \delta_j \sum_{i=1}^{N} u_{ij}}{\sum_{j=1}^{n} \varepsilon_j \sum_{i=1}^{N} u_{ij}} \tag{5-5}$$

其中 uij 专家 i 对句子 j 给出的效用值。按照所有专家给出的效用值之和进行排序，若句子 j 属于前 e 个句子，那么 ε_j 的值为 1，否则为 0。若候选文摘中包含句子 j，则 δ_j 的值为 1，否则为 0。

前面基于句子选择的方法显然粒度过大，只能精确匹配完全相同的句子。事实上，在自然语言的文本中，常常会出现两个完全不同的句子表达相同意思的情况。对同一篇文档，两个不同的人来写摘要，他们写出的摘要中可能完全没有相同的句子。因此，人们想要寻找比句子更细致的粒度，自然是单词，以及以单词为基础的短语、N 元组、单词序列等等。

最基本的词粒度的方法，是向量空间模型中的余弦距离[115]。将候选文摘和参考文摘都表示成词空间中的一个向量，则可以用向量 X 和 Y 之间的夹角余弦来评价候选文摘：

$$\cos(X,Y) = \frac{\sum_i x_i \cdot y_i}{\sqrt{\sum_i (x_i)^2} \cdot \sqrt{\sum_i (y_i)^2}} \tag{5-6}$$

另一种相似性度量的方法是单元重复度（Unit Overlap）[116]：

$$overlap(X,Y) = \frac{\|X \cap Y\|}{\|X\| + \|Y\| - \|X \cap Y\|} \tag{5-7}$$

其中 X 和 Y 都是词袋法表示的文档，也就是单词的集合，||X||表示 X 中的元素数量。

最长公共子序列（Longest Common Subsequence，LCS），也是一种常见的相关性

的度量[23]：

$$LCS(X,Y)=\frac{length(X)+length(Y)-edit_{di}(X,Y)}{2} \quad (5\text{-}8)$$

其中 X 和 Y 表示成单词的序列形式。LCS(X,Y)是 X 和 Y 之间的最长公共子序列，length(X)是序列 X 的长度，editdi(X,Y)是 X 和 Y 之间的编辑距离。

N 元组的评测方法，目前已经得到广泛的认同。所谓 n 元组就是文档中的连续 n 个单词构成的序列。N 元组的评测方法，就是比较候选文摘与参考文摘之间的 n 元组的重复情况。该方法最早是 Lin 等人于 2003 年提出的[121]，并成为 2004 年提出的 ROUGH 评测工具包中的重要组成部分。

Pyramid 方法是 2005 年提出的一种评测方法[122]，其基本思想比较候选文摘和参考文摘之间的摘要内容单元（Summarization content unit，SCU）[95][98]。该方法在实际的应用中取得了良好的效果。但是由于内容单元是人工划分的，该方法实际上是一种半自动的评测方法，代价较高。

5.2 文本理解会议中使用的自动文摘评价方法

前三届 DUC 会议中使用的是人工评价方法 SEE，之后的会议中均使用自动评价方法。最主要的评价方法 ROUGH 工具包，包含 ROUGH-N、ROUGH-L、ROUGH-W、ROUGH-S 等评价方法。此外，还有 Pyramids 和 BE 等方法。

ROUGH-N 就是计算自动文摘与参考文摘集之间的 n 元组的召回率，公式如下：

$$ROUGE\text{-}N=\frac{\sum_{s\in\{ReferenceSummaries\}}\sum_{gram_n\in S}Count_{match}(gram_n)}{\sum_{S\in\{ReferenceSummaries\}}\sum_{gram_n\in S}Count(gram_n)} \quad (5\text{-}9)$$

这里 n 代表 n 元组的长度，$Count_{match}(gram_n)$是在自动文摘与参考文摘集之间都出现的 n 元组的最大数量。显然 ROUGE-N 是一个基于召回率的度量，因为上式的分母是出现在参考文摘集中的 n 元组的总数量。如果增加更多的参考文摘，那么公式中的分母也会随之增大。每当增加参考文摘的时候，就扩展了可选择摘要的空间。通过控制加入的参考摘要类型，可以让评价侧重于文摘的不同侧面。分子中求和是针对所有参考文摘的，这意味着，如果自动文摘中的某个 n 元组在多个参考文摘中出现，那么分子的值会较大，获得的 ROUGH-N 评分也会较高。这一点是符合常识的，因为人们

通常会认为，如果一篇摘要与多篇参考摘要中共通的部分很接近的话，那么这篇摘要也是个好摘要。

ROUGE-L，衡量的是最长公共子序列。一个序列 Z=[z1,z2,…,zn]是另一个序列 X=[x1,x2,…,xm]的子序列，当且仅当存在一个严格的递增的 X 的下标序列[i1,i2,…,ik]，对于所有的 j=1,2,…,k，都有 $x_{ij}=z_j$[112]。给定两个序列 X 和 Y，它们包含很多相同的子序列，这些相同的子序列中长度最大的，称为最长公共子序列（Longest Common Subsequence，LCS）。

要在文摘评价当中使用 LCS，把一个句子看做是单词的序列。如果两个句子的最长公共子序列越长，就认为这两个句子越接近。对于长度为 m 的句子 X 和长度为 n 句子 Y，假设 X 是参考摘要的句子，Y 是候选摘要的句子，则有：

$$R_{lcs}=\frac{LCS(X,Y)}{m} \tag{5-10}$$

$$P_{lcs}=\frac{LCS(X,Y)}{n} \tag{5-11}$$

$$F_{lcs}=\frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2 P_{lcs}} \tag{5-12}$$

其中 LCS(X,Y)代表 X 和 Y 之间的最长公共子序列。基于最长公共子序列的 F 测度称为 ROUGH-L。当 X=Y 时，ROUGH-L=1。当 X 和 Y 没有公共子序列时，即 LCS(X,Y)=0 时，ROUGH-L=0。因为 ROUGH-L 只需要比较连续的序列，所以比较次数相对 n-gram 来说减少很多。另外，ROUGH-L 不需要预先知道 n-gram 的长度。

以上讨论的 ROUGH-L 是句子级的。当把 ROUGH-L 应用于文摘时，取每一个参考摘要中的句子 ri 和每一个候选摘要中的句子 cj 的最大公共子序列的合集。给定一个包含 u 个句子 m 个单词的参考文摘，和一个包含 v 个句子 n 个单词的候选文摘，文摘级的基于最长公共子序列的 F 测度为：

$$R_{lcs}=\frac{\sum_{i=1}^{u}LCS_{\cup}(r_i,C)}{m} \tag{5-13}$$

$$P_{lcs}=\frac{\sum_{i=1}^{u}LCS_{\cup}(r_i,C)}{n} \tag{5-14}$$

$$F_{lcs}=\frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2 P_{lcs}} \tag{5-15}$$

其中 $LCS_{\cup}(r_i, C)$ 表示参考文摘中的句子 ri 与候选文摘 C 的所有句子之间的最长公共子序列的合集的 LCS 评分。举例来说，设 ri=w1w2w3w4w5，C 包含两个句子：c1=w1w2w6w7w8 和 c2=w1w3w8w9w5，那么 ri 和 c1 的最长公共子序列是 w1w2，ri 和 c2 的最长公共子序列是 w1w3w5，那么 ri 与 c1 和 c2 的最长公共子序列合集是 w1w2w3w5，所以 Rlcs 的值为 4/5。

ROUGH-L 有很多优点，但是它也有明显的问题，那就是在很多情况下不能区分嵌入序列的不同。例如，给定一个参考序列 X 和两个候选序列 Y1 和 Y2 如下：

X: [A B C D E F G]

Y1: [A B C D H I K]

Y2: [A H B K C I D]

Y1 和 Y2 会得到相同的 ROUGH-L 评分。然而在这个例子中，Y1 应该是比 Y2 更好的选择，因为 Y1 的匹配是连续的。为了改善这一缺点，可以把匹配的长度计算在内，称之为带权重的最长公共子串（weighted longest common Subsequence，WLCS）。给定两个句子 X 和 Y，其 WLCS 评分可以通过如下动态规划过程来计算：

（1）初始化 c(i,j)=0;w(i,j)=0;

（2）For (i=1;i<=m;i++)

For(j=1;j<=n;j++)

If xi=yi then

k=w(i-1,j-1);

c(i,j)=c(i-1,j-1)+f(k+1)-f(k);

w(i,j)=k+1;

Otherwise

If c(i-1,j)>c(i,j-1) then

c(i,j)=c(i-1,j);

w(i,j)=0;

Else c(i,j)=c(i,j-1)

w(i,j)=0

（3）WLCS(X,Y)=c(m,n)

其中，c 是一个动态规划表，c(i,j)存储的是到 xi 和 yj 处为止的 WLCS 评分。w(i,j)则存储的是截止到 c(i,j)处的连续匹配的长度。f 是一个连续匹配的权重函数，通过调

节函数 f，就可以调整 WLCS 算法中连续匹配所占的比重。

给定两个序列，长度为 m 的 X 和长度为 n 的 Y，基于 WLCS 的 F 测度可以通过下式计算：

$$R_{wlcs} = f^{-1}\left(\frac{WLCS(X,Y)}{f(m)}\right) \tag{5-16}$$

$$P_{wlcs} = f^{-1}\left(\frac{WLCS(X,Y)}{f(n)}\right) \tag{5-17}$$

$$F_{wlcs} = \frac{(1+\beta^2)R_{wlcs}Pwlcs}{R_{wlcs}+\beta^2 P_{wlcs}} \tag{5-18}$$

其中 f-1 是 f 的逆函数。把基于权重最长公共子序列的 F 测度称为 ROUGH-W。对于之前的例子，如果使用 f(k)=k2 的 ROUGH-W 评分，得到的结果是 ROUGH-W(Y1)=0.571，ROUGH-W(Y2)=0.286。因此，在使用 ROUGH-W 评分时，Y1 比 Y2 获得更好的分数。这与一般人们的直觉选择相符。

跳跃二元组（skip-bigram）是指出现在一个句子中的一对单词，这两个单词之间可以有任意个单词。跳跃二元组的共现的统计，就是度量参考文摘与候选文摘之间的跳跃二元组的重复程度。例如有如下四个句子：

S1. police killed the gunman

S2. police kill the gunman

S3. the gunman kill police

S4. the gunman police killed

每个句子都包含 $C_4^2 = 6$ 个跳跃二元组。比如 S1 包含的跳跃二元组有 police killed、police the、police gunman、killed the、killed gunman、the gunman。S2 与 S1 相同的跳跃二元组有三个：police the、police gunman、the gunman。S4 与 S1 相同的跳跃二元组有两个：police killed、the gunman。

给定一个长度为 m 的句子 X，和长度为 n 的句子 Y，假设 X 为参考文摘，而 Y 是候选文摘，则基于跳跃二元组的 F 测度可以计算如下：

$$R_{skip2} = \frac{SKIP2(X,Y)}{C(m,2)} \tag{5-19}$$

$$P_{skip2} = \frac{SKIP2(X,Y)}{C(n,2)} \tag{5-20}$$

$$F_{skip2}=\frac{(1+\beta^2)R_{skip2}P_{skip2}}{R_{skip2}+\beta^2 P_{skip2}} \quad (5\text{-}21)$$

其中 SKIP2(X,Y)是 X 和 Y 之间相同的跳跃二元组数量，C 是组合函数。基于跳跃二元组的 F 测度称为 ROUGH-S。

ROUGH-S 的一个问题在于，如果两个句子没有相同的跳跃二元组，则会认为这两个句子毫无关系。例如：

S5. gunman the killed police

句子 S5 与 S1 之间没有任何相同的跳跃二元组，所以 ROUGH-5 的评分结果为 0。但是可以看到，S5 与 S1 不是完全无关的，S5 中单词的排列顺序与 S1 完全相反。为了能够把 S5 和与 S1 完全没有相同单词的句子区分开来，就需要将跳跃二元组与一般的单词统计相结合。这样的扩展结果称之为 ROUGH-SU。

2005 年, 哥伦比亚大学的 Mckeown、Nenkova、Passonneau 等人共同提出了 Pyramid 方法[122]。首先，将文摘句人工划分为若干个文摘内容单元(Summar ization Content Unit，SCU) , 每个 SCU 表示一个核心概念。一个 SCU 被越多的标准文摘包含就越重要。将所有 SCU 按照重要程度排序，同等重要的 SCU 排列在同一行，由上向下重要程度逐行递减，构成所谓的/ Pyramid0。通过计算自动文摘包含的 SCU 的数量和重要程度来判断自动文摘的质量。初步研究表明, Pyr amid 与人工评价有较好的一致性。但是, 由于各个语义单元的大小不固定，且同一语义的表述方式多种多样，致使自动生成这些语义单元存在很大困难。而且人工标注成本高，不利于大规模地对多个系统进行评价。

为了解决 Pyramid 方法的问题，Lin 等人又在 2006 年提出了 BE (Basic Elements) 方法[123]。首先由机器自动生成标准文摘的较小的 n 元语法单元, 然后对它们进行合并，实现自底向上的构造语义单元。这样便可以实现单元的自动识别，而且在一定程度上降低了匹配表示相同概念的不同语义单元的难度, 这些基本单元被称为 BE。具体方法是构造一个句法分析器，然后生成一棵分析树，并定义一系列剪枝规则从分析树中抽取有效的 BE。但是目前 BE 的定义、打分策略以及匹配方法等问题还没有得到很好的解决，有待通过研究得以解决。

5.3 基于概率主题模型的自动评价方法

在第二章已经讨论了在概率主题模型下的潜在主题空间中度量句子相似性的方

法，该方法可以将不同粒度的语言单元映射到相同的潜在主题空间中统一度量。本节将 LDASim 度量与 ROUGH 工具包的思想相结合，探索新的文摘评价方法，称之为 LS 评价方法。

LS 评价方法，首先要对源文档集进行 LDA 建模，使用 EM 算法，确定 α 和 β 的值。接着将要评价的候选文摘、作为标准的参考文摘以及源文档集都表示成潜在主题空间的向量形式。以文档相关的 LS-T 评测、句子相关的 LS-S 评测、句子对应的 LS-M 评测为基础，从不同的侧面定义文摘的评分，综合而得到 LS 评价方法的最终评分。

5.3.1 文档相关的 LS-T 评测

设 Corpus 表示原文档集，对其进行 LDA 建模，主题空间的维度为 k，则原文档集可以表示成向量的形式：

$$V_z(Corpus)=(t_1,t_2,\cdots,t_k) \tag{5-22}$$

共有 n 篇用于作为标准的参考文摘，记为 RS_1～RS_n，则有：

$$V_z(RS_i)=(t_{i1},t_{i2},\cdots,t_{ik}),\ \ i\in[1,n] \tag{5-23}$$

候选文摘为 CS，则有：

$$V_z(CS)=(w_1,w_2,\cdots,w_k) \tag{5-24}$$

理想的文摘，应该能真实准确的反应原文档集的内容。因此，可以直接用 LDASim 来衡量候选文摘 CS 与原文档集 Corpus 的相似程度，即：

$$LDASim(CS,Corpus)=\frac{w_1t_1+w_2t_2+\cdots+w_kt_k}{\sqrt{w_1^2+w_2^2+\cdots+w_k^2}\cdot\sqrt{t_1^2+t_2^2+\cdots+t_k^2}} \tag{5-25}$$

按照传统的文摘评价思路，候选文摘与参考文摘越接近，则候选文摘的评分应该越高。因此，可以用候选文摘与参考文摘的 LDASim 度量作为评价标准，即：

$$LDASim(CS,RS_i)=\frac{w_1t_1+w_2t_2+\cdots+w_kt_k}{\sqrt{w_1^2+w_2^2+\cdots+w_k^2}\cdot\sqrt{t_{i1}^2+t_{i2}^2+\cdots+t_{ik}^2}} \tag{5-26}$$

如果候选文摘与原文档集接近，那么候选文摘应该得到较高的分数，因为它能很好的反应原文档集的内容；如果候选文摘与某一篇参考文摘接近，那么候选文摘也应该得到较高的分数，因为它符合某些人对原文档集内容的把握。取上述评分的最大值，作为最终的评分，即：

$$LS\text{-}T(CS) = \max(LDASim(CS, OC), LDASim(CS, RS_i)) \tag{5-27}$$

由于 LDASim 度量是基于狄利克雷采样的，每一次评价的结果会因为狄利克雷采样的不同而不同。因此通常会取多次评价的平均值。

5.3.2 句子相关的 LS-S 评测

在句子层面也可以使用 LDASim 度量来评价文摘。假设有 n 篇参考文摘，分别记为 RS_1~RS_n。把这些参考文摘都看做是句子的集合，即 $RS_i = \{S_{i1}, S_{i2}, \cdots, S_{il_i}\}$，其中 l_i 为参考文摘 RS_i 中所包含的句子个数。将所有的参考文摘合并在一起，构成参考句子集，即：

$$RS = \bigcup_1^n RS_i = \{S_{r1}, S_{r2}, \cdots, S_{rL}\} \tag{5-28}$$

其中 L 为参考句子集中句子的总个数，即 L=||RS||。

将候选文摘 CS 也看做是句子的集合，即 CS={S_1,S_2,…,S_n}。依次对 CS 中的每个句子 S_i，求其与 RS 中所有句子的相关度，找出与其最相关的参考句子，其相关度作为 S_i 的评分，即：

$$Score(S_i) = \max_{1 \le j \le L}(LDASim(S_i, S_{rj})) \tag{5-29}$$

由上式可以看出，若句子 S_i 出现在参考句子集中，那么 Score(S_i)=1。如果参考句子集中没有与 S_i 完全相同的句子，那么就从参考句子集中选择与 S_i 最相关的句子，其相关度作为 S_i 的评分，Score(S_i)∈[0,1]。

按照上述方法计算候选文摘中的所有句子的评分，之后取其均值作为该候选文摘的评分，即：

$$LS\text{-}S(CS)=\frac{1}{\|CS\|}\sum Score(S_i)=\frac{1}{\|CS\|}\sum_{i=1}^{\|CS\|}\max_{1\le j\le L}(LDA\text{-}Sim(S_i,S_{rj})) \qquad (5\text{-}30)$$

其中，||CS||表示候选文摘 CS 中的句子个数。

从粒度上看，LS-S 评测也是基于句子的。但是与前面介绍的基于句子选择的计算准确率和召回率的方法不同。之前的方法要求句子完全匹配，只有完全匹配的句子才会被有效的计算。而 LS-S 则不要求句子完全匹配，只是当存在完全匹配的句子时，会得到较高的评测分数。

5.3.3 句子对应的 LS-M 评测

上面基于句子的评测方法 LS-S 有其弊端。假设参考文摘为 RS={$S_1,S_2,\ldots,S_n$}，我们构造一个候选文摘 CS={$S_1,S_1,\ldots,S_1$}，即完全由一个句子 S_1 重复 n 次得到。按照公式（5.30），会得到 LS-S(CS)=1，评价结果是完美文摘，但是实际上这样的文摘显然不是我们需要的。虽然这种极端的情况在实际中并不会出现，因为从原文档集抽取句子的过程就保证了文摘中的句子不会完全重复，但是上面的例子也充分说明了 LS-S 的弊端。当 CS 中的句子内容过于集中，含有较大冗余时，LS-S 都会得到较高的评分。

为了避免上述问题，可以将参考文摘与候选文摘中的句子进行一一对应起来。也就是说，对候选文摘 CS 中的每个句子，都找到 RS 中的一个句子与之对应，且这些对应的句子不重复。设候选文摘 CS 与参考文摘 RS 具有相同的句子数 n，即 CS={$S_{c1},S_{c2},\ldots,S_{cn}$}和 RS={$S_{R1},S_{R2},\ldots,S_{Rn}$}。对 CS 中的每一个句子 S_{ci}，用 $R(S_{ci})$表示 RS 中与 S_{ci} 相对应的句子 S_{Rj}。

采用一个贪心算法来完成句子的对应过程：使用第二章提出的 LDASim 度量，依次计算 CS 中每一个句子与 RS 中每一个句子的相似度 $LDASim(S_{ci},S_{Rj})$；找出其中相似度最高的一对句子 S_{ci} 和 S_{Rj}，记 $R(S_{ci})=S_{Rj}$；将 S_{ci} 和 S_{Rj} 从 CS 和 RS 中剔除；重复上述过程，直到 CS 中的每一个句子 S_{ci} 均求出对应的 $R(S_{ci})$和 $N(S_{ci})$。

上述贪心算法求解的前提的 CS 与 RS 具有相同的句子数。当 CS 与 RS 句子数不同时，不妨设 CS 中包含 n_c 个句子，RS 中包含 n_R 个句子。若 $n_c<n_R$，则上述贪心算法不必更改。这样得到的结果是，CS 中所有句子对应的 $R(S_{ci})$构成的集合是 RS 的一个子集。若 $n_c>n_R$，则在进行上述贪心算法时，会出现 $CS\neq\Phi$ 而 $RS=\Phi$ 的情况。此时可将 RS 重新置回初始值，算法便可继续进行。这样得到的结果是，CS 中所有句子对应

的 $R(S_{ci})$中存在重复的情况，但是 RS 中的每一个句子都至少被 CS 中的句子对应一次。

将 CS 与 RS 中的句子进行对应之后，我们就可以用类似上一节中的方法来评价 CS 与 RS 之间内容上的关联程度。

定义 5-1 对应相关度评分

设候选文摘 CS 包含 n_c 个句子，参考文摘 RS 包含 n_R 个句子。对 $\forall S_{ci} \in CS$，通过贪心算法找出所有的 $R(S_{ci})$。将 CS 中所有句子与对应的 $R(S_{ci})$之间相似度的均值称为 CS 与 RS 的对应相关度评分，记作 Csim，即

$$Csim(CS,RS)=\frac{1}{n_c}\sum_{i=1}^{n_c}LDASim(S_{ci},R(S_{ci})) \tag{5-31}$$

由定义 5-1 可以看出，CS 与 RS 的对应相关度评分，取决于二者中的句子的对应情况。当 CS 与 RS 中的句子完全相同时，二者之间存在完美的一一对应，此时有 Csim(CS,RS)=1。这就避免了 LS-S 方法存在的问题。若 CS 内容过于集中，含有较大冗余，那么 CS 中多数的句子与对应句的相关度不会很高，也就不会得到较高的对应相关度评分。

计算候选文摘 CS 与所有的参考文摘 RS_i 之间的对应相关度评分，取其最大值作为句子对应的评测方法的最终结果，即

$$LS\text{-}M(CS)=\max_i(Csim(CS,RS_i)) \tag{5-32}$$

LS-M 评测，是建立在 CS 和 RS 的对应相关度评分基础上的。如果 CS 与 RS 之间存在较好的对应关系，即 CS 中的每个句子都可以从 RS 中找到不同的句子来对应，这说明 CS 很好的涵盖了 RS 中的内容，是质量较好的文摘。当然 LS-M 也有其自身的问题，就是贪心算法所找到的未必是最佳对应。通过完全搜索可以解决最佳对应的问题，但是要耗费较多的时间。通过实践检验发现，贪心算法已经可以在大多数问题上得到比较好的对应关系。因此本文研究中依旧采用贪心算法来实现句子间的对应。

5.3.4 LS 自动评价方法

前面介绍了三种评测 LS-T、LS-S、LS-M，评测的角度各不相同，分别适应于不同的情况。

LS-T 评测，得到的是候选文摘与参考文摘或原文档集的整体相似度。这种整体相似度是基于潜在语义的，它能在一定程度上反映候选文摘内容的优劣，却无法反映出人们对文摘的阅读体验。假设对一篇候选文摘 CS 的评测结果，LS-T 评测分很高，而 LS-S 和 LS-M 的评测分都很低，说明 CS 中选择的句子与人们习惯选择的句子（RS 中的句子）存在很大的不同，也就是说 CS 很有可能不符合人们的阅读习惯，是可读性较差的文摘。因而可以用 LS-S 和 LS-M 的评测分与 LS-T 评测分的差距来衡量 CS 的可读性。

定义 5-2 LS 可读性评分

对文档集 Corpus 的候选文摘 CS，进行 LS-T、LS-S、LS-M 评测，将 LS-S 和 LS-M 的乘积与 LS-T 的比值称为 LS 可读性评分，记作 LS-Read，即：

$$LS\text{-}Read(CS)=\frac{LS\text{-}S(CS)\cdot LS\text{-}M(CS)}{LS\text{-}T(CS)} \tag{5-33}$$

LS-S 评测，得到的是候选文摘与参考文摘之间的具体句子的相关性。正如 5.3.3 节中所举例说明的一样，LS-S 有其固有的缺欠。如果一篇候选文摘 CS，LS-S 的评测分很高，而 LS-T 和 LS-M 的评测分都很低，那么很有可能像 5.3.3 节中所给出的例子一样，CS 中的句子含有较多的冗余。而 LS-S 评分与其它两项评分的差距越大，则说明冗余的程度越大，反之则冗余程度越小。因此可以用 LS-S 与其它两项评测的差距来衡量 CS 的冗余程度。

定义 5-3 LS 无冗余性评分

对文档集 Corpus 的候选文摘 CS，进行 LS-T、LS-S、LS-M 评测，将 LS-T 和 LS-M 的乘积与 LS-S 的比值称为 LS 无冗余性评分，记作 LS-Irre，即：

$$LS\text{-}Irre(CS)=\frac{LS\text{-}T(CS)\cdot LS\text{-}M(CS)}{LS\text{-}S(CS)} \tag{5-34}$$

LS-M 评测，同样也有其不足。假设 CS 中仅包含一个句子 S_1，且 S_1 包含在参考文摘 RS 中，那么对 CS 进行 LS-M 评测结果为 1。但是实际上这样的 CS 并不是人们希望得到的文摘，因为其包含的内容不够全面。在这种情况下，LS-T 的评测分就会很低。因此，可以用 LS-M 与 LS-T 的差距来衡量 CS 的全面性。

定义 5-4 LS 全面性评分

对文档集 Corpus 的候选文摘 CS，进行 LS-T、LS-M 评测，将 LS-T 的平方与 LS-M 的比值称为 LS 全面性评分，记作 LS-Tota，即：

$$LS\text{-}Tota(CS)=\frac{(LS\text{-}T(CS))^2}{LS\text{-}M(CS)} \tag{5-35}$$

以上定义了三种评分，这三种评分分别可以在一定程度上反映一篇文摘的可读性、无冗余性、全面性。将这三种评分进行加权平均，作为最终的 LS-Score 评分，即：

$$LS\text{-}Score(CS)=\lambda_1\cdot LS\text{-}Read(CS)+\lambda_2\cdot LS\text{-}Irre(CS)+\lambda_3\cdot LS\text{-}Tota(CS) \tag{5-36}$$

其中 λ_1、λ_2 和 λ_3 为加权系数，通常令 $\lambda_1+\lambda_2+\lambda_3=1$。从三种评分的定义可以看出，这三种评分理论上并不一定保证在[0,1]，因此加权系数的值也要根据实际情况来调节。在本文研究过程中进行的大量实验表明，没有极端特例的情况下，通常取 0.3、0.5、0.2 为相对较为合理的系数值。

LS 自动评价方法，就是通过计算 LS-Score 评分来对文摘进行评价。在提供了参考摘要之后，不需要进行人工干预，LS 评价方法就能综合评判候选文摘的可读性、无冗余性、全面性。只要合理的设置加权系数，就会得到比较理想的评价结果。

5.4 实验及结果分析

使用的实验数据是 DUC2006 会议的数据集，包含 50 个不同的类别，每个类别中有 25 篇文档。取其中的前 10 个类别作为实验数据。

首先采用人工的方式生成参考文摘。请三位专家阅读数据集中的文档，对每个类别都写一篇摘要。同时要求，摘要中的所有句子都要从原文档集中选择，不能自己新造句子。这样得到的文摘称为抽取参考文摘，记为 ECS。再另外请三位专家，同样编写文摘，但是不用从原文档集中抽取句子，而是完全人工新写。这样得到的文摘称为自写参考文摘，记为 ACS。

接下来采用 3.4.2 节中提到的方法一生成的文摘作为被评价的候选文摘。分别采用准确率、ROUGE-1、ROUGH-2、Pyramids、LS-T、LS-S、LS-M 七种方法对产生的文摘进行评价，表 5.1 给出的评价结果是以抽取参考文摘 ECS 为标准的。

表 5.1 以 ECS 为标准的不同方法的评价结果对比

	准确率	ROUGE-1	ROUGH-2	pyramids	LS-T	LS-S	LS-M
D0601	51.65%	0.0854	0.0831	0.0522	0.4829	0.4024	0.2507
D0602	52.89%	0.0853	0.0831	0.0529	0.4650	0.4831	0.3318
D0603	59.30%	0.0808	0.0766	0.0508	0.4283	0.4531	0.3442
D0604	50.50%	0.0803	0.0793	0.0485	0.4090	0.4262	0.2534
D0605	66.33%	0.0849	0.0816	0.0536	0.4882	0.4620	0.3423
D0606	52.58%	0.0820	0.0826	0.0519	0.4347	0.4642	0.3421
D0607	69.54%	0.0855	0.0756	0.0452	0.4212	0.4045	0.2691
D0608	55.78%	0.0835	0.0824	0.0534	0.4787	0.4762	0.3317
D0609	66.52%	0.0845	0.0823	0.0488	0.4880	0.4879	0.2743
D0610	65.12%	0.0806	0.0784	0.0521	0.4654	0.4546	0.3442

再通过 LS-T、LS-S、LS-M 评测来计算 LS 可读性评分、LS 无冗余性评分、LS 全面性评分，并最终得到 LS-Score。表 5.2 给出了与表 5.1 对应的 LS 自动评价结果。

表 5.2 以 ECS 为标准的 LS 自动评价结果

	LS-Read	LS-Irre	LS-Tota	LS-Score
D0601	0.2088	0.3008	0.9304	0.3992
D0602	0.3447	0.3193	0.6517	0.3934
D0603	0.3642	0.3254	0.5329	0.3785
D0604	0.2640	0.2431	0.6601	0.3328
D0605	0.3239	0.3618	0.6963	0.4173
D0606	0.3654	0.3203	0.5522	0.3802
D0607	0.2584	0.2802	0.6593	0.3495
D0608	0.3299	0.3335	0.6909	0.4039
D0609	0.2743	0.2743	0.8680	0.3931
D0610	0.3362	0.3524	0.6293	0.4029

再请专家对同样的文摘进行评分，评分的方法采用 5.1 节中的方法，分四个方面

进行评分，每方面的评分为 1 到 5 分，然后取四个方面的平均分作为最终的评价结果，如表 5.3 所示。

表 5.3 专家对 10 篇摘要的人工评分结果

	专家评分结果
D0601	2.75
D0602	3.25
D0603	3
D0604	2.5
D0605	3.5
D0606	3
D0607	2.75
D0608	3.5
D0609	3
D0610	2.75

将各种评价方法的结果与专家结果进行对比，计算它们之间的 pearson 相关系数，得到结果如表 5.4 所示。

表 5.4 以 ECS 为参考的自动评价方法与人工评价结果的对比

	pearson 系数
准确率	0.1122
ROUGE-1	0.3874
ROUGH-2	0.4561
pyramids	0.6378
LS-T	0.5866
LS-S	0.6664
LS-M	0.6308
LS-Score	0.6994

由表 5.4 可以看出，本文提出方法，都与人工评价结果有着较强的相似性，均好

于 ROUGH 方法。LS-S 甚至比 Pyramids 方法的效果更佳。而 LS 自动评价方法的最终结果 LS-Score 则表现最佳。

再以 ACS 为参考标准进行自动评价，得到结果如表 5.5 所示。

表 5.5 以 ACS 为标准的不同方法的评价结果对比

	准确率	ROUGE-1	ROUGH-2	pyramids	LS-T	LS-S	LS-M
D0601	0.00%	0.0744	0.0713	0.0376	0.4373	0.4062	0.2888
D0602	0.00%	0.0772	0.0731	0.0398	0.4890	0.4580	0.3470
D0603	0.00%	0.0768	0.0691	0.0369	0.4803	0.4560	0.2502
D0604	11.11%	0.0755	0.0652	0.0374	0.4026	0.4081	0.3336
D0605	0.00%	0.0757	0.0687	0.0421	0.4511	0.4276	0.3417
D0606	0.00%	0.0783	0.0706	0.0446	0.4807	0.4794	0.3500
D0607	0.00%	0.0763	0.0714	0.0365	0.4199	0.4475	0.2641
D0608	0.00%	0.0766	0.0692	0.0410	0.4936	0.4589	0.3081
D0609	0.00%	0.0754	0.0652	0.0374	0.4708	0.4932	0.2778
D0610	10.00%	0.0769	0.0669	0.0387	0.4929	0.4511	0.2726

表 5.6 以 ACS 为标准的 LS 自动评价结果

	LS-Read	LS-Irre	LS-Tota	LS-Score
D0601	0.2683	0.3110	0.6621	0.3684
D0602	0.3251	0.3705	0.6890	0.4205
D0603	0.2375	0.2635	0.9220	0.3874
D0604	0.3382	0.3291	0.4858	0.3632
D0605	0.3239	0.3605	0.5955	0.3965
D0606	0.3490	0.3510	0.6603	0.4123
D0607	0.2815	0.2478	0.6674	0.3419
D0608	0.2865	0.3315	0.7907	0.4098
D0609	0.2910	0.2652	0.7980	0.3795
D0610	0.2495	0.2978	0.8914	0.4020

同样再通过 LS-T、LS-S、LS-M 评测来计算 LS 可读性评分、LS 无冗余性评分、LS 全面性评分，并最终得到 LS-Score。表 5.6 给出了与表 5.5 对应的 LS 自动评价结果。

将表 5.5 和表 5.6 的结果与表 5.2 的专家结果进行对比，计算它们之间的 pearson 相关系数，得到结果如表 5.7 所示。

表 5.7 以 ACS 为参考的自动评价方法与人工评价结果的对比

	pearson 系数
准确率	-0.6023
ROUGE-1	0.2310
ROUGH-2	0.2835
pyramids	0.5828
LS-T	0.5729
LS-S	0.3268
LS-M	0.3421
LS-Score	0.6504

由表 5.3 可以看出，由于 ACS 参考标准是人工重新编写的，不再从原文档集中抽取句子，因此准确率的方法命中率极低，基本上无法对文摘做出评价了。其他各种方法的效果也均有所降低。其中 LS-S 和 LS-M 评价结果仍旧好于 ROUGH 工具，LS-T 的结果与 Pyramids 几乎相当。而 LS 自动评价方法的最终评分 LS-Score 则变化最小，评价效果最好，与人工评价最接近。

5.5 本章小结

本章研究的对象是自动文摘的评价方法。首先阐述了自动文摘评价方法的现状，然后对文本理解会议中的自动评价方法做了介绍和分析。在概率主题模型的潜在空间向量的相似度基础上，提出了基于概率主题模型的自动文摘评价方法 LS。该方法包含了句子相关的 LS-S、文档相关的 LS-T、句子对应的 LS-M 三种评测，并在这三种评测的基础上，定义了 LS 可读性评分、LS 无冗余性评分、LS 全面性评分，将三种评分进行加权平均得到最终的评分 LS-Score。经过实验证明，LS 评价方法是有效的，

在多数情况下与人工的评价方法更接近，效果好于 ROUGH、Pyramids 等评价方法。

结论

一、论文工作总结

自动文摘技术是自然语言理解领域的重要研究内容，具有重要的理论意义和应用价值。本文主要针对多文档自动文本摘要技术中的文本单元之间的相似性度量、文摘句的抽取算法、文摘句排序算法、文本摘要的自动评价方法等问题进行了深入的研究。具体来说主要有：

（1）研究了文本单元的相似性度量问题，这是自动生成文本摘要的基础。分析了现有度量技术的不足，并在此基础上从文本的建模方法入手，将生成模型中的潜在狄利克雷分配与传统的向量空间模型相结合，提出了 LDASim 度量。

（2）研究了文摘句的抽取问题，这是自动文本摘要技术的核心。针对现有抽取技术的不足，定义了表层距离和语义距离，提出了基于差分进化的句子聚类方法；将句子的静态权重和动态权重相结合，提出了基于概率主题模型的逐句递减规则；在此基础上，提出了基于差分进化和概率主题模型的句子抽取算法。

（3）研究了抽取出的文摘句的排序问题，这是多文档文摘中的重要问题。本文从潜层表层、本位语境多个方面刻画了句子间的相似度，提出了基于相关距离的文摘句排序算法；从原文档集中学习句子的顺序关系，提出了基于原文顺序的文摘句抽取算法；利用四种顺序标准，提出了基于概率主题模型的层次性文摘句排序算法。

（4）研究了文摘的自动评价问题。本文分析了文摘自动评价问题的现状，给出了国际会议中常用的自动评价方法。并利用基于潜在狄利克雷分配的相似性度量，提出了 LS 评测方法。

二、论文的主要创新点

在本文的研究过程中，参考和借鉴了大量国内外学者在自动文本摘要领域的研究成果，并在以下几个方面取得了创新性的研究成果：

（1）提出了基于概率主题模型的文本相似性度量方法 LDASim。该方法可以有效的避免传统向量空间模型中的稀疏问题，可以将不同粒度的文本单元映射到相同的

潜在主题空间中，从而进行统一的度量。

（2）提出了基于差分进化和概率主题模型的句子抽取算法 DEPTM。该算法通过差分进化的句子聚类，保证了文摘反映原文内容的全面性；通过静态权重与动态权重相结合的逐句递减规则，保证了文摘与原文内容的整体相似性。实验证明 DEPTM 算法的效果好于 SumBasic 系统和 NGD 方法以及 DUC2006 会议的参会系统。

（3）提出了三种文摘句的排序算法：基于相关距离的文摘句排序算法 SOCD、基于原文顺序概率的文摘句排序算法 SOSOP、基于概率主题模型的层次性文摘句排序算法 BSOP。从潜层表层、本文语境、前置后置等多种层面刻画句子之间的顺序关系，弥补了传统算法中由于评判标准的片面性而带来的不足。实验结果表明，SOCD 算法和 SOSOP 算法优于传统的 MO 和 CO 算法，BSOP 算法优于 AGL 算法。

（4）提出了基于概率主题模型的文本摘要自动评价方法 LS。该方法综合了文档相关的 LS-T、句子相关的 LS-S、句子对应的 LS-M 三种评测，可以从可读性、无冗余性、全面性三个不同的侧面对文摘进行自动评分。实验表明 LS 评价方法效果稳定，优于目前普遍采用的 ROUGH 和 Pyramids 评价方法。

三、下一步的研究工作

在本文上述研究工作的基础上，下一步的研究工作包括：

（1）本文中对文本的建模，采用的是狄利克雷分配建立的潜在主题空间。由于潜在狄利克雷分配中潜在主题分布的随机性，造成评分结果的范围较大。如何削弱甚至消除这种随机性的影响，是下一步研究的重点。如果能够寻找到一种随机性更低的模型代替狄利克雷分布，必将使自动文摘的效果更上一层楼。

（2）本文使用的狄利克雷建模需要 EM 算法的反复迭代才能完成，差分进化算法也需要较多代的繁衍才能最终结束。这就导致本文所提出的自动文摘生成方法耗时较长。如何提高效率缩短时间，是下一步研究的一个重要课题。

（3）本文提出的文摘句排序算法较为复杂，虽然比传统方法有所提高，但提高幅度较小，与人工排序的差距还是相当大。因此句子排序方法的研究还有很大的提高空间。

（4）本文提出的自动文摘生成方法和自动文摘评价方法，都是基于文本的表层统计信息的，这也是目前研究的主流。但是由于统计方法的雷同性，使得自动文摘的

评价效果并不理想。如果能够将深层语义挖掘加入到文摘评价工具中，应该能实现更为理想的效果。这是下一步研究中的重要工作。

参考文献

[1]. Radev D R, Hovy E, McKeown K. Introduction to the special issue on summarization[J]. Computational Linguistics, 2002, 28(4): 399-408.

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

[3]. Baxendale P. Machine-made index for technical literature - an experiment[J]. IBM Journal of Research Development, 1958, 2(4):354-361.

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

[5]. Mani J. Automatic Summarization[M]. Amsterdam/Philadephia: John Benjamins Publishing Company. 2001.

[6]. Rush J E, Salvador R, Zamora A. Automatic abstracting and indexing: Production of indicative abstracts by application of contextual inference and syntactic coherence criteria[J]. Journal of the American Society for Information Science, Vol.22, No.4.

[7]. Rau L F, Jacobs P S, Zernik U. Information Extraction and Text Summarization Using Linguistic Acquisition[J]. Information Processing and Management. Vol.25, No.4.1989.

[8]. Kupiec J, Pedersen J, Chen F. A trainable document summarizer[C]. Proceedings SIGIR '95, pages 68-73, New York, NY, USA.

[9]. Aone C, Okurowski M E, Gorlinsky J, Larsen B. A trainable summarizer with knowledge acquired from robust nlp techniques[C]. Advances in Automatic Text Summarization, pages 71-80. 1999. MIT Press.

[10]. Lin C Y, Hovy E. Identifying topics by position[C]. In Proceedings of the Fifth conference on Applied natural language processing, pages 283-290, 1997.

[11]. Lin C Y. Training a selection function for extraction[C]. In Proceedings of CIKM '99, pages 55-62, 1999.

[12]. Conroy J M, O'leary D P. Text summarization via hidden markov models[C]. In Proceedings of SIGIR '01, pages 406-407, 2001.

[13]. Osborne M. Using maximum entropy for sentence extraction[C]. In Proceedings of the ACL'02 Workshop on Automatic Summarization, pages 1-8, 2002.

[14]. Nenkova A. Automatic text summarization of newswire: Lessons learned from the document understanding conference[C]. In Proceedings of AAAI 2005, Pittsburgh, USA.

[15]. Svore K, Vanderwende L, Burges C. Enhancing single-document summarization by combining RankNet and third-party sources[C]. In Proceedings of the EMNLP-CoNLL, pages 448-457, 2007.

[16]. Burges C, Shaked T, Renshaw E, Lazier A, Deeds M, Hamilton N, Hullender G. Learning to rank using gradient descent[C]. In ICML '05:Proceedings of the 22nd international conference on Machine learning, pages 89-96, New York, NY, USA. 2005.

[17]. Barzilay R, Elhadad M. Using lexical chains for text summarization[C]. In Proceedings ISTS'97. 1997.

[18]. McKeown K R, Radev D R. Generating summaries of multiple news articles. In Proceedings of SIGIR '95, pages 74-82, 1995.

[19]. 秦兵, 刘挺, 王洋等. 基于常问问题集的中文问答系统的研究[J]. 哈尔滨工业大学学报, 2003, 35(10):1179 - 1182.

[20]. Ono K, Sumita K, Miike S. Abstract generation based on rhetorical structure extraction[C]. In Proceedings of Coling '94, pages 344-348, 1994.

[21]. Marcu D. Improving summarization through rhetorical parsing tuning[C]. In Proceedings of The Sixth Workshop on Very Large Corpora, pages 206-215, pages 206-215, 1998

[22]. Selman B, Levesque H J, Mitchell D G. A new hard satis_ability problems[C]. In AAAI, pages 440-446, 1992.

[23]. Radev D, Teufel S, Saggion H, Lam W, Blitzer J, Qi H, Celebi A, Liu D, Drabek E. Evaluation Challenges in Large-Scale Document Summarization. In Proceeding of the 41st meeting of the Association for Computational Linguistics, Sapporo, Japan, 2003.

[24]. McKeown K, Klavans J, Hatzivassiloglou V, Barzilay R, Eskin E. Towards multidocument summarization by reformulation: Progress and prospects[C]. In AAAI/IAAI, pages 453-460. 1999.

[25]. Barzilay R, McKeown K, Elhadad M. Information fusion in the context of multi-document summarization[C]. In Proceedings of ACL '99. 1999.

[26]. Carbonell J, Goldstein J. The use of MMR, diversity-based reranking for reordering documents and producing summaries[C]. In Proceedings of SIGIR '98, pages 335-336, New York, NY, USA. 1998.

[27]. Radev D R, Jing H, Budzikowska M. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies[C]. In NAACL-ANLP 2000 Workshop on Automatic summarization, pages 21-30, Morristown, NJ, USA. 2000.

[28]. Collins M. Head-Driven Statistical Models for Natural Language Parsing[D]. University of Pennsylvania.

[29]. William M Darling, Fei Song. PathSum: A Summarization Framework Based on Hierarchical Topics[C]. Proceedings of the Workshop on Automatic Text Summarization, pages 5-16, May 24, 2011

[30]. Mani I, Bloedorn E. Multi-document summarization by graph search and matching[C]. In AAAI/IAAI, pages 622-628, 1997.

[31]. Salton G, Buckley C. On the use of spreading activation methods in automatic information[C]. In Proceedings of SIGIR '88, pages 147-160, 1988.

[32]. Radev D R, Jing H, Stys M, Tam D. Centroid-based summarization of multiple documents[J]. Information Processing and Management 40 (2004), 40:919-938.

[33]. Evans D K. Similarity-based multilingual multi-document summarization[R]. Technical Report CUCS-014-05, Columbia University. 2005.

[34]. Hovy E, Lin C Y. Automated text summarization in summarist[M]. In Advances in Automatic Text Summarization, Mani and Maybury, pages 81-94, 1999.

[35]. Mani I, Firmin T, House D, Klein G, Sundheim B, Hirschman L. The TIPSTER Summac Text Summarization Evaluation. In Proceedings of the 9th Meeting of the European Chapter of the Association for Computational Linguistics, 77–85, 1999.

[36]. Dempster A P, Laird N M, Rubin D B. Maximum likelihood from incomplete data via the EM algorithm[J]. Journal of the Royal Statistical Society B, 1977, V39(1): 1–38.

[37]. Lin C Y. Rouge: A package for automatic evaluation of summaries[C]. In Marie-Francine Moens, S. S., editor, Text Summarization Branches Out: Proceedings of the ACL-04 Workshop, pages 74-81, Barcelona, Spain. 2004.

[38]. Lin C Y, Hovy E. Manual and automatic evaluation of summaries[C]. In Proceedings of the ACL-02 Workshop on Automatic Summarization, pages 45-51, Morristown, NJ, USA. 2002.

[39]. J-L.Minel, S.Nugier, G.Piat. How to Appreciate the Quality of Automatic Text Summarization[A]. Proceedings of the ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization[C].New

Brunswick, New Jersey: Association for Computational Linguistics, 1997, 25-30.

[40]. H.Saggion, G.Lapalme. Concept Identification and Presentation in the Context of Techincal Text Summarization[A]. Proceedings of the Workshop on Automatic Summarization[C]. New Brunswick, New Jersey: Association for Computational Linguisitics, 2000, 1-10.

[41]. J.Rowley. Abstracting and Indexing[M]. London:Clive Bingley, 1982.

[42]. A.Morris, CiKasper, D.Adams. The Effects and Limitations of Automatic Text Condensation on Reading Comprehension Performance[J]. Information Systems Research. 1992,3(1):17-35.

[43]. I.Mani, T.Firmin, D.House, et al. The TIPSTER SUMMAC Text Summarization Evaluation: Final Report. MITRE Technical Report MTR 98W0000138[M]. McLean, VA: The MITRE Corporation, 1998.

[44]. D.R.Radev, H.Jing, M.Budzikowska. Summarization of multiple documents: clustering, sentence extraction, and evaluation[A]. Proceedings of the Workshop on Automatic Summarization[C]. New Brunswick, New Jersey: Association for Computational Linguistics, 2000,21-30.

[45]. R.L.Donaway, K.W.Drummey, L.A.Landauer, R.Harshman. Indexing by Latent Semantic Analysis[J]. Jounal of the American Society for Information Science, 1990, 41(6):391-407.

[46]. C.Y Lin. Summary Evaluation Environment[OL]. http://www.isi.edu/~cyl/SEE

[47]. K.Papineni, S.Roukos, T.Ward, W.J.Zhu. BLEU: a Method for Automation Summarization of Machine Translation[M].IBM Research Report RC22176 (W0109-022), 2001

[48]. C.Y.Lin, E.Hovy. Automatic Evaluation of Summarization Using N-gram Co-Occurrence Statistics[A]. Proceedings of the Human Technology Conference[C]. Edmonton, Canada,2003.

[49]. 穗志方,俞士汶. 基于骨架依存树的语句相似度计算模型[A] . 中文信息处理国际会议(ICCIP’98) [C]. 1998 :23 - 27.

[50]. Salton G, Wong A , Yang C S. A vector space model for automatic indexing[J]. Communications of the ACM , 18 (11) : 613 – 620, 1975.

[51]. Baetz-Yates R,Ribeiro-Neto B. Modern Information Retrieval[M]. Association for Computing Machine(ACM) Press,1999.

[52]. Lewis D.D. Representation and Learning in Information Retrieval[D]. Ph.D. dissertation, University of Massachusetts,1992.

[53]. Lewis,D.D. An Evaluation of Phrasal and Clustered Representations on a Text Categorization task[C]. Proceedings of the 15th annual international ACM SIGIR conference on Research and

development in information retrieval,Copenhagen, Denmark, 1992: 37-50.

[54]. C. Apte, F. Damerau, and S.M. Weiss. Automated Learning of Decision Rules for Text Categorization[J]. ACM Transactions on Information Systems,1994.

[55]. Dumais, S. T., Platt, J., Heckerman, D., and Sahami, M. Inductive learning algorithms and representations for text categorization[C]. Proceedings of the 7th ACM International Conference on Information and Knowledge Management(CIKM-98).Washington, US, 1998:148 – 155.

[56]. Lovis,J.B. Development of a Stemming Algorithm[J]. Mechanical Translation and Computational Linguistics, 1968,11:22-31.

[57]. Fuhr, N. and Buckley, C. A probabilistic learning approach for document indexing[J]. ACM Transactions on Information Systems, 1991,9 (3):223 – 248.

[58]. Z. Fei, J. Liu, G. Wu. Sentiment classification using phrase patterns[C]. Proceedings of the 4th International Conference on Computer and Information Technology(CIT-04), Wuhan,China, 2004: 1147 – 1152.

[59]. Lewis,D. D., Croft,W.Bruce. Term Clustering of syntactic Phrases[C]. Proceedings of the 13th Annual International ACM SIGIR Conference on Research and Development in Inforamtion Retrieval,1990: 385- 404.

[60]. Sam Scott, S.Matwin. Feature Engineering for Text Classication[C]. Proceedings of the 16th International Conference on Machine Learning (ICML-99),1999.

[61]. 姚天顺,朱靖波等. 自然语言理解—一种让机器懂得人类语言的研究[M].北京:清华大学出版社. 2002.10.第二版

[62]. Salton G., Christopher Buckley. Term-weighting approaches in automatic text retrieval, 1998, 24(5):523-523.

[63]. Miller G.A, Beckwidth R, Fellbaum C. Introduction to Wordnet: An On-line Lexical database[J]. International Journal of Lexicography, 1990, 3(4):235-244.

[64]. Stephanie C,Narayanyan K. Semantic Feature Selection Using Wordnet[A]. Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, 2004,9:166-172.

[65]. Masuyama T. Nakagawa H. Applying cascaded feature selection to SVM text categorization[A]. In the DEXA Workshops, 2002:241-245.

[66]. Fodor I K. A Survey of Dimension Reduction Techniques[R]. LLNL technical report, UCRL- ID-148494, http: //www.llnl.gov/CASC/sapphire/pubs.html, 2002

[67].J Lin, D Gunopulos. Dimensionality Reduction by Random Projection and Latent Semantic Indexing[C]. In: Text Mining Workshop, at the 3rd SIAM International Conference on Data Mining, 2003

[68].Kaski S.Dimensionality Reduction by Random Mapping: Fast Similarity Computation for Clustering[C].In: Proceedings of International Joint Conference on Neural Networks(IJCNN'98) , IEEE Service Center, Piscataway, NJ, 1998: 413~418

[69].Bingham E, Mannila H.Random. Projection in Dimensionality Reduction: Applications to Image and Text Data[C].In: Proc SIGKDD(2001) , 2001: 245~250

[70].Lee D, Seung H.Algorithms for Non- negative Matrix Factorization[C]. In: Adv Neural Info Proc Syst, 2001; 13: 556~562

[71].Lee D, Seung H.Learning the Parts of Objects by Nonnegative Matrix Factorization[J].Nature, 1999; 401(21) : 788~791

[72].George Karypis, Eui- Hong(Sam) Han.Concept Indexing: A Fast Dimensionality Reduction Algorithm with Applications to Document Retrieval & Categorization[C].In: ACM CIKM Conference, 2000

[73].S Dumais, G Furnas, T Landauer et al.Using Latent Semantic Analysis to Improve Access to Textual Information[C].In: Proceedings of the Conference on Human Factors in Computing Systems CHI'88, Washington, DC, USA, 1988

[74].于秀林, 任雪松. 多元统计分析[M]. 北京: 中国统计出版社, 1999

[75].Blei D M, Griffiths T L, Jordan M I, Tenenbaum J B. Hierarchical topic models and the nested Chinese restaurant process[M]. In Advances in Neural Information Processing Systems 16. Cambridge, MA: MIT Press. 2004.

[76].Griffiths T L, Steyvers M. A probabilistic approach to semantic representation[C]. In Proceedings of the 24th Annual Conference of the Cognitive Science Society. 2002

[77].Griffiths T L, Steyvers M. Prediction and semantic association[M]. In Neural information processing systems 15. Cambridge, MA: MIT Press. 2003.

[78].Griffiths T L. Steyvers M. Finding scientific topics[C]. Proceedings of the National Academy of Science, 101, 5228-5235. 2004.

[79].Hofmann T. Probabilistic Latent Semantic Analysis[C]. In Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence. 1999.

[80]. Hofmann T. Unsupervised Learning by Probabilistic Latent Semantic Analysis[J]. Machine Learning Journal, 42(1), 177-196. 2001.

[81]. Rosen-Zvi M, Griffiths T, Steyvers M, Smyth P. The Author-Topic Model for Authors and Documents[C]. In 20th Conference on Uncertainty in Artificial Intelligence. Banff, Canada, 2004.

[82]. Steyvers M, Smyth P, Rosen-Zvi M, Griffiths T. Probabilistic Author-Topic Models for Information Discovery[C]. The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Seattle, Washington. 2004.

[83]. Griffiths T L, Steyvers M, Blei D M, Tenenbaum J B. Integrating topics and syntax[C]. In Advances in Neural Information Processing 17. Cambridge, MA: MIT Press. 2005

[84]. Hofmann, T., Puzicha, J., and Jordan, M. I. Unsupervised learning from dyadic data[J]. In Advances in Neural Information Processing Systems. 1999. vol. 11.

[85]. Saul L, Pereira F. Aggregate and mixedorder Markov models for statistical language processing[C]. In Proceedings of the 2nd International Conference on Empirical Methods in Natural Language Processing. 1997.

[86]. McLachlan G, Basford K E. Mixture Models[M]. Marcel Dekker, INC, New York Basel, 1988.

[87]. Daniel Jacob Gillick. The Elements of Automatic Summarization[D]. Berkeley: University of California, 2011.

[88]. E Filatova, V Hatzivassiloglou. Eventbased extractive summarization[C]. In Proceedings of ACL Workshop on Summarization, volume 111, 2004.

[89]. D.S. Hochbaum. Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems[M]. PWS Publishing Co. Boston, MA, USA, pages 94 – 143, 1996.

[90]. G L Nemhauser, L A Wolsey. Integer and combinatorial optimization[M], volume 18. Wiley New York, 1988.

[91]. A Nenkova, L Vanderwende. The impact of frequency on summarization[J]. Technical Report MSR-TR-2005-101, Microsoft Research, Redmond, Washington, 2005.

[92]. G J Rath, A Resnick, R Savage. The formation of abstracts by the selection of sentences: Part 1: sentence selection by man and machines[J]. American Documentation, 2(12):139 – 208. 1961.

[93]. Hans Halteren, Simone Teufel. Examining the consensus between human summaries: initial experiments with factoid analysis[C]. In HLT-NAACL DUC Workshop. 2003.

[94]. Dragomir Radev, Simone Teufel, Horacio Saggion, W. Lam. Evaluation challenges in large-scale multidocument summarization[C]. In ACL. 2003.

[95]. Ani Nenkova, Rebecca Passonneau. Evaluating content selection in summarization: The pyramid method[C]. In Proceedings of HLT/NAACL 2004.

[96]. Terry Copeck, Stan Szpakowicz. Vocabulary agreement among model summaries and source documents[C]. In Proceedings of the Document Understanding Conference DUC’04. 2004.

[97]. Michele Banko, Lucy Vanderwende. Using n-grams to understand the nature of summaries[C]. In Proceedings of HLT/NAACL’04. 2004.

[98]. David Kirk Evans, Kathleen McKeown. Identifying similarities and differences across english and arabic news[C]. In Proceedings of the International Conference on Intelligence Analysis. 2005.

[99]. Jones K S. Automatic summarizing: The state of the art. Information Processing and Management, 43, 1449–1481. 2007.

[100]. Li Y, Luo C, Chung S M. Text clustering with feature selection by using statistical data[J]. IEEE Transactions on Knowledge and Data Engineering, 20, 641–652. (2008).

[101]. Ramiz M. Aliguliyev. A new sentence similarity measure and sentence based extractive technique for automatic text summarization[J]. Expert Systems with Applications, 36, 7764-7772, 2009.

[102]. Cilibrasi R L, Vitányi P M B. The Google similarity measure[J]. IEEE Transaction on Knowledge and Data Engineering, 19, 370–383. 2007.

[103]. Regina Barzilay , Noemie Elhadad , and Kathleen R. McKeown. Sentence Ordering in Multidocument Summarization[A]. In : Proceedings of the 1st Human Language Technology Conference[C]. San Diego , California , 2001 :32 - 38.

[104]. R. Barzilay, N. Elhadad, and K.R. McKeown. Inferring strategies for sentence ordering in multidocument news summarization[J]. Journal of Artificial Intelligence Research, 2002,17:35-55.

[105]. Jing, H. Summary generation through intelligent cutting and pasting of the input document. Tech. rep., Columbia University. 1998.

[106]. McKeown K, Barzilay R, Evans D, Hatzivassiloglou V, Kan M, Schiffman B, Teufel S. Columbia multi-document summarization: Approach and evaluation[C]. Document Understanding Workshop, 2001.

[107]. Hynek J, Jezek K. Practical Approach to Automatic Text Summarization. In Proceedings of the ELPUB’03 Conference, Guimaraes, Portugal, 378–388, 2003.

[108]. Cohen W, Schapire R, Singer Y. Learning to order things[J]. Journal of Artificial Intelligence, 1999, 10:243-270.

[109]. Ernst Althaus, Nikiforos Karamanis, Alexander Koller. Computing locally coherent discourses[C]. In Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL’04), Main Volume, pages 399–406, July 2004.

[110]. D Bollegala, N Okazaki, M Ishizuka. A bottom-up approach to sentence ordering for multi-document summarization[J]. Information Processing & Management, 2010, 46(1):89–109.

[111]. Tsutomu Hirao, Jun Suzuki, Hideki Isozaki, Eisaku Maeda. Dependency-based Sentence Alignment for Multiple Document Summarization[A]. In : 20th International Conference on Computational Linguistics[C]. 2004: 446 - 452.

[112]. Cormen, T R, C E Leiserson, R L Rivest. Introduction to Algorithms[C]. The MIT Press. 1989.

[113]. Josef S, Karel J. Evaluation Measures For Text Summarization[J]. Computing and Informatics, Vol. 28, 1001–1026, 2009.

[114]. Zhuli Xie, Xin Li, Barbara Di Eugenio, Weimin Xiao, ThomasM Tirpak, Peter C Nelson. Using Gene Expression Programming to Construct Sentence Ranking Functions for Text Summarization[A]. In 20th International Conference on Computational Linguistics[C]. pages 1381-1384, 2004.

[115]. Salton G. Automatic Text Processing[M]. Addison-Wesley Publishing Company, 1988.

[116]. Saggion, H Radev, D Teufel, S Lam, W Strassel S. Developing Infrastructure for the Evaluation of Single and Multi-Document Summarization Systems in a Cross-Lingual Environment[C]. In Proceedings of LREC, Las Palmas, Spain, 2002.

[117]. Marcu D. The Automatic Const ruction of Large Scale Corpora for Summarization Research [C]. In Proceedings of ACM SIGIR 1999, 137-144, University of California, Berkely, 1999.

[118]. Jing H, K R McKeown. The Decomposition of Human Written Summary Sentences[C]. In Proceedings of ACM SIGIR 1999, 129-136, University of California, Berkeley, 1999.

[119]. Van Rijsbergen C J. Information Retrieval, 2nd edition [M]. Dept. of Computer Science, University of Glasgow. 1979.

[120]. David M Blei, Andrew Y Ng, Michael I Jordan. Latent Dirichlet Allocation[J]. Journal of Machine Learning Research Vol.3, pages 993-1022, 2003

[121]. Lin Ch, Hovy E. Automatic Evaluation of Summaries Using n-Gram Co-Occurrence Statistics[C].

In Proceedings of HLT-NAACL, Edmonton, Canada, 2003.

[122]. Nenkova A, Passonneau R. Evaluating Content Selection in Summarization: The Pyramid Method. In Document Understanding Conference, Vancouver, Canada, 2005.

[123]. Eduard Hovy, Chin Yew Lin, Liang Zhou. Automated Summarization Evaluation wit h Basic Elements[C]. In Proceedings of the Fifth Conference on Language Resources and Evaluation (LREC 2006). Genoa, Italy.

[124]. 秦兵, 刘挺, 李生.多文档自动文摘综述[J].中文信息学报. Vol.19,No.16,2006

[125]. Yohei Seki. Sentence Extraction by tf/ idf and Position Weighting from Newspaper Articles [C] . Proceedings of the Third NTCIR Workshop on Research in Information Retrieval. Automatic Text Summarization and Question Answering, Tokyo, 2002:55-59.

[126]. Rie Kubota Ando, Branimir K Boguraev, Roy J Byrd, Mary S Neff. Multi-document Summarization by Visualizing Topical Content[C]. ANLP-NAACL 2000. Advanced Summerization Workshop, Seattle, WA, 2000:12-19.

攻读学位期间发表论文与研究成果清单

科研项目

1. 野战辅助决策、战场信息挖掘与知识获取技术，“十五”国防科研项目。

2. “XX 情报信息智能化处理”之通用信息处理平台技术。

撰写论文情况

[1] Tiedan Zhu, Kan Li. The Similarity Measure Based On LDA For Automatic Summarization. Procedia Engineering. 2012,29: 2944-2949.（EI Compendex，Accession number: 20121214883066）

[2] Tiedan Zhu, Xinxin Zhao. Five Criterions Based Approach to Sentence Ordering for Multi-Document Summarization. International Journal of Digital Content Technology and its Applications(JDCTA). （已录用，EI 刊源）

[3] Tiedan Zhu, Qiongxin Liu. Sentence descending algorithm for automatic text summarization. 2011 International Conference on Computational and Information Sciences（ICCIS 2011），（EI Compendex，Accession number: 20115114621644）

[4] Tiedan Zhu, Xinxin Zhao, Yushu Liu. A new text classification model based on the Sentence Space. 2005 International Conference on Machine Learning and Cybernetics（ICMLC 2005），（EI Compendex，Accession number: 2005509539028）

[5] Tiedan Zhu, Xinxin Zhao. An Improved approach to sentence ordering for multi-document summarization. 2012 4th International Conference on Machine Learning and Computing（ICMLC 2012），（EI Compendex）

[6] Tiedan Zhu, Qiongxin Liu. Topic Space Model based sentence descending algorithm for automatic text summarization. International Conference on Computational Intelligence and Software Engineering（CiSE 2011）

[7] Xinxin Zhao, Tiedan Zhu, Yushu Liu. Document classification in different granularity. Computer Engineering, Vol.32, 2006. （EI Compendex，Accession number: 20065110317804）

致谢

感谢我的导师刘玉树教授。刘老师在学术上有着严谨求真的治学态度，在工作中有着精益求精的工作作风，在生活中有着谦逊宽厚的人格魅力，这些都深深的影响着我，为我树立了学习的榜样。在理工大学这近十年的求学生涯中，不论是物质上还是精神上，刘老师都竭尽全力给我最大的支持和关爱。能够成为刘老师的学生，是我一生中最幸运的事之一。从刘老师身上学到的东西，将是我一生的宝贵财富。

感谢牛振东教授。当我处于人生的低谷，想要放弃的时候，是牛教授把我拉回到正常的学习轨道上来。牛教授的教诲和鼓励，让我鼓起勇气大胆前行。

感谢李侃老师。在我开始论文研究的初期，李老师就给了我十分中肯的建议，帮助我对论文课题做出了合理的规划。在论文写作过程当中，李老师不断的提出了建设性的修改意见，帮助我提高课题研究的理论深度，使我的论文更加精彩。

感谢实验室的所有老师。感谢贺跃老师，总是像慈母一样关怀着我；感谢高春晓老师，帮我评审论文，并在诸多事务上为我提供了大力的支持；感谢刘琼昕老师，协助我申请实验数据，并在日常的学习中给予我无微不至的关怀；感谢孙新老师，在科研、生活、工作等方方面面都给予我极大的帮助；感谢郑军老师，适时的提点让我看到前进的方向。

感谢黄春光参谋长和张全新老师。即使身在国外，还不断的关心着我的学业。尽力为我提供各种可能的帮助。在思想上鼓励我，让我树立信心；在科研中帮助我，使我不断进步。

感谢实验室的赵欣欣、吕琳、李文清、白敬华、张凯、曹朝、曹健、高影繁、周世斌、张军、王丽、索红光、付调平、陈云飞、阎光伟、刘旭红、张国英、沙芸、王军等兄弟姐妹们。他们与我并肩战斗，一起攻克难关；他们的学术成果为我的研究打下基础；他们总是不遗余力的为我提供各种帮助。我取得的每一点成果，其中都包含着他们的劳动；我取得的每一点进步，他们都由衷的为我感到高兴。同门的所有师兄弟，都是我永生难忘的好朋友。

感谢我的父亲和母亲。他们是这个世界上最爱我的人。我的每一点挫折或进步，都无时无刻不在牵动着他们的心。他们永远默默的站在我身后，永远无私的将他们的一切给予我。我相信我可以战胜一切困难，因为我有世界上最好的爸爸妈妈。

感谢我的妻子赵元芳。她始终陪伴在我的身边。当我低迷时，她鼓励我；当我忘形时，她点醒我；当我懈怠时，她鞭策我；当我犯错时，她宽容我。是她的默默付出，才能使我的论文顺利完成。

感谢本文所有引用文献的作者；感谢本文的评阅专家；感谢所有关心我的亲人朋友们……本文的完成，包含着众多人的心血，请原谅我无法一一点名致谢。所有人对我的帮助和关怀，我将铭记一生，并化为我继续向前的动力。