基于文档主题结构的关键词抽取方法研究

（申请清华大学工学博士学位论文）

培 养 单 位：计算机科学与技术系
学　　科：计算机科学与技术
研 究 生：刘 知 远
指 导 教 师：孙 茂 松 教 授

二〇一一年三月

Research on Keyword Extraction Using Document Topical Structure

Dissertation Submitted to

Tsinghua University

in partial fulfillment of the requirement

for the degree of

Doctor of Engineering

by

Liu Zhiyuan

(Computer Science and Technology)

Dissertation Supervisor : Professor Sun Maosong

March, 2011

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：

清华大学拥有在著作权法规定范围内学位论文的使用权，其中包括：（1）已获学位的研究生必须按学校规定提交学位论文，学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；（2）为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆、资料室等场所供校内师生阅读，或在校园网上供校内师生浏览部分内容；（3）根据《中华人民共和国学位条例暂行实施办法》，向国家图书馆报送可以公开的学位论文。

本人保证遵守上述规定。

（保密的论文在解密后应遵守此规定）

作者签名：_____ 导师签名：_____

日　　期：_____ 日　　期：_____

摘 要

关键词是快速获取文档主题的重要方式，在信息检索和自然语言处理等领域均有重要应用。传统的方法仅依靠词汇的统计信息进行推荐，没有考虑文档主题结构对关键词抽取的影响。本文主要研究考虑文档主题结构的关键词抽取方法。本文针对文档主题结构在关键词抽取中的重要作用，从四个方面提出考虑文档主题结构的关键词抽取方法：基于文档内部信息构建主题的关键词抽取，基于隐含主题模型构建主题的关键词抽取，综合利用隐含主题模型和文档结构的关键词抽取，以及基于文档与关键词主题一致性的关键词抽取。论文工作包括：

基于文档内部信息，利用文档的词聚类算法构建文档主题，进行关键词抽取。 该方法仅依靠文档内部信息，通过度量文档中词与词之间的相似度，利用聚类的方法构建文档主题，并根据不同主题在文档中的重要性，进行关键词抽取。实验证明，该方法能够在一定程度上发现文档主要话题，并抽取出与文档主题相关的关键词，提高了关键词对文档主题的覆盖度。

基于文档外部信息，利用隐含主题模型构建文档主题，进行关键词抽取。 针对基于文档内部信息通过聚类算法进行关键词抽取受限于文档提供信息不足的缺点，提出利用机器学习算法中广泛使用的隐含主题模型构建文档主题，进行关键词抽取。并针对隐含主题模型训练速度较慢的瓶颈，提出了一种高效的并行隐含主题模型。实验证明，该方法能够更好地构建文档主题，并有效抽取关键词。

综合利用隐含主题模型和文档结构信息，进行关键词抽取。 针对隐含主题模型无法考虑文档结构信息的缺点，提出综合利用隐含主题模型和文档结构信息的方法——基于主题的随机游走模型——进行关键词抽取。该方法一方面能够通过隐含主题模型构建文档主题，同时能够通过文档图的随机游走模型考虑文档结构为关键词抽取提供信息，实验证明，该方法能够综合隐含主题模型和文档结构信息进行关键词抽取的优势，有效抽取关键词。

基于文档与关键词主题一致性的前提，提出基于机器翻译模型的关键词抽取方法。 针对文档和关键词之间存在较大词汇差异的问题，基于文档和关键词主题一致性的前提，提出利用机器翻译中的词对齐模型计算文档中的词到关键词的翻译概率，然后进行关键词抽取。实验证明该方法能够有效的建立文档词汇与关键词之间的语义联系，能够有效推荐关键词。

关键词： 语言网络； 自然语言处理； 关键词抽取； 文档主题

Abstract

Keywords are an important way to catch the main idea of a document for human beings. Automatic keyword extraction plays an important role in information retrieval and natural language processing, etc. Traditional methods for keyword extraction simply rank keywords according to the statistical information of words, without considering the influence of document topic structure to keyword extraction. This thesis focus on the document topic structure, study the methods to extract keywords by considering document topics, including using internal information of documents for keyword extraction, using external information via latent topic models for keyword extraction, integrating latent topic models and document structure for keyword extraction, and keyword extraction based on the topic consistency of documents and keywords. We introduce the four methods in detail as follows:

Using internal information of documents for keyword extraction. This method simply uses the internal information of a document. By measuring the semantic relatedness between words, we perform clustering for these words, and each cluster can be regarded as a topic of this document. We select an exemplar word for each cluster. Activated by these exemplar words, we extract keywords from the document. This method can extract keywords based on the topical clusters which make the extracted keywords have better coverage.

Using latent topic models for keyword extraction. Keyword extraction based on word clustering falls short because the internal information of a document is not sufficient to identify the document topics. We thus propose to use latent topic models to build topics based on large-scale datasets. Moreover, we propose a parallel algorithm for Latent Dirichlet Allocation (LDA), a typical latent topic model, to speedup the learning process of LDA.

Using latent topic models and document structure for keyword extraction. Keyword extraction using latent topic models does not take the document structure into account. We propose a new method to integrate latent topic models and document structure together for keyword extraction. This method takes advantages of both latent topic models and graph-based methods for keyword extraction.

Using topic consistency of document and keywords to solve vocabulary gap in keyword extraction. In many cases, there is a large vocabulary gap between a document and its keywords. By assuming the document and its keywords are topically consistent, we propose a method using word alignment models in statistical machine translation for keyword extraction.

Key words: language network; natural language processing; keyword extraction; document topics

目 录

主要符号对照表

d	文档*d*
w	词项*w*
p	关键词*p*
D	文档集合
W	词项集合
M	推荐关键词个数
TF	term frequency
TFIDF	term frequency – inverse document frequency
C	聚类个数
SVD	奇异值分解(Singular Value Decomposition)
SP	谱聚类(Spectral Clustering)
HC	层次聚类(Hierarchical Clustering)
AP	信任传播聚类(Affinity Propagation)
NGD	归一化Google距离(Normalized Google Distance)
PMI	点互信息(Pointwise Mutual Information)
K	主题个数
z	隐含主题*z*
LDA	隐含狄利克雷分配模型 (Latent Dirichlet Allocation)
AD-LDA	Approximate Distributed LDA
AS-LDA	Asychronous Distributed LDA
PLDA	Parallel LDA
TPR	Topical PageRank
SMT	统计机器翻译(Statistical Machine Translation)
WAM	词对齐模型(Word Alignment Model)
NB	朴素贝叶斯模型(Naive Bayes)
*k*NN	*k*近邻(*k* Nearest Neighborhood)

第 1 章 引言

1.1 研究背景

一篇文档的关键词(keyword)通常是几个词或者短语，作为对该文档主要内容的提要。关键词是人们快速了解文档内容、把握主题的重要方式。关键词广泛应用于新闻报道、科技论文等领域，以方便人们高效地管理和检索文档。随着网络时代信息爆炸式的增长，关键词成为用户在海量信息中检索感兴趣内容的主要工具，诞生了如Google、百度等基于关键词的搜索引擎公司。在社会科学中，历史文档中关键词使用频度、内在含义等方面的变化也成为研究人类社会、文化和政治观念演变的重要途径[1–3]。

进入Web2.0时代，许多网站向用户提供了为感兴趣的对象(如链接、图片、视频、书籍和电影等)自由标注标签(tags)的功能，便于用户分享、管理、收藏和检索对象。与关键词类似，大部分标签是词或者短语，常常表示用户对标注对象主题的理解和概括。由于不同的用户都可以对同一个对象标注标签，具有较强的社会交互性，因此又称为社会标签(social tags)。用户标注的标签汇总起来成为一个体系，叫做大众分类法(Folksonomy)，这强调了与专家制定的分类体系(Taxonomy)之间的差异。广义上来讲，可以将社会标签看作Web2.0时代的关键词。

图 1.1给出了几个关键词示例。图 1.1(a)为美国计算机学会(ACM) ① 主办的某学术会议论文作者为其论文标注的关键词。从图中可以看到，论文作者提供了三种类型的关键词：第一种是“类别和主题描述(Categories and Subject Descriptions)”，这是作者从ACM提供的4层分类体系树中选择的，表示该论文的学科类别；第二种是“通用术语(General Terms)”，这是作者从ACM提供的含有16个词的词表中选择的，这些词汇无法归入上面的分类体系树，用来表示论文的类型；第三种是典型意义上的“关键词(Keywords)”，由论文作者自由指定，用来概括论文的主题。图 1.1(b)为著名技术网站MIT Technology Review中文版 ② 编辑为某篇新闻标注的关键词。由于该网站新闻一般是从英文翻译而来，为了方便读者理解，所标注的汉语关键词也大都同时显示其对应的英文。图 1.1(c)则是国

① http://portal.acm.org/

② http://www.mittrchinese.com/

ABSTRACT

This paper presents a new query recommendation method that generates recommended query list by mining large-scale user logs. Starting from the user logs of click-through data, we construct a bipartite network where the nodes on one side correspond to unique queries, on the other side to unique URLs. Inspired by the bipartite network based resource allocation method, we try to extract the hidden information from the Query-URL bipartite network. The recommended queries generated by the method are asymmetrical which means two related queries may have different strength to recommend each other. To evaluate the method, we use one week user logs from Chinese search engine Sogou. The method is not only 'content ignorant', but also can be easily implemented in a paralleled manner, which is feasible for commercial search engines to handle large scale user logs.

Categories and Subject Descriptors: H.3.3[Information Storage and Retrieval]: Information Search and Retrieval

General Terms: Algorithms, Experimentation.

Keywords: Asymmetrical query recommendation, user log analysis, network resource allocation, bipartite network.

（a）论文关键词

（b）新闻关键词

（c）社会标签

图 1.1　关键词示例：(a)为ACM会议论文关键词；(b)为著名技术网站MIT Technology Review中文新闻关键词；(c)为豆瓣网站上用户对电影《红高粱》标注的社会标签。

内影评网站豆瓣网 ① 用户对电影《红高粱》标注的社会标签，标签后面括号的数字表示对电影《红高粱》标注该标签的用户数量。

在传统科技论文库和新闻库中，人们一般请专家为待标注对象标注关键词。然而随着信息技术和社会的高速发展，每时每刻都有大量文档等信息产生。人工标注如此海量的信息已经变得不现实。社会迫切需求计算机能够为文档自动标注关键词。因此，**关键词自动标注**逐渐成为自然语言处理和信息检索的热点研究问题。目前，关键词自动标注技术已广泛应用于搜索引擎、新闻服务、电子图书馆等领域，在全文检索、文本分类、信息过滤和文档摘要等任务中发挥着重要作用。Web2.0时代的到来为关键词自动标注注入了新的活力。一方面，群体智慧(collective intelligence)产生了丰富的信息，如维基百科等知识库的建立，为关键词标注提供了更加丰富的知识和信息。另一方面，在社会标签服务中，为了方便用户标注标签，网站通常建立标签自动推荐系统为用户推荐标签，这也为关键词自动标注提供了广阔舞台。与此同时，海量信息处理也为关键词自动标注技术提出了新的挑战。

综上所述，关键词是信息时代人们管理、检索资源的重要手段和便捷工具，关键词自动标注技术是人们在海量信息中遨游的重要依赖，而标签推荐技术也与关键词标注有着重要联系。本文将着重研究海量信息中的高效关键词自动标注技术，并探索其在社会标签推荐中的应用。

为了更好地分析关键词自动标注所面临的挑战，接下来本文首先回顾和评述关键词自动标注和社会标签自动推荐的已有工作。

① http://www.douban.com

1.2　关键词自动标注的主要方式与算法

关键词自动标注主要有两种方式：关键词抽取(keyword extraction)与关键词分配(keyword assignment)[4]。关键词抽取，顾名思义，是从文档内容中寻找并推荐关键词；而关键词分配是从一个预先构建好的受控词表(controlled vocabulary)中推荐若干个词或者短语分配给文档作为关键词。

1.2.1　关键词抽取

目前研究者主要在新闻、学术论文、网页等文体上研究关键词抽取技术。关键词抽取一般分为两步：选取候选关键词和从候选集合中推荐关键词。

1.2.1.1　选取候选关键词

关键词一般是单个词或者由多个单词组成的短语。从文档中选取候选关键词的难点在于如何正确判定候选关键词的边界。寻找正确的短语在多种任务中都会涉及到，目前在英文关键词抽取中，一般选取N元词串(N-gram，N一般为1到3)，然后通过计算N元词串内部联系的紧密程度来判断它是否是一个有独立语义的短语。该任务与搭配抽取(collocation extraction)和多词表达(multi-word expression)抽取任务类似，都需要准确地判断边界。

搭配抽取曾尝试多种方式度量内部紧密程度[5]，如均值与方差(mean and variance)[6]，t测试(t-test)[7]，卡方测试(χ^2 test)，点互信息(point-wise mutual information)[8]和二项式似然比测试(binomial likelihood ratio test，BLRT)[9]等。Tomokiyo和Hurst[10]提出利用语言模型度量词串内部联系的紧密程度。Silva和Lopes[11]提出使用多词表达抽取技术提取候选关键词。

Hulth[12]则发现大部分关键词是名词性词组，符合一定的词性模式，如“形容词+名词”是最常见的模式。因此可以选取符合某种词性模式的词组作为候选关键词。就调研文献所知，大部分英文关键词抽取研究利用以上两种方式之一选取候选关键词。

1.2.1.2　推荐关键词

在得到文档的候选关键词集合后，研究者提出两种途径解决关键词选取问题。一种途径是无监督的方法，利用候选关键词的统计性质，如term frequency‐inverse document frequency(TFIDF)[13]等，对他们排序，选取最高的若干个作为关键词。另外一种途径是有监督的方法[4,14]，将关键词抽取问题转换为判断每

个候选关键词是否为关键词的二分类问题，它需要一个已经标注关键词的文档集合训练分类模型。无监督方法和有监督方法各有其优势和缺点：无监督方法不需要人工标注训练集合的过程，因此更加快捷，但由于无法有效综合利用多种信息对候选关键词排序，所以效果无法与有监督方法媲美；而有监督方法可以通过训练学习调节多种信息对于判断关键词的影响程度，因此效果更优，但在信息爆炸的网络时代，标注训练集合非常耗时耗力；更何况文档主题往往随着时间变化剧烈，随时进行训练集合标注更不现实。因此，最近关键词抽取研究主要集中在无监督方法方面。

在信息检索领域，Google的两位创始人Page和Brin提出了根据网页链接关系对网页进行排序的PageRank算法[15]，其基本思想是一个网页的重要性由链向它的其他网页重要性来决定，也就是说如果越多重要的网页指向某网页，那么该网页也就相应越重要。PageRank算法一度成为Google用来度量网页重要性的重要依据。

受到PageRank在信息检索领域巨大成功的启发，PageRank算法也在自然语言处理领域逐渐受到重视。2004年，Mihalcea和Tarau[16]提出一种基于图的排序算法TextRank，用以进行关键词抽取和文档摘要。该方法的基本思想是将文档看作一个词的网络，该网络中的链接表示词与词之间的语义关系。基于与PageRank相似的思想，TextRank认为一个词的重要性由链向它的其他词的重要性来决定，利用PageRank计算网络中词的重要性，然后根据候选关键词的PageRank值进行排序，从而选择排名最高的若干个词作为关键词。Litvak和Last[17]将同样最初用于网页排序的HITS算法[18]用于候选关键词排序，在关键词抽取性能上HITS算法与TextRank表现相似。

TextRank等基于图的关键词抽取算法取得了较以往有监督方法更优的效果，因此引起了研究领域的巨大兴趣。基于图的算法成为无监督关键词抽取的主流方法。许多研究者提出了各种方法引入外界知识帮助关键词排序，如考虑文档近邻[19,20]、与文档摘要互相增强补充[21,22]、考虑文档标题的作用[23]，等等。此外，通过词汇链(lexical chain)[24]、复杂网络统计性质[25]等进行关键词抽取，都以文档的词网作为基础。

1.2.2　关键词分配

受控词表一般收录某个或者某些领域的专业词汇。相比关键词抽取，关键词分配为每个文档推荐的关键词不一定在正文中出现。如果将受控词表中的关键词看作分类标签，那么关键词分配算法在本质是多类别多标签分类(multi-class

multi-label classification)[26,27]问题，也就是将一个文档打上若干个分类标签。此外，也有研究者着重研究如何利用受控词表提供的信息帮助关键词抽取[28–30]。

在研究中，常用的受控词表包括Open Directory Project (ODP)① 提供的多层次树状分类体系[31]，维基百科词条[32,33]等。关键词分配的性能主要受到两个方面的影响：

1. 受控词表的构建问题。受控词表的大小和覆盖度决定了关键词分配推荐关键词的范围和效果，如果受控词表无法覆盖待标注对象的主题，则会极大影响关键词分配的效果。
2. 关键词分配算法设计。实用关键词分配系统一般维护较大的受控词表。例如ODP含有超过100万分类标签，而英文维基百科词条数也高达350万(2010年2月27日获取的数据)。这带来的挑战是如何设计高效的关键词分配算法。

1.2.2.1 受控词表构建

受控词表可以看作与一个或者多个领域相关的专业词典，其中收录的词汇都是与某个领域相关的专门词汇。传统意义的词典都是由专家手工编纂的，这样做费时费力，已不能适用于网络时代的大规模应用。

目前一般利用网络信息自动构建词典，通过若干领域相关词汇作为种子，快速准确地获取领域新词。例如，Zheng等人[34]基于输入法用户数据从预先定义好的专业词典出发，自动扩充和发现领域相关的新词。Eisenstein等人[35]利用主题模型自动发现与地域相关的新词。Wang等人[36–38]提出了利用Web半结构化信息自动扩充命名实体(人名、地名、机构名)的方法。Pasca等人[39–42]提出从搜索引擎查询日志和网页等数据中抽取新词。邹刚、刘华等人[43,44]则分别提出通过分析大规模网页自动获取新词。为了具有较好的性能，受控词表一般包含几万甚至几十万词汇。

1.2.2.2 关键词分配算法

首先，当受控词表较大时，每个分类标签的的平均标注训练数据数量一般很少。在这种情况下为了保证关键词分配的效果，一般通过半监督学习(semi-supervised learning)方法[45]综合利用少量标注数据和大量无标注数据建立模型。

另一方面，传统分类算法如SVM等无法处理如此规模巨大的分类标签集合。

① http://www.dmoz.org/

Xue等人[31]在带层次的分类体系上先利用搜索策略缩小待分类标签的数量，然后再利用传统算法进行分类。Madani等人[46]提出了专门针对大量分类标签问题的分类算法，称为Feature Focus Algorithm (FFA)。FFA利用在线学习[47]思想计算文档中特征词与分类标签的关系，并限制每个特征词只记录其最相关的少数分类标签，从而得到较小的时间和空间复杂度，同时在多标签分类效果上优于以往方法。此外，在多标签分类研究领域，有学者开始研究标签之间的依赖关系对分类效果的影响[48]。

1.3 社会标签推荐

社会标签是目前大部分Web2.0网站(如社交网站、图片分享网站、产品评论网站等)不可缺少的一部分。为了方便用户标注标签，这些Web2.0网站往往为用户提供社会标签推荐功能。因此，社会标签自动推荐也成为自然语言处理和信息检索的研究热点。由于标签也是对标注对象主题的一种概括，因此从某种意义上讲，社会标签是传统关键词在Web2.0时代的化身；而社会标签推荐则可以看作是关键词自动标注在Web2.0时代的应用。按照利用的信息不同，社会标签推荐一般分为基于图的方法和基于内容的方法。

1.3.1 基于图的方法

基于图的方法中，所谓的图是根据用户标注历史建立的。在社会标签系统中，用户可以对感兴趣的对象标注他们自定义的标签，因此对于同一个对象会有多个用户标注不同标签。这种社会化的标注机制一般被称为协同标注。假如定义一次协同标注行为是一个用户对一个对象标注了一个标签，那么每次标注都会将一个用户、一个对象和一个标签构成一个三元组，它们互相之间建立了联系。而无数次协同标注行为就将不同的用户、对象和标签联系在了一起，构成“用户-对象-标签”三部分图(tri-partite graph)。

许多研究者根据这个图进行标签推荐，统称为基于图的方法。如Xu等人[49]采用推荐系统中常用的协同过滤[50]的思想进行标签推荐。Jaschke等人[51]提出FolkRank算法，采用类似PageRank的技术进行标签推荐。而Rendle等人[52]采用矩阵分解技术进行个性化标签推荐。这些方法的基本思想是，当一个用户准备标注一个对象，可以根据用户以往的标注行为以及该对象以往被标注的标签，为这个用户推荐“可能标注的标签”，也就是借助社群的喜好提供个性化的标签推荐服务。

基于图的方法对标注历史有较大的依赖，因此会存在以下几个问题：

1. 当一个用户或者对象新加入时，由于缺乏标注历史，系统很难有效推荐标签；
2. 同样的，当一个标签在历史上从来未标注过，而随着新事物涌现出来的时候，系统也很难有效地进行推荐。

这些问题可以统称为冷启动(cold-start)问题，也就是当缺乏这些用户、对象和标签的标注历史的时候，系统无法进行推荐。而其中新用户问题可以通过推荐最流行的标签解决；因此针对新对象和新标签的推荐在实际应用中问题更加突出。而基于内容的方法在新对象问题上是对基于图方法的有力补充。

1.3.2 基于内容的方法

在Web2.0标签服务中，大部分对象会包含文本描述信息。因此也有一部分研究者尝试进行基于内容的标签推荐。

很多研究是基于文档中的词作为特征进行标签推荐的。一部分研究者将社会标签看作分类标签，因此将社会标签推荐看做多标签文本分类问题[53–56]，但效果不尽如人意。ECML/PKDD Discovery Challenge 2008评测结果[57]显示，分类方法和其他方法相比，效果较差。此外，许多研究者提出采用K近邻法推荐社会标签，其基本思想是：给定一个对象，根据其内容找出与它最相似的K个对象，然后将这K个对象被标注的最流行标签推荐给原对象。K近邻法由于原理简单、具有较强鲁棒性，非常适合社会标签系统这样庞大而复杂的环境，因此在社会标签推荐中被广泛应用[58–62]。

除了以词作为特征进行推荐之外，利用隐含主题作为特征进行推荐也是最近的研究热点。隐含主题模型(latent topic model)[63–65]是机器学习领域对文档建模的主要方式。隐含主题模型认为文档可以表示为若干隐含主题的分布(topic distribution)，而一个隐含主题则表示为词上的分布(word distribution)。在社会标签推荐方面，许多研究者对文档和标签建立隐含主题模型，然后以隐含主题表示文档和标签并度量其相关程度，从而推荐与文档主题最相关的标签[66–69]。隐含主题模型可以很好地表示文档和标签主题，有效降低标签系统中噪音的影响。但是另外一个方面，隐含主题相对于词而言粒度较粗，对于具体实体(如人名、地名、机构名和产品名)的标签没有办法做到很好地区分，因此对这些细粒度标签推荐效果较差。

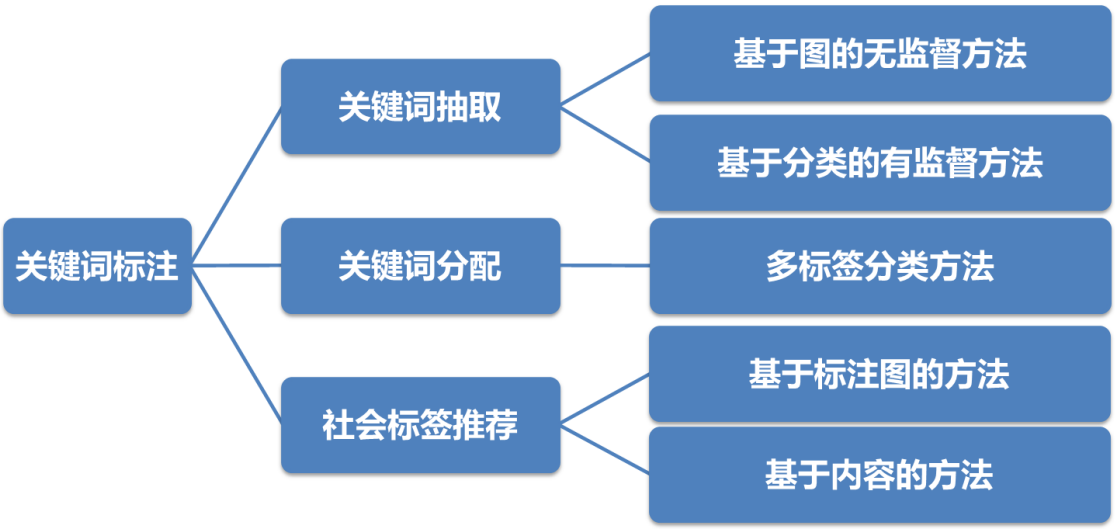


图 1.2　关键词标注的主要方式和方法总结。

1.4　关键词标注面临的挑战

以上介绍了关键词标注和社会标签推荐的已有工作。如果将社会标签推荐看作一种特殊的关键词标注方式，那么关键词标注的主要方式和方法可以参见图 1.2。

从对以上方法的介绍可以发现，无论是传统的关键词标注，还是新型的社会标签推荐，都没有系统地考察文档主题结构对关键词标注的作用。而一个文档往往会涉及多个主题(multiple topics/themes)，例如一篇关于关键词抽取的学术论文，可能既会涉及“关键词抽取”这个主题，也会涉及“图方法”这个主题。而关键词作为文档主题的摘要，应当具有较好的覆盖度。综合考虑，文档关键词通常需要同时具有以下三个特点：

1. 可读性(readability)。即关键词本身应该是有意义的词或者短语。例如，“关键词抽取”是一个有意义的短语，而“关键词抽”则不是。
2. 相关性(relevance)。既关键词必须与文档主题相关。例如，一篇主要介绍用图方法进行关键词抽取的学术论文，其中可能只顺带提到“文档摘要”这个短语，这时就不希望这个短语被选取作为文档关键词。
3. 覆盖度(coverage)。关键词要能够对文档的主题有较好的覆盖，不能只集中在文档某个主题而忽略了文档其他主题。

从以上三个特点，可以看到目前的关键词标注算法面临以下两个重要的挑战：关键词对文档主题覆盖度的问题，以及文档和主题之间的词汇差异问题。接下来将分别详细介绍这两个挑战。

1.4.1 文档主题的覆盖度问题

在传统关键词标注的方法中，以TextRank为代表的图方法的优势在于考虑文档中词与词之间的语义关系；以TFIDF为代表的统计方法则仅仅考虑词的统计性质。但是TFIDF和TextRank等方法均没有考虑所抽取的关键词对文档主题的覆盖度问题，导致推荐的关键词往往集中在某一个大的主题中，而没有顾及文档的其他主题。

对于文档主题的建模，已经有很多工作。有研究者探索利用人工标注的知识库如WordNet[70]等对文档主题进行建模，如利用WordNet构建文档词汇链(lexical chain)来表示文档主题进行文档摘要等[71,72]。但是WordNet等人工标注知识库中收录的词汇往往有限，无法满足网络时代日新月异的变化。也有研究者提出隐含主题模型[63–65]，在大规模文档集合中学习得到若干隐含主题，然后就利用这些隐含主题来表示文档的主题结构信息。

然而，在关键词标注中，如何有效利用文档主题结构信息，提高关键词对文档的主题覆盖度，提高关键词标注的性能，还没有被系统地探索研究过。

1.4.2 文档与关键词的词汇差异问题

在关键词标注中，关键词与文档的相关性是推荐关键词的重要指标。传统的方法如TFIDF仅依靠候选关键词在文档中的统计性质进行排序，而TextRank虽然在一定程度上考虑了文档中词与词之间的关系，但仍然倾向于选择文档中出现较为频繁的词作为关键词。而文档的关键词与文档往往存在一定的词汇差异现象，主要表现在两个方面：

1. 很多关键词在文档中的统计特性并不显著，也就是说文档的某些关键词本身并不一定在文档中频繁出现。
2. 在某些情况下，如文档较短的时候，一些关键词甚至并不出现在文档中。

那么，应当如何应对这一挑战呢？本文提出，文档和关键词都是对同一个事物的描述，因此他们具有“主题一致性”，基于这样一个假设，可以通过某些算法来建立文档中的词与关键词之间的语义关系，在文档和关键词之间建立语义映射关系，从而能够推荐语义相关的关键词。

针对以上这两个挑战，本文将以关键词抽取作为主要研究对象，探索如何有效利用文档主题结构进行关键词抽取。需要强调的是，虽然本文主要是在关键词抽取这个任务上进行相关研究，这些研究成果在一定程度上可以应用于关键词标注的其他任务上。

1.5　本文的主要工作内容

针对如何更好地构建文档主题，并利用文档主题提高关键词抽取性能的科学问题，本文从提高关键词对文档主题的覆盖度问题，以及解决文档与关键词的词汇差异问题入手，进行了以下几方面工作：

1. 基于文档内部信息，提出利用文档的词聚类算法构建文档主题，进行关键词抽取。该方法仅利用文档内部信息，通过度量文档中词与词之间的相似度，利用聚类的方法构建文档主题，并根据不同主题在文档中的重要性，进行关键词抽取。实验证明，该方法能够在一定程度上发现文档主要话题，并抽取出与文档主题相关的关键词，在一定程度上提高了关键词对文档主题的覆盖度。
2. 基于文档外部信息，提出利用隐含主题模型构建文档主题，进行关键词抽取。针对基于文档内部信息通过聚类算法进行关键词抽取受限于文档提供信息不足的缺点，提出利用机器学习算法中广泛使用的隐含主题模型构建文档主题，进行关键词抽取。并针对隐含主题模型训练速度较慢的瓶颈，提出了一种高效的并行隐含主题模型。实验证明，该方法能够更好地构建文档主题，并有效抽取关键词。
3. 提出综合利用隐含主题模型和文档结构信息，进行关键词抽取。针对隐含主题模型无法考虑文档结构信息的缺点，提出综合利用隐含主题模型和文档结构信息进行关键词抽取，即基于主题的随机游走模型。该方法一方面能够通过隐含主题模型构建文档主题，同时能够通过文档图的随机游走模型考虑文档结构为关键词抽取提供信息，实验证明，该方法能够综合隐含主题模型和文档结构信息进行关键词抽取的优势，有效抽取关键词。
4. 基于文档与关键词主题一致性的前提，提出基于机器翻译模型的关键词抽取方法。针对文档和关键词之间存在较大词汇差异的问题，基于文档和关键词主题一致性的前提，提出利用机器翻译中的词对齐模型计算文档中的词到关键词的翻译概率，然后进行关键词抽取。实验证明该方法能够有效的建立文档词汇与关键词之间的语义联系，有效地推荐关键词。
5. 根据以上基于文档主题结构的关键词抽取研究成果，选择新浪微博作为平台设计并实现了一个微博关键词抽取原型系统，在实际应用中验证了本文研究的有效性。

最后，第7章对关键词标注未来可能的研究方向进行展望。

第 2 章　基于文档内部信息构建主题的关键词抽取方法 ①

文档主题结构最直观的构建方法是基于文档内部信息。一篇文档往往包含多个主题，每个主题都在文档中对应不同的词语。例如一篇描写“北京”的文档，会从“位置”、“气候”和“文化”等方面进行描述。在词袋模型(bag of words)的假设下，除了没有表意功能的功能词(function words)外，每个词都被用来描述其中一个主题。根据这些主题，词语可以被分为若干个聚类，在同一个聚类中的词语在语义上更加相似。例如，“温度”、“寒冷”和“冬天”等词语是在主题“气候”下的词语，他们之间的语义相似度要大于他们与其他主题的词语的相似度。

基于这个直观的观察，本章提出基于文档词汇聚类构建文档主题进行关键词抽取的方法，提高抽取关键词对文档主题的覆盖度。基于词聚类的关键词抽取主要包括以下几个步骤:

1. 候选词选取。首先，需要将停用词去掉，为关键词抽取选取合适的候选词。
2. 计算候选词之间的语义相似度。
3. 根据语义相似度对候选词进行聚类。
4. 选取每个聚类中心词，在文档中选取合适的关键词。

下面将对每个步骤进行详细介绍。

2.1　候选词选取

并不是所有的词都有可能成为关键词。为了消除噪音的影响，本文利用一些启发式的方法选取能够成为关键词的候选词。选取过程如下:

1. 如果是英语，对文本进行断词(tokenization)；如果是汉语等没有单词分隔标志的语言，则首先对文本进行分词。
2. 去掉停用词，选取剩下的单词作为候选词。

在很多研究中[4,73]，候选关键词是通过N-gram发现的。而本章只是将单词作为关键词的候选词。到聚类后发现聚类中心词后，再将单个候选词扩展为可能含多个词的短语。

① 本章主要内容以“Clustering to Find Exemplar Terms for Keyphrase Extraction”为题作为口头报告论文发表在2009年的国际学术会议“The Conference on Empirical Methods in Natural Language Processing (EMNLP’09)”上。

2.2　词汇语义相似度

有两种方式来度量文档中词与词之间的相似度。一种是基于文档内的词同现(co-occurrences)关系，另外一种则利用外部知识库。

2.2.1　基于文档内同现关系的相似度

在文档中，词与词如果在短距离内同时出现过多次，说明他们具有较强的语义关系。因此可以利用文档内的同现情况来度量词与词之间的相似度。在这里，词与词的同现关系简单地表示为两个词在一个最多为w个词的滑动窗口内同现的次数。在本章，窗口大小w一般设为2到10之间的数值。

在计算同现相似度时，每个文档首先被转换为词的序列。这里有两种可能的转换词序列的方法。一种是利用文档中的每个词，没有做任何的过滤。另外一种，则是过滤了其中停用词等没有实际意义的词。这里选择第一种，原因是，虽然停用词本身不具备成为关键词的可能、也无需计算相似度，但是他们的存在能够帮助提供距离信息来判断两个词是否具有较高的相关度。例如，两个词如果中间没有任何其他词，他们的相关度要高于中间有若干词间隔的两个词。

2.2.2　基于维基百科的相似度

有很多研究者探索了利用外部知识库度量词与词之间相似度的方法。受到Gabrilovich和Markovitch的Explicit Semantic Analysis(ESA)方法[74]的启发，这里利用维基百科来度量词与词之间的相似度。维基百科是目前最大的在线百科全书，目前已收录1,800万篇百科词条，其中英文词条超过350万条①。

利用维基百科计算词汇相似度的基本思想是：将每个维基百科词条看作是一个独立的概念(concept)。这样，一个词的语义信息就可以用维基百科概念上的分布来表示，其中在某个概念上的权重(weight)可以用这个词在该概念词条中的TFIDF值来表示。这样就可以通过比较两个词的概念向量来度量他们的相似度。实验证明[74]该方法作为相似度度量非常有效。

这里选择余弦相似度(cosine similarity, COS)、欧氏距离(Euclid distance, EUC)、点互信息(point-wise mutual information, PMI)和规范化Google距离(normalized Google distance, NGD)来计算相似度。假如某个词t_i的维基百科概念向量为$C_i = \{c_{i1}, c_{i2}, ..., c_{iN}\}$，其中$N$表示维基百科词条数目，$c_{ik}$表示$w_i$在第$k$个维基

① 该数据获取自维基百科中关于“Wikipedia”的词条 http://en.wikipedia.org/wiki/Wikipedia（获取时间: 2011年3月12日18点55分）。

百科概念中的TFIDF权重。那么，余弦相似度表示为:

$$cos(i, j) = \frac{C_i \cdot C_j}{||C_i||||C_j||} \tag{2-1}$$

欧氏距离相似度表示:

$$euc(i, j) = \sqrt{\sum_{k=1}^{N}(c_{ik} - c_{jk})^2} \tag{2-2}$$

点互信息是一种常用的度量相似度的方法。这里采用三种办法来用PMI度量词与词之间的相似度。一种是利用维基百科的词条数目：

$$pmi_p(i, j) = \log_2 \frac{N \times p(i, j)}{p(i) \times p(j)} \tag{2-3}$$

其中$p(i, j)$是同时包含t_i和t_j的维基百科词条，$p(i)$和$p(j)$分别表示包含t_i和t_j的词条数目。第二种是利用维基百科的词数目：

$$pmi_t(i, j) = \log_2 \frac{T \times t(i, j)}{t(i) \times t(j)} \tag{2-4}$$

其中T表示维基百科中词的数目，$t(i, j)$表示t_i和t_j在维基百科中相邻出现的次数，而$t(i)$和$t(j)$分别表示t_i和t_j在维基百科中出现的次数。第三种则是对以上两种方法的融合:

$$pmi_c(i, j) = \log_2 \frac{N \times pw(i, j)}{p(i) \times p(j)} \tag{2-5}$$

其中$pw(i, j)$是出现t_i和t_j相邻情况的维基百科词条数目，很显然，$pmi_c(i, j) \leq pmi_p(i, j)$，而$pmi_c(i, j)$的要求更加严格，也更能准确地度量相似度。

规范化Google距离是一种度量词汇相似度的新型方法[75]。NGD的理论基础是信息距离(information distance)和科尔莫戈罗夫复杂度原理(Kolmogorov complexity)。NGD利用网络大规模数据中词的出现情况来度量相似度。本章在维基百科上利用NGD度量词汇相似度：

$$ngd(i, j) = \frac{\max(\log p(i), \log p(j)) - logp(i, j)}{\log N - \min(logp(i), logp(j))} \tag{2-6}$$

其中N是维基百科词条数目，这里作为一个规范化因子(normalized factor)。

2.3　聚类方法

聚类是典型的无监督机器学习问题，它的任务是将对象划分成不同的组，每个组内的对象互相比较相似，而组与组之间的对象比较不同[76]。这里选择三

种广泛使用的典型的聚类算法：层次聚类(hierarchical clustering)、谱聚类(spectral clustering)和信任传播聚类(Affinity Propagation)。本章将使用这些聚类方法将一个文档中的词按照它们之间的相似度进行聚类。下面分别介绍这三种算法的基本思想。

2.3.1　层次聚类

层次聚类将数据点按照不同的聚类粒度建立一个聚类层次树。这个树有多层，每一层由它的下一层聚类组成。层次聚类的算法流程如下：

1. 计算数据集合中每对数据点之间的相似度(或者距离)；
2. 不断将集合中距离最近的两个点组合成一个新的点，这样就形成一个多层的二叉树；
3. 决定在哪一层划分，得到相应的聚类结果。

从以上流程可以看到，层次聚类需要事先显式地指定聚类个数C。本章采用Matlab提供的层次聚类。这里虽然采用层次聚类算法，但是并不需要聚类所提供的层次信息。

2.3.2　谱聚类

近年来，谱聚类成为最流行的聚类算法之一。谱聚类利用数据相似矩阵的谱(spectrum)信息来进行特征降维，将数据点聚类到少数几个维度。由于谱聚类方法易于实现，性能优于传统聚类算法如k-means等，因此得到广泛使用。关于谱聚类的更具体的介绍可以参考[77]。

本章采用Chen等[78]提供的开源谱聚类工具①。由于基于同现的词汇相似度矩阵一般比较稀疏，谱聚类中的特征值分解可能会经常报错，因此，这里采用奇异值分解(Singular Value Decomposition，SVD)作为代替。对于谱聚类，有两个参数需要事先指定：

- 聚类个数C；
- 转换因子σ，用来根据数据点距离建立数据点相似度矩阵，如下所示

$$s(i,j)=\exp\left(\frac{-d(i,j)^2}{2\sigma^2}\right) \tag{2-7}$$

其中$s(i,j)$和$d(i,j)$分别表示数据点i和j之间的相似度和距离。

① http://www.cs.ucsb.edu/~wychen/sc.html。

2.3.3　信任传播聚类

另外一个著名聚类算法是信任传播聚类(Affinity Propagation，AP)。该算法是基于消息传递(message passing)技术的。AP最早由Frey等[79]提出，其论文报告指出AP能够得到比其他方法更优的结果。本章采用Frey等人开发的工具①。关于AP算法细节可以参考[79]，AP算法共有三个主要参数：

- 聚类偏好(preference) p。与k-means等传统算法需要事先指定聚类个数不同，AP算法不需要显式地指定聚类个数，但是需要为每一个数据点设定一个实数值p，作为他们成为聚类中心的偏好程度。这样，具有较大p的数据点就越有可能被选择成为聚类中心(exemplars)。一般而言，如果对数据点没有先验的偏好，可以将所有数据点的偏好值设为数据点间相似度$s(i,j), i \neq j$的最大值(maximum)、最小值(minimum)、平均值(mean)或者中位值(median)。
- 收敛判据(convergence criterion)。AP算法是一个迭代算法，它一旦满足以下两个条件之一，即停止：(1)如果每个数据点所属聚类的决定保持I_1次迭代不变；(2)算法迭代次数达到I_2次。这里设定$I_1 = 100$，$I_2 = 1,000$。
- 衰减因子(Damping factor) λ。当进行消息传递的时候，需要避免数值震荡(numerical oscillations)的发生，因此，每个消息设置为它上轮迭代时的数值与λ的积加上它将要更新的值与$1-\lambda$的积。λ取值范围是0到1，本章取值为$\lambda = 0.9$。

2.4　从聚类中心词扩展关键词

词聚类完成后，选取每个聚类的中心词作为种子词。在信任传播聚类中，算法本身会提供聚类中心；在层次聚类中，聚类中心词可以通过Matlab计算获得；而谱聚类选取距离聚类中心点最近的词作为聚类中心词。

根据Hulth的研究结论[80]，大部分手工标注的关键词是名词短语(noun phrase)。因此首先将文档进行词性标注，然后选取模式为0个或者多个形容词跟随1个或者多个名词的词串作为名词短语，将其作为候选关键词。本章在英文上进行实验，因此使用Stanford Log-Linear Tagger②进行词性标注。从这些名词短语中选取那些包含一个或者多个聚类中心词的作为文档的关键词。

在这个过程中，发现很多选取出来的关键词只有一个单词。但实际上只有很少的关键词是单词。因此，作为后处理，本章选用一个常用词列表，将那些频繁

① http://www.psi.toronto.edu/affinitypropagation/。
② http://nlp.stanford.edu/software/tagger.shtml

出现的词过滤掉。

2.5　实验结果与分析

由于汉语的词性标注效果还有待于进一步提高，因此这里只在英文上进行实验。实验选用来自*Inspec*的论文摘要作为待抽取关键词文档集合。该数据集合包含了人工标注的关键词，该数据是由Anette Hulth提供的，并在Hulth、Rada等人的工作[80,81]中使用过。每个摘要包含两种关键词：一种是受控关键词，是在一个给定的词典中选取；另外一种是非受控关键词，由专家自由标注。在这里，如Hulth、Rada等人的工作[80,81]那样，本章选用非受控关键词作为标准答案来评价各种方法的效果。另外，本章只考虑在文档出现的非受控关键词作为标准答案。在评价时，方法抽取的关键词和标准答案都进行stemming获取词干进行比较。

在Hulth的实验[80]中，对于有监督方法，Hulth将2,000个摘要划分为1,000个作为训练集合(training set)，500作为验证集合(validation set)，500个作为测试集合(test set)。而在Rada的实验[81]中，由于TextRank是无监督方法，因此仅在测试集合上与Hulth的方法进行了比较。本章也将在500个测试集合上与Hulth、TextRank进行比较。

在计算基于维基百科的相似度时，本章选用2005年11月11日的维基百科数据快照① 进行实验。后处理中的常用词表也是利用维基百科数据计算得到的，实验把在维基百科中出现超过1,000次的单词作为常用词。

实验采用准确率、召回率和F_1值(precision/recall/F_1-Measure)来评价关键词抽取的效果。

$$\begin{aligned} Precision &= \frac{c_{correct}}{c_{extract}} \\ Rrecall &= \frac{c_{correct}}{c_{standard}} \\ F_1 - Measure &= \frac{2pr}{p+r} \end{aligned} \tag{2-8}$$

其中$c_{correct}$是一个方法所有准确抽取的关键词数目，$c_{extract}$是所有抽取的关键词数目，而$c_{standard}$是所有人工标注的标准答案数目。

2.5.1　相似度度量方法对关键词抽取的影响

本章首先分析词汇相似度度量方法对关键词抽取的影响。在实验中发现基

① http://www.cs.technion.ac.il/~gabr/resources/code/wikiprep/

表 2.1　相似度度量方法对关键词抽取的影响。

参数	Precision	Recall	F_1-Measure
基于同现的相似度			
$w = 2$	0.331	0.626	0.433
$w = 4$	0.333	0.621	0.434
$w = 6$	0.331	0.630	0.434
$w = 8$	0.330	0.623	0.432
$w = 10$	0.333	0.632	0.436
基于维基百科的相似度			
cos	0.348	0.655	0.455
euc	0.344	0.634	0.446
pmi_p	0.344	0.621	0.443
pmi_t	0.344	0.619	0.442
pmi_c	0.350	0.660	**0.457**
ngd	0.343	0.620	0.442

于维基百科的相似度要优于基于同现关系的相似度，但优势并不十分明显。在表 2.1中列出了在谱聚类方法下，不同相似度度量方法的效果。表格中的w表示计算同现的滑动窗口长度。而cos，euc等表示不同的计算基于维基百科相似度的方法。

这里选取谱聚类的原因是它的效果要优于其他两个两个聚类方法。关于聚类方法对关键词抽取的影响将在下一部分介绍。表 2.1中的结果是在聚类个数$C = \frac{2}{3}n$的时候得到的，这里n表示该文档中的候选词个数。此外，对于欧氏距离、Google距离，设公式(2-7)中的$\sigma = 36$来进行距离和相似度的转换，这是在尝试$\sigma = 9, 18, 36, 54$等设置后得到的最优结果。

如表 2.1所示，虽然基于维基百科的相似度优于基于同现的相似度，它们之间的差距并不显著。而由于维基百科是利用维基百科上的统计信息得到的，因此它应当比基于文档内同现关系的相似度要更加精确，这也使关键词抽取的效果更好。但是另一个方面，基于维基百科的相似度没有考虑文档内部的信息。本章也尝试将两种相似度方法进行线性加权，但在关键词抽取效果上并没有提升。

通过以上比较，本章总结认为，虽然基于维基百科的相似度效果好于基于同现的相似度，但是由于维基百科上的相似度度量计算复杂度要比基于文档同现的方法高得多，因此基于文档内的同现相似度更加实用。

表 2.2　聚类方法对关键词抽取的影响。

参数	Precision	Recall	F_1-Measure
层次聚类			
$C = \frac{1}{4}n$	0.365	0.369	0.367
$C = \frac{1}{3}n$	0.365	0.369	0.367
$C = \frac{1}{2}n$	0.351	0.562	0.432
$C = \frac{2}{3}n$	0.346	0.629	0.446
$C = \frac{4}{5}n$	0.340	0.657	0.448
谱聚类			
$C = \frac{1}{4}n$	0.385	0.409	0.397
$C = \frac{1}{3}n$	0.374	0.497	0.427
$C = \frac{1}{2}n$	0.374	0.497	0.427
$C = \frac{2}{3}n$	0.350	0.660	**0.457**
$C = \frac{4}{5}n$	0.340	0.679	0.453
信任传播聚类			
$p = max$	0.331	0.688	0.447
$p = mean$	0.433	0.070	0.121
$p = median$	0.422	0.078	0.132
$p = min$	0.419	0.059	0.103

2.5.2　聚类方法对关键词抽取的影响

这里，将在基于维基百科的相似度pmi_c方法下分析聚类方法对关键词抽取的影响。上一部分已经证明该方法优于其他相似度度量。

在表 2.2中显示了三种聚类算法的关键词抽取效果。对于层次聚类和谱聚类，本章显式地将聚类个数C设置为候选词数的比例；而对信任传播聚类，本章将偏好值设置为数据点相似度$s(i, j)$的最大值、平均值、中位值和最小值，分别用max，$mean$，$median$和min表示。

如表中所示，在关键词抽取的性能上，谱聚类要优于层次聚类和信任传播聚类。而在这些方法中，只有信任传播聚类在某些参数上的效果很差。

2.5.3　与其他方法比较

在表 2.3中列出了聚类方法与其他方法[80,81]在相同数据集合上最好效果的比较。对于每个方法，除了准确率、召回率和F_1之外，还列出了推荐的关键词总数、平均每篇文档推荐的关键词数，以及准确推荐的关键词总数和平均每篇文档准确推荐的关键词数。在表格中，层次聚类、谱聚类和信任传播聚类分别用“HC”，“SC”和“AP”表示.

表 2.3　与Hulth方法、TextRank等方法的比较。

	抽取		正确				
方法	总计	平均	总计	平均	Precision	Recall	F_1-Measure
Hulth's	7,815	15.6	1,973	3.9	0.252	0.517	0.339
TextRank	6,784	13.7	2,116	4.2	0.312	0.431	0.362
HC	7,303	14.6	2,494	5.0	0.342	0.657	0.449
SC	7,158	14.3	2,505	5.0	**0.350**	0.660	**0.457**
AP	8,013	16.0	2,648	5.3	0.330	**0.697**	0.448

Hulth的结果是在[80]中在相同文档集合上报告的最好结果；这是一个基于分类的有监督方法，它的优点是对候选关键词利用了更多的语言学特征。它的做法是，通过N-gram选取候选关键词，将候选关键词的词性标注信息作为特征学习关键词分类器。

TextRank的结果是在[81]中在相同文档集合上报告的最好结果。TextRank首先根据文档利用长度为$w = 2$的滑动窗口建立词汇同现关系图，然后在图上运行PageRank算法，利用词的PageRank值对候选关键词进行排序。

在这个表格中，层次聚类的最优结果是在聚类个数设为$C = \frac{2}{3}n$、在维基百科上使用欧氏距离得到的；谱聚类的参数如上一节所示；而信任传播聚类的最优结果是在偏好值设为$p = max$、在维基百科上使用欧氏距离得到的。

从这个表格可以观察到，基于聚类的方法要优于TextRank和Hulth的方法，对于谱聚类，F_1值甚至要超过TextRank和Hulth方法约9.5%。

此外，由于基于聚类的方法是无监督的，因此不需要任何训练集合。本章也在Hulth提供的2,000篇摘要上进行实验，评价效果与在500篇上类似。在2,000篇上，最优结果的参数是：使用谱聚类、设置聚类个数$C = \frac{2}{3}n$，并使用基于维基百科的pmi_c进行相似度度量，这与在500篇上的设置相同。实验总共抽取了2,9517的关键词，其中9,655被准确抽取出来。准确率、召回率和F_1值分别为0.327，0.653和0.436。这说明基于聚类的关键词抽取方法的稳定性。

2.5.4　分析与讨论

下面将针对这篇EMNLP2009论文的摘要进行关键词抽取，观察基于聚类方法的效果。摘要如表 2.4所示。利用谱聚类算法和基于维基百科pmi_c相似度进行关键词抽取的结果如图 2.1。图中的关键词是抽取词干(stemming)后的结果。从该表发现当$C = \frac{1}{4}n, \frac{1}{3}n, \frac{1}{2}n$时抽取的关键词是相同的，其中$C = \frac{1}{3}n$时的聚类中心词(exemplar terms)用加粗字体显示。可以看到“unsupervised”，“exemplar term”

表 2.4　EMNLP2009论文摘要。

Keyphrases are widely used as a brief summary of documents. Since manual assignment is time-consuming, various unsupervised ranking methods based on importance scores are proposed for keyphrase extraction. In practice, the keyphrases of a document should not only be statistically important in the document, but also have a good coverage of the document. Based on this observation, we propose an unsupervised method for keyphrase extraction. Firstly, the method find exemplar terms by leveraging clustering techniques, which guarantees the document to be semantically covered by these exemplar terms. Then the keyphrases are extracted from the document using the exemplar terms. Experiments demonstrate that the method outperforms the state-of-the-art graph-based ranking methods on precision, recall and F_1-Measure.

图 2.1　对EMNLP2009论文摘要的抽取关键词结果示例。

$C = \frac{1}{4}n, \frac{1}{3}n, \frac{1}{2}n$时抽取的关键词
unsupervis method; various **unsupervis** rank method; **exemplar term**; state-of-the-art **graph-bas** rank method; **keyphras**; **keyphras** extract
$C = \frac{2}{3}n$时抽取的关键词
unsupervis method; manual assign; brief summari; various unsupervis rank method; exemplar term; document; state-of-the-art graph-bas rank method; experi; keyphras; import score; keyphras extract

和“keyphrase extraction”都被正确抽取了出来。事实上，“clustering technique”也应当被作为关键词抽取出来，但是，由于“clustering”因以-ing结尾而被错误标注为动词，这与关键词的名词短语模式不匹配，因此这个关键词没有被抽取为关键词。

当$C = \frac{2}{3}n$，抽取的关键词包含了更多的噪音，例如夹杂了很多单词。这是由于聚类个数的增加，更多的聚类中心词被识别出来，因此更多关键词会被聚类中心词抽取出来，如果设置$C = n$，所有的词就都会被作为聚类中心词。在这种极端情况下，所有的名词性短语都会被抽取出来作为关键词，这显然是有问题的。因此，对于基于聚类的方法最重要的就是如何更好地判断聚类个数。

实验发现常用词表对关键词抽取的效果非常重要。如果没有经过常用词表过滤，最好的抽取效果将会下降大约5个百分点，到40%。但是，目前这种处理噪音的办法仍然过于简单粗暴。

2.6　本章小结

本章提出一种聚类的方法构建文档主题结构，并以此为基础进行无监督的关键词抽取。本方法首先将候选词组成若干个聚类，然后选取每个聚类的聚类中心词。然后，再用这些聚类中心词从文档中抽取名词短语作为关键词。基于聚类的关键词抽取方法能够更好地保证抽取的关键词对文档主题的覆盖度。

但是，也从实验中也可以看到，基于聚类的关键词抽取方法的效果会较大地受到聚类个数的影响。然而，如何确定聚类个数本身是聚类方法的重要研究问题，至今研究领域提出了许多方法，但都没有一个很好的广泛适用的解决方案。本章采用常用词表的方法来过滤噪音作为对聚类方法的补充。因此，基于聚类的方法虽然可以在一定程度上构建文档主题，提高抽取关键词对文档主题的覆盖度，但仍然有比较大的局限，这很大程度上是由于仅对文档内部的词语进行聚类造成的限制。下章将提出一种基于外部大规模文档集合构建文档主题的方法进行关键词抽取。

第 3 章　基于隐含主题模型构建主题的关键词抽取方法 ①

上一章介绍了通过词聚类的方法构建文档主题进行关键词抽取的方法。该方法利用了文档内部信息，也就是词的聚合度来发现文档主题。但该方法的不足在于：一方面，一篇文档的信息有限，往往无法为发现文档主题提供足够的信息；另一方面，该方法会受到词汇相似度度量和聚类方法性能的较大影响，而目前，如何为聚类算法找到合适的聚类个数，仍然是一个困难的研究问题。以上两个方面的不足，造成基于聚类的关键词抽取方法性能并不稳定。本章提出基于隐含主题模型构建主题进行关键词抽取的方法。该方法的特点在于：它使用大规模文档集合学习隐含主题，这避免了一篇文档自身信息不足的问题，同时也能够得到比较有意义的、稳定的主题信息，避免了在一篇文档上聚类的不确定性。

3.1　隐含主题模型及其加速算法

3.1.1　隐含主题模型

隐含狄利克雷分配模型(latent Dirichlet allocation, LDA)最早由Blei、Ng和Jordan于2003年提出[65]，用来对文档建模。在LDA中，每篇文档被表示为K个隐含主题的混合分布(mixture)，而每个主题k，则表示为在W个词上的多项分布$\boldsymbol{\phi}_k$。对任何一个文档d_j，它的主题分布$\boldsymbol{\theta}_j$产生于一个狄利克雷先验(Dirichlet prior)，该先验参数为α。对文档d_j中的第i个词x_{ij}，它的主题$z_{ij} = k$产生于$\boldsymbol{\theta}_j$，而词x_{ij}则产生于$\boldsymbol{\phi}_k$。LDA的产生过程因此可以表示为:

$$\boldsymbol{\theta}_j \sim Dir(\alpha), \boldsymbol{\phi}_k \sim Dir(\beta), z_{ij} = k \sim \boldsymbol{\theta}_j, x_{ij} \sim \boldsymbol{\phi}_k, \tag{3-1}$$

其中$Dir(*)$表示狄利克雷分布。LDA的概率图表示可见Fig. 3.1，其中可见变量，即文档中的词x_{ij}以及超参数(hyper-parameters)α和β用阴影表示。

Griffiths与Steyvers[82]提出利用吉布斯采样(Gibbs Sampling)的方法学习LDA模型。吉布斯采样是一种典型的蒙特卡洛马尔可夫链方法(Markov-chain Monte Carlo，MCMC)。MCMC广泛应用于隐含主题模型的学习，例如Author-Topic Mode[83]，Pachinko Allocation[84]，以及 Special Words with Background Model[85]。更重

① 本章主要内容以“PLDA+: Parallel Latent Dirichlet Allocation with Data Placement and Pipeline Processing”为题作为学术论文发表在“ACM Transactions on Intelligent Systems and Technology (ACM TIST): Special Issue: Large Scale Machine Learning and Applications”上。

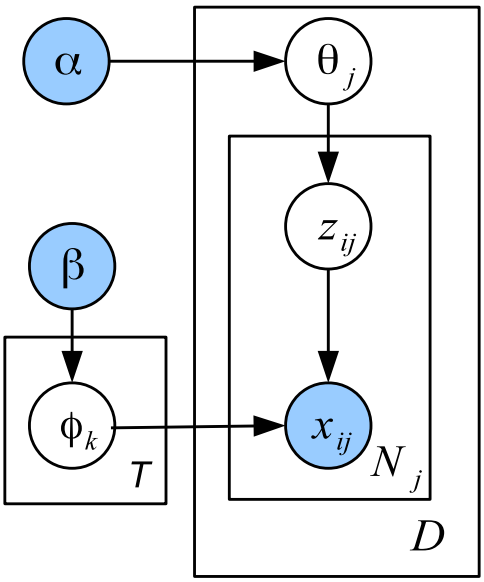


图 3.1　隐含主题模型LDA的概率图表示。

要的是，由于变分方法相对而言需要使用更多的内存，因此吉布斯采样的应用更加广泛[86]，因此本文也将主要介绍利用吉布斯采样学习LDA的方法。

在吉布斯采样中，一般会利用狄利克雷分布和多项分布之间的对偶性质(Dirichlet-multinomial conjugacy)将变量$\boldsymbol{\theta}$和$\boldsymbol{\phi}$积分掉，只对隐含变量$\boldsymbol{z}$进行采样。这种采样方法称为**collapsed**吉布斯采样(Collapsed Gibbs Sampling)[87]。在LDA的吉布斯采样中，需要同时维护两个矩阵：(1)“词-主题”同现次数矩阵C^{word}，其中每个元素C_{wk}代表在训练语料库中词w被赋给主题k的次数；(2)“文档-主题”同现次数矩阵C^{doc}，其中每个元素C_{kj}表示文档d_j中的词被赋给主题k的次数。此外，为了计算方便，还要维护一个记录主题次数的向量C^{topic}，其中每个元素C_k表示在训练语料库中主题k出现的次数。

假如现在要对w_{ij}计算其主题z_{ij}，给定训练语料库中除w_{ij}外其他的词以及它们上一轮采样得到的主题，可以得到z_{ij}的条件概率如下：

$$\Pr(z_{ij}=k|\boldsymbol{z}^{\neg ij},\boldsymbol{x}^{\neg ij},x_{ij}=w,\alpha,\beta)\propto\frac{C_{wk}^{\neg ij}+\beta}{C_{k}^{\neg ij}+W\beta}\left(C_{kj}^{\neg ij}+\alpha\right) \tag{3-2}$$

其中$\neg ij$表示对应的词w_{ij}及其上一轮采样的主题被排除在外。每次通过公式(3-2)采样得到w_{ij}的新主题z_{ij}后，矩阵C^{word}，C^{doc}和向量C^{topic}都会进行相应更新。当经过足够多次采样的迭代后，可以根据矩阵C^{word}，C^{doc}和向量C^{topic}计

算LDA的两个参数$\boldsymbol{\theta}$和$\boldsymbol{\phi}$:

$$\boldsymbol{\theta}_{kj} = \frac{C_{kj} + \alpha}{\sum_{k=1}^{K} C_{kj} + K\alpha} \qquad \boldsymbol{\phi}_{wk} = \frac{C_{wk} + \beta}{\sum_{w=1}^{W} C_{wk} + W\beta} \tag{3-3}$$

其中$\boldsymbol{\theta}_{kj}$表示文档d_j中主题k的概率，而$\boldsymbol{\phi}_{wk}$表示词w上主题k的概率。

3.1.2　隐含主题模型的加速算法

LDA的计算复杂度较高，因此在大规模训练语料库上的学习会遇到计算瓶颈。因此，有很多研究者对LDA提出了各种加速算法。

Mimno与McCallum[88]提出一种Dirichlet Compound Multinomial LDA (DCM-LDA)算法。DCM将训练语料库划分为多个部分，分发到分布式机器上，然后在每台机器上各自进行吉布斯采样，每台机器各自维护一个局部主题模型(local topic model)。DCM在学习过程中不同机器之间没有任何通信，当每台机器完成学习后，会通过聚类的方式将不同机器上学习得到的主题模型融合在一起。

Newman等[89]提出Approximate Distributed LDA (AD-LDA)算法。类似于DCM，在AD-LDA算法中每一台分布式的机器对训练语料库的一部分进行吉布斯采样。但是与DCM不同，AD-LDA在所有机器完成每次迭代后，都会进行全局同步(synchronization)主题模型，即通过reduce-scatter操作将不同机器上的局部主题模型(local topic model)融合为一个统一的模型(global topic model)，然后分发到不同的机器上。相对于LDA而言，AD-LDA只在每次迭代后更新机器上的主题模型，因此被称为一种近似(approximate)算法。

Asuncion等[90]提出一种完全异步(asynchronous)的分布式LDA算法(AS-LDA)。与AD-LDA不同，AS-LDA不再进行全局同步，而是当每台机器完成一次局部吉布斯采样迭代后，随机地寻找其他也刚完成一次迭代的机器，互相交换主题模型，进行两个局部模型的融合。

此外，Yan等[91]提出一种基于图形处理器(Graphic Processor Unit, GPU)的LDA并行算法。GPU的特点是内置了并行处理器，处理器间共享内存，因此避免了AD-LDA和AS-LDA中的机器间通信问题。

除了对LDA的并行化外，还有研究者直接对LDA学习过程进行优化。例如，Gomes等[92]提出一种改进的变分学习方法，能够在内存受限的情形下高效地学习LDA主题模型。此外，Porteous[93]提出一种加速公式(3-2)的方法，该方法利用了文档d_j的主题概率$\boldsymbol{\theta}_j$一般比较稀疏的性质，是一种非近似的方法。

3.1.3　PLDA: 基于MPI的AD-LDA实现

此前已经实现了一个MPI的AD-LDA[89] — PLDA[94]。PLDA现在已经得到较为广泛的应用，如社区推荐[95]等。

AD-LDA将D个训练文档分配到P台机器上，平均每台机器需要维护$D_p = D/P$篇文档。也就是说AD-LDA将文档内容$\boldsymbol{x} = \{\boldsymbol{x}_d\}_{d=1}^{D}$划分为$\{\boldsymbol{x}_{|1}, \ldots, \boldsymbol{x}_{|P}\}$，并将每个词分配的主题信息$\boldsymbol{z} = \{\boldsymbol{z}_d\}_{d=1}^{D}$划分为$\{\boldsymbol{z}_{|1}, \ldots, \boldsymbol{z}_{|P}\}$，其中$\boldsymbol{x}_{|p}$和$\boldsymbol{z}_{|p}$仅在机器$p$上出现。类似地，文档-主题矩阵$C^{doc}$也与对应文档一起分配到$P$台机器上，这里可以将每台机器$p$上的文档-主题矩阵表示为$C_{|p}^{doc}$。每台机器还维护一份本地的词项-主题矩阵$C^{word}$。因此这里用$C_{|p}^{word}$存储在机器$p$上进行本地文档中单词进行主题分配的结果。

在每次吉布斯采样迭代中，每台服务器p对每个词上的主题$z_{ij|p} \in \boldsymbol{z}_{|p}$进行抽样并更新$\boldsymbol{z}_{|p}$。抽样概率是近似的后验概率(approximate posterior distribution):

$$\Pr(z_{ij|p} = k \mid \boldsymbol{z}^{\neg ij}, \boldsymbol{x}^{\neg ij}, x_{ij|p} = w) \propto \frac{C_{wk}^{\neg ij} + \beta}{C_{k}^{\neg ij} + W\beta}\left(C_{kj|p}^{\neg ij} + \alpha\right) \tag{3-4}$$

更新$\boldsymbol{z}_{|p}$的同时，还会根据新分配的主题更新$C_{|p}^{doc}$和$C_{|p}^{word}$。每次迭代后，每台机器重新计算各自的词项-主题矩阵$C_{|p}^{word}$，然后利用AllReduce操作将各台机器上的词项-主题矩阵进行融合，形成最新的C^{word}并分发给各台机器进行下一轮吉布斯采样迭代。更具体的实现细节可以参考[94]。

此前，也曾利用MapReduce[96,97]实现过AD-LDA，如论文[94]所述。利用MapReduce，许多计算可以通过三个基本过程来实现: mapping，shuffling和reducing。可以采用MapReduce框架来实现AD-LDA中的重要操作AllReduce。但是在MapReduce版本的AD-LDA中，每次吉布斯采样迭代的前后，都需要进行磁盘(disk)读写操作来获取和更新词项-主题矩阵。此外，每台机器的本地数据必须被重新写到硬盘上，来供下一次迭代使用。这样做虽然提高了MapReduce的容错性，但是也使之不能适用于LDA这种迭代算法。而对于MPI，可以通过简单设置检查点(check point)来实现容错功能。由于MapReduce的这种机制，导致相同数据的两次相邻吉布斯采样不一定能够在同一台机器上完成，因此每次迭代开始都需要重新从硬盘读入数据。因此，论文[94]实验已经证明相比MapReduce，MPI更适合实现LDA的加速算法。

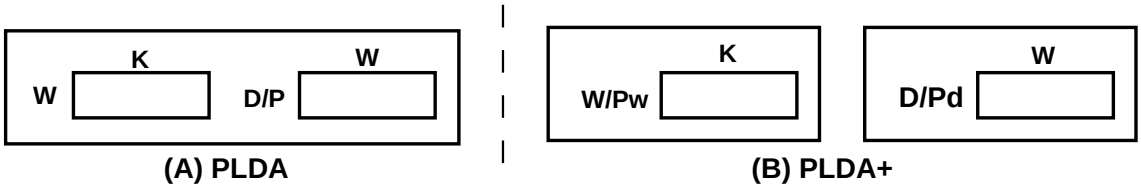


图 3.2　PLDA和PLDA+中文档和词项-主题矩阵的分配策略。

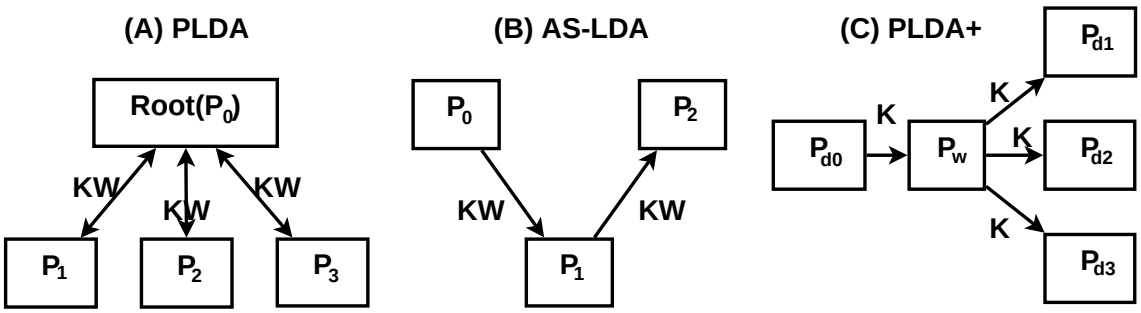


图 3.3　PLDA、AS-LDA和PLDA+中某个词在某台机器上主题更新信息的传播模式。

3.2　PLDA+并行算法

3.2.1　已有工作的不足

如上一节的介绍，在PLDA中，文档集合D在P机器上平均分配约D/P文档，在图 3.2(A)表示为一个$\frac{D}{P}$-W矩阵，其中W表示文档集合中的词项个数。词项-主题矩阵则是在每台机器上维护一个本地版本，也就是图 3.2(A)中的W-K矩阵。

在PLDA中，每次吉布斯采样迭代完成后，每台机器上的本地词项-主题矩阵都会进行全局的同步操作。这个同步操作往往由于两种原因导致它的时间过长：(1)由于词项-主题矩阵太大，需要机器间进行大量通信量；(2)由于同步操作需要等待所有机器都完成本轮吉布斯采样才能进行，所以完成较快的机器需要等待最慢的那台机器完成后才能进行同步。为了避免不必要的延时，AS-LDA[90]提出不进行全局的同步。在AS-LDA中每台机器只与另外一台刚完成本轮迭代的机器进行同步。但是，由于词项-主题矩阵可能会因此过时，所以AS-LDA需要进行更多的迭代次数才能够收敛到PLDA类似的学习结果。图 3.3(A)和图 3.3(B)显示了PLDA和AS-LDA的某个词在某台机器上主题更新信息的传播模式。PLDA需要通过一轮吉布斯采样迭代后的全局同步实现更新，而AS-LDA则每次只更新少数几台机器。更重要的是，由于都需要将整个词项-主题矩阵放入内存，PLDA和AS-LDA的内存要求都是O(KW)。

虽然这些方法提供了不同的策略来进行不同机器间的模型融合，已有的这些LDA并行方法有两个共同的特点：

- 这些方法都需要在每台机器的内存中维护一个完整的词项-主题矩阵。
- 这些方法都需要在机器间发送和接收整个词项-主题矩阵来进行机器间的同步。

对于第一个特点，假设需要从一个大规模文档集合估计有W个词和K个主题的$\boldsymbol{\phi}$，那么当W或者K很大的时候，词项-主题矩阵对内存的需求将很容易超过机器的内存配置。而对于第二个特点，通信瓶颈将会极大地限制算法的加速性能。一个高性能计算的研究[98]表明：浮点指令(floating-point instructions)运算速度平均每年提高59%，而机器间的通信带宽每年提高26%，而机器间通信的延迟(latency)每年只下降15%，因此通信瓶颈将会随着科技发展越来越严重。

3.2.2　PLDA+的策略

本章提出PLDA+算法，通过解决机器间通信的瓶颈，进一步提高LDA的并行性能。该算法采取了以下策略：

1. **数据分布(data placement)**。通过精心设计数据分布方式，将面向CPU的任务(CPU-bound tasks)和面向通信的任务(communication-bound tasks)划分到两组不同的机器上。这样能够设计一种流水线的吉布斯采样模式，将通信时间隐藏在计算时间内。
2. **流水线处理(Pipeline processing)**。为了使得一个面向CPU运算的机器不会被通信瓶颈所阻碍(block)，PLDA+设计了一种新型的吉布斯采样模式，可以在对一组词(word bundle)进行吉布斯采样的同时，在后台进行机器间的通信。例如，假设现在要对词“foo”和“bar”进行吉布斯采样。PLDA+在对词“foo”进行吉布斯采样的同时，可以在后台通过机器间通信获取用来对“bar”进行采样所需的信息。这样，获取“bar”信息的通信时间就会被“foo”的采样时间所覆盖。
3. **词项分组(word bundling)**。为了保证通信时间能够被有效地覆盖，进行采样的CPU计算时间必须足够长。再以上述“foo”和“bar”为例，为了覆盖通信时间，对“foo”进行吉布斯采样的计算时间必须要长于针对“bar”的通信时间。假设按照传统的方法对文档中的词依次进行采样，每次吉布斯采样时间将会太短，无法有效覆盖通信时间。然而幸运的是，LDA将文档视为词袋模型，因此并没有考虑词语之间的顺序关系，因此可以对机器上的词语按照任意的顺序进行吉布斯采样。而组合词语的操作，是将一些词组成一个较

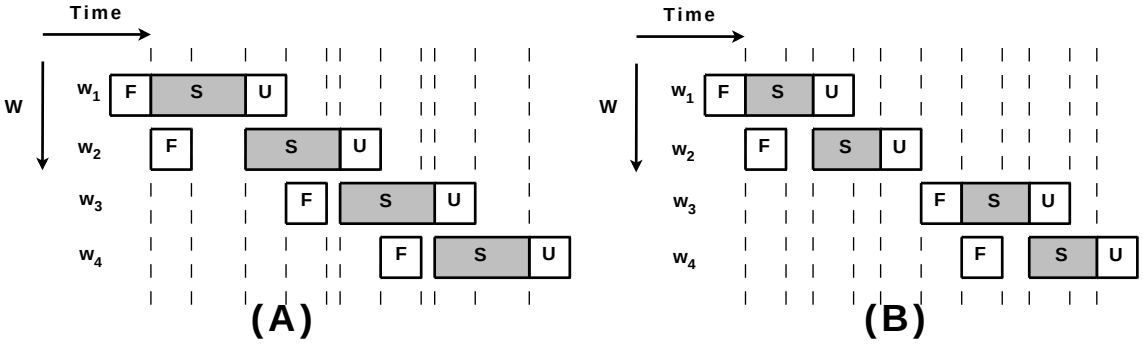


图 3.4　PLDA+中的流水线吉布斯采样策略。(A)：$t_s \geq t_f + t_u$；(B)：$t_s < t_f + t_u$。在该图中，F表示获取主题分布信息操作(fetching)，U表示更新主题分布信息操作(updating)，而S表示采样操作(sampling)。

大的组合来进行吉布斯采样，从而保证通信时间的有效覆盖。

4. **优先调度(priority-based scheduling)**。**数据分布**和**组合词语**是静态的分配策略，用来提高流水线处理方式的性能。但是，许多运行时的随机因素(run time factors)会影响静态策略的有效性。因此，PLDA+同时才与优先调度的策略来帮助解决随机因素造成的瓶颈。

这里首先介绍PLDA+的流水线吉布斯采样方法(pipeline-based Gibbs sampling)。流水线技术被广泛应用于各种领域来提高吞吐量，如现代CPU和GPU设计中的指令流水线(instruction pipeline)[99,100]。如图 3.4所示，是利用流水线吉布斯采样方法对四个词w_1，w_2，w_3和w_4进行采样的过程。图 3.4(A)展示了当$t_s \geq t_f + t_u$的情况，而图 3.4(B)展示了$t_s < t_f + t_u$的情况，其中t_s，t_f和t_u分别表示采样(sampling)，获取用来进行采样的主题分布信息(fetching)和更新主题分布信息(updating)的时间。

在图 3.4(A)中PLDA+首先为w_1获取主题分布信息，然后开始对w_1进行吉布斯采样，同时，它开始获取w_2的主题分布信息。当它完成w_1的吉布斯采样后，PLDA+更新P_w上的w_1的主题分布信息。当$t_s \geq t_f + t_u$的时候，PLDA+能够在完成w_1的计算后马上开始对w_2的采样。这样，PLDA+处理W个词的理想时间是$Wt_s + t_f + t_u$。

图 3.4(B)则显示了一种次优情形(suboptimal scenario)，这里的通信时间没有完全被覆盖。PLDA+没有办法在完成w_2的采样后及时开始对w_3的采样，直到w_2的主题信息被更新以及w_3的主题信息被获取。这个例子表明，为了能够更有效地将通信时间覆盖，必须调度任务，保证$t_s \geq t_f + t_u$。

为了保证$t_s \geq t_f + t_u$，PLDA+将机器划分为两类：一类机器用来维护文档和文档-主题矩阵，进行吉布斯采样，表示为P_d；另外一类机器用来维护词项-主题矩阵，表示为P_w。这种划分方式如图 3.2(B)所示。在每轮吉布斯采样迭代中，一台P_d机器按照以下三个步骤对一个词分配一个新的主题：

1. 从一台P_w获得该词的主题分布信息；
2. 对这个词进行吉布斯采样，分配一个新的主题；
3. 根据分配的新的主题，更新P_w上该词的主题信息。

PLDA+所对应的主题信息传播模式如图 3.3(C)所示，这种模式既避免了PLDA的全局同步，也避免了AS-LDA需要经过大量迭代才能收敛的问题。

PLDA+的一个重要性质是利用了每轮吉布斯采样都可以采取不同的词项顺序。由于LDA认为每个文档是一个词袋，没有考虑单词的顺序，因此可以根据任意的词项顺序进行吉布斯采样。当一个词项在P_d的文档中出现多次，它们可以一起被进行吉布斯采样。而对于那些没有频繁出现的词项，可以将它们与频繁出现的词组合，从而保证采样时间t_s足够长。实际上，如果事先知道$t_f + t_u$的具体时间，算法就可以决定每轮吉布斯采样中的一批(batch)中选取出现过多少次的词项来处理，从而保证$t_s - (t_f + t_u)$最小化。

为了按照词项顺序而非文档顺序进行吉布斯采样，PLDA+在每台P_d机器上建立文档的倒排索引，可以按照词项来检索出现的文档。这里将这些词项组成一个词汇的循环队列(circular queue)，如图 3.5所示。吉布斯采样就按照这个循环队列来进行。为了避免多台P_d机器并发地访问同一个词的主题信息，算法要求不同的P_d机器从队列的不同位置开始访问。例如，图 3.5显示了四台P_d机器，P_{d0}，P_{d1}，P_{d2}和P_{d3}，他们开始吉布斯采样的位置分别是从词项w_0，w_2，w_4和w_6开始。为了保证这个调度算法的运行，PLDA+还需要将词项-主题矩阵按照循环的方式分配到P_w机器上去。这种静态的分配方式有两个优势：

1. 这保证了不同P_w机器的工作量是相对平衡的；
2. 这避免了两台P_d机器同时访问和更新同一个词的主题信息，从而更好地保证了P_w机器上词项-主题矩阵的一致性(serializability)。

需要注意的是，PLDA+的分配方式保证了比PLDA更好的一致性，这是因为在同一轮吉布斯采样中，PLDA+中一台P_d机器可以获得其他P_d机器更新后的词项-主题矩阵。对于数据分布更详细的介绍参见第3.2.3.1节。

虽然可以找到一种最优的方式来分配词项，但是调度却需要处理运行时的动态变化：

1. 首先，一些机器可能会比另外一些机器运行得更快，这将导致在一些P_w机

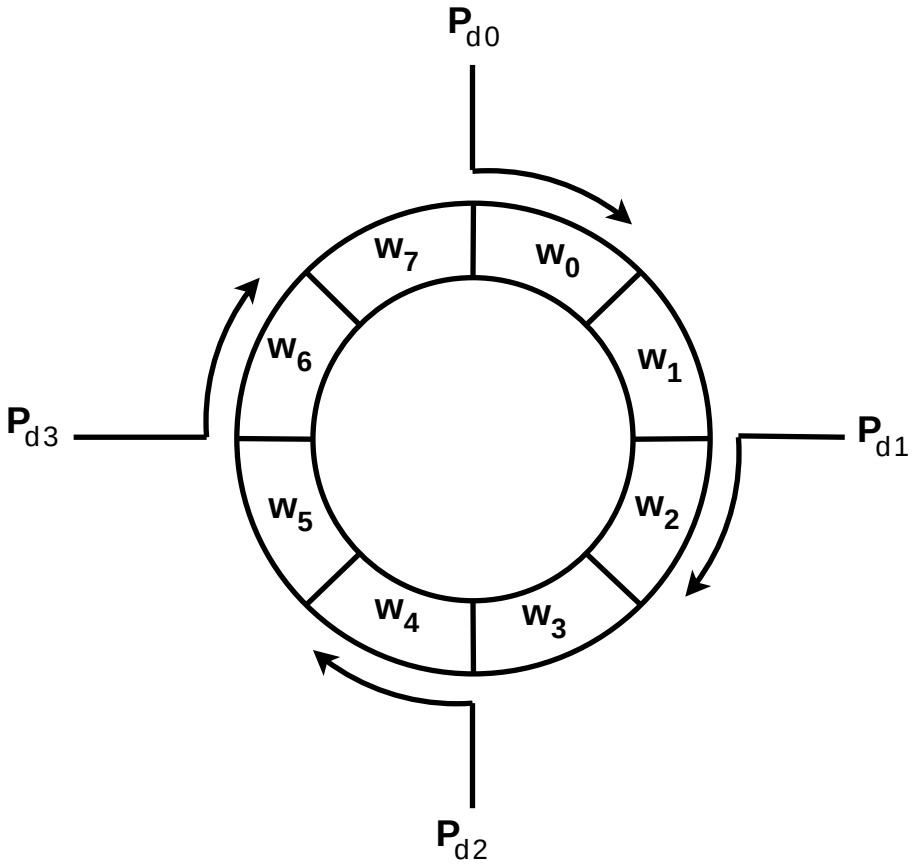


图 3.5　PLDA+中的词汇循环队列(circular queue)。

器上的处理瓶颈;

2. 另外，当多个请求被挂起(pending)，必须能够根据请求的截止时间(deadline)设置优先级。

PLDA+的优先调度策略将在第 3.2.4.3节介绍。

3.2.3　P_w机器上的算法

P_w机器的任务是处理来自P_d的请求。PLDA+将词项-主题矩阵分配到P_w台机器上，每台P_w机器大概需要维护$\frac{W}{|P_w|}$个词和他们的主题分布。

3.2.3.1　P_w机器上的词项分配

词项分配的目标是保证P_w机器间的负载均衡，也就是在每一轮吉布斯采样迭代中，希望保证所有的P_w机器处理大致相同数量的请求。

为了能够实现分配的负载均衡，需要维护两个数据结构:

1. 首先，使用m_i记录出现w_i的P_d机器数目，这是每个词的权重，表明这个词在每轮吉布斯采样中需要发出多少请求。那么对于W个词项，维护一个向量$\vec{m} = (m_1, \ldots, m_W)$。
2. 另外，还要记录每个P_w机器的负载，也就是它所需要负责的词项的请求数目，这个负载向量可以表示为$\vec{l} = (l_1, \ldots, l_{|P_w|})$。

一种简单的分配策略是将词项随机地分配给P_w台机器，这种方法被称为**随机分配(Random Word Placement)**。不幸的是，这种分配方法经常导致负载不均衡。为了保证负载均衡，算法采用**带权轮询调度(Weighted Round-Robin)**方法进行词项分配。这种方法首先将词按照权重降序排列，然后首先从最大权重的词开始取出一个词w_i，将其放入目前负载最低的P_w机器pw上，并更新pw的负载。这种分配过程一直重复直到所有的词项都被分配完毕。经验证明带权轮训调度能够很大概率实现负载均衡[101]。

3.2.3.2　处理来自P_d机器的请求

当为P_w机器分配完词项和他们的主题分布后，P_w机器开始处理来自P_d机器的请求。一台P_w 机器pw首先通过接收来自P_d机器的初始化主题分布，构建它自己的词项-主题矩阵$C_{|pw}^{word}$。然后，每台P_w机器pw开始处理来自P_d机器的请求。PLDA+定义以下三种请求:

- $fetch(w_i, pw, pd)$: 来自P_d机器pd，发往p_w机器pw，用于获取词项w的主题分布的请求。对于每个请求，pw将向pd返回词项w的主题分布$C_{w|pw}^{word}$，这将用于吉布斯采样中的公式(3-2)中的$C_{wk}^{\neg ij}$部分。
- $update(w, \vec{u}, pw)$: 用来更新机器pw上词项w的主题分布的请求。pw将利用$\vec{u}$更新所维护的词项w的主题分布。
- $fetch(pw, pd)$: 来自P_d机器pd，发往p_w机器pw，用于获取P_w机器pw上的所有主题分布数目的请求。pw将对本地所有词项上的主题分布数目求和得到$C_{|pw}^{topic}$，一旦来自所有的P_w机器的$C_{|pw}^{topic}$被pd获取后，pd将求和用于吉布斯采样公式(3-2)中的$C_k^{\neg ij}$部分。

每个P_w机器负责它所维护的词的所有相关请求。为了保证所有的请求能够及时被响应，算法为每个请求设置了一个与截止时间有关的优先级。根据一台P_d机器本地处理的情况，它会对发送的请求有一个允许的最大交互时间。当这台P_d机器发送请求到P_w机器时，每个请求中都包含一个截止时间，而P_w机器会根据请求的截止时间先后来处理请求。

3.2.4　P_d机器上的算法

P_d机器上的算法包括以下几个步骤：

1. 最初，它将文档分配给P_d台机器，然后每台机器上建立文档的倒排索引。
2. 然后，将词汇表中的词项分组，用来进行吉布斯采样和发送请求。
3. 然后，它将调度词项组合的顺序，最小化通信瓶颈。
4. 最后，它开始进行流水线吉布斯采样，迭代直至算法收敛。

接下来分别详细介绍这四个步骤的算法细节。

3.2.4.1　文档分配和构建倒排索引

在进行吉布斯采样之前，首先需要将文档集合D分配到P_d台机器上。文档分配的目的是在P_d间获得更好的CPU计算负载平衡。在PLDA中，由于它的全局同步需要等待每台机器都结束本轮迭代后才能开始，因此会对负载不均衡更加敏感。而PLDA+中的吉布斯采样不需要进行同步。换句话说，一台机器完成本轮吉布斯采样迭代后，可以马上进行下一轮迭代，无需等待其他较慢的机器。但是，这种情况下，算法仍然不希望一些机器太慢，从而相对其他机器少进行若干轮吉布斯采样。这可能会导致与AS-LDA类似的问题——需要更多的迭代次数才能收敛。因此，仍然需要根据负载均衡的原则分配文档。这里通过**随机分配(Random Document Allocation)**的方式实现这一目标。每台P_d机器得到大约$D/|P_d|$个文档，这个过程的时间复杂度为$O(D)$。

文档分配结束后，算法在每台P_d机器上建立对应文档的倒排索引(inverted index)。利用这个倒排索引，一台P_d机器每次获得一个词w的主题分布后，它可以对这个词在这台机器上出现的所有位置都可以进行采样。采样结束后，这台机器可以将更新的主题分布发送给对应的P_w机器。这样做的好处是，对一个词的多次出现仅需要进行两次通信，一次是获取主题分布，另外一次是更新主题分布。这样充分地降低了通信时间。每个词w的索引结构为：

$$w \rightarrow \{(d_1, z_1), (d_1, z_2), (d_2, z_1) \ldots\} \tag{3-5}$$

其中w在文档d_1中出现了2次，因此有两条记录。在具体实现中，为了节约内存，算法会将词w在文档d_1的出现情况保存为一个记录：$(d_1, \{z_1, z_2\})$。

3.2.4.2　词项分组

将词项分组是为了避免一段吉布斯采样的时间太短而不能有效覆盖通信。举一个极端的例子：如果一个词在一台P_d机器上只出现一次。那么在这个词上进行

一次吉布斯采样的时间会非常短，远远短于获取和更新主题分布的时间。解决这个问题的办法是，将一些词组成一个小组，这样一旦获取他们的主题分布信息，就可以进行比较长的吉布斯采样。这里需要确保一个小组内的词项来自同一台P_w机器，这样一次通信的IO就可以获取组内所有词项的主题分布信息。

每台P_d机器根据词项对应的P_w机器进行分组。对于同一台P_w机器的所有词，首先按照他们出现的次数进行倒序排列，构成一个词项列表。然后，反复从这个列表中选取一个最高频的词项和若干个最低词频的词项组成一个小组。当建立词项分组后，每次都发送一个请求获取组内的所有词项的主题分布信息。例如，当从包含12年NIPS论文的NIPS数据集合上学习LDA主题模型时，会把{*curve, collapse, compiler, conjunctive, …*}组成一个小组，其中*curve*是一个高频词，而剩下的是低频词。

3.2.4.3　构建请求调度器(Request Scheduler)

一旦将词项分组完成后，需要构建一个请求调度器能够有效地决定选择哪一个词项分组来发送获取主题分布的请求。这里采用一种伪随机调度策略(pseudo-random scheduling scheme)。

在该策略中，词项存储在一个循环队列中。当进行吉布斯采样的时候，采用顺时针或者逆时针的方式选取词项。每台P_d机器在不同的位置进入这个循环队列，这样可以避免并发地访问同一台P_w机器上的同一个词项。因此，不同P_d机器的不同次吉布斯采样迭代的起始位置都不同。这种随机性可以保证不会每次迭代都形成相同的瓶颈。但是这种循环调度的方法仍然是一种静态的调度策略，当多台P_d机器的请求同时访问同一台P_w机器的时候，仍然会形成并发的瓶颈。因此，一些P_d机器的请求需要等待一段时间才能够得到响应(response)。算法通过为每个请求设置截止日期来解决这个问题，如第 3.2.3.2节所述。在一台P_w机器上的所有请求都按照他们的截止日期来被响应。如果一个请求在截止日期前没有被响应，这个请求就会被自动丢弃。由于吉布斯采样的随机性质(stochastic nature)，偶尔缺失了对一些词的吉布斯采样将不会影响整体的学习效果。算法的伪随机调度策略能够保证同一个词的请求被反复丢弃的可能性很低。

3.2.4.4　流水线吉布斯采样

最后进行流水线吉布斯采样。如公式(3-2)所示，要对文档d_j中的词$x_{ij} = w$计算并分配一个新的主题，需要获得C_w^{word}，C^{topic}和C_j^{doc}。文档d_j的主题分布被一台P_d机器维护，而C_w^{word}则由一台P_w机器维护，C^{topic}则需要通过收集所有P_w机器

上的主题分布获得。因此，在能够给词w分配主题前，这台P_d机器需要从P_w机器上获取C_w^{word}和C^{topic}。然后，这台P_d机器计算和分配新的主题。最后，这台P_d机器将更新的词w的主题分布返回给对应的P_w机器。综上，对于一台P_d机器pd，流水线吉布斯采样的步骤如下:

1. 获取全局主题分布C^{topic}。
2. 选择F个词项组合，将请求其主题分布的请求放入线程池tp中。一旦某个请求被P_w机器响应，则返回对应的主题分布，放入等待队列Q_{pd}中。
3. 对于Q_{pd}中的每个词，获取它的主题分布，对该词的所有出现位置进行吉布斯采样。
4. 吉布斯采样之后，将更新该词主题分布的请求放入线程池tp中。
5. 选择一个新的词项组合，将请求其主题分布的请求放入线程池tp中。
6. 如果满足更新的条件，重新获取新的全局主题分布C^{topic}。
7. 如果不满足停止条件，跳转到第(3)步进行其他词的吉布斯采样。

在第(1)步，pd获取全局主题分布C^{topic}。在这一步，pd发送请求$fetch(pw, pd)$给每一台P_w机器。每台P_w机器返回$C_{|pw}^{topic}, pw \in \{0, \ldots, |P_w| - 1\}$后，$pd$将他们求和得到$C^{topic} = \sum_{pw} C_{|pw}^{topic}$。

由于线程池tp能够并行地发送请求和处理返回的结果，所以在第(2)步中，它同时发出若干个请求，并行地获取主题分布，以避免一些请求可能会出现延迟。由于这些请求是同时发出的，所以他们分配了相同的截止时间。一旦收到某个请求的响应，将马上开始这个词的吉布斯采样。这里设预先发出的请求数目是F。在PLDA+中，F应当设置为一个合适的数字，以保证等待队列Q_{pd}中经常性的有返回的主题分布等待进行吉布斯采样。如果不能满足这一点，PLDA+将停下来等待Q_{pd}中有成员加入，这是PLDA+的通信时间的一部分。为了保证线程池中的线程能够有效利用，F应当大于线程池中的线程数目。

P_w处理全局主题分布的请求是一个非常耗时的操作，这是因为这个操作需要遍历每台P_w机器上的所有词的主题分布情况。幸运的是，AD-LDA[86]的研究显示，吉布斯采样中的主题分配对全局主题分布并不敏感。这里可以通过降低获取全局主题分布的频度来提高P_w机器的效率。因此，在第(6)步不会经常获取全局主题分布。实验将表明，仅仅在每轮吉布斯采样迭代后获取更新的全局主题分布，PLDA+能够获得与LDA和PLDA相同的主题模型。

流水线吉布斯采样的策略可以参考图 3.6，为了让图示更加简洁，图中没有显示获取全局主题分布C^{topic}的过程。

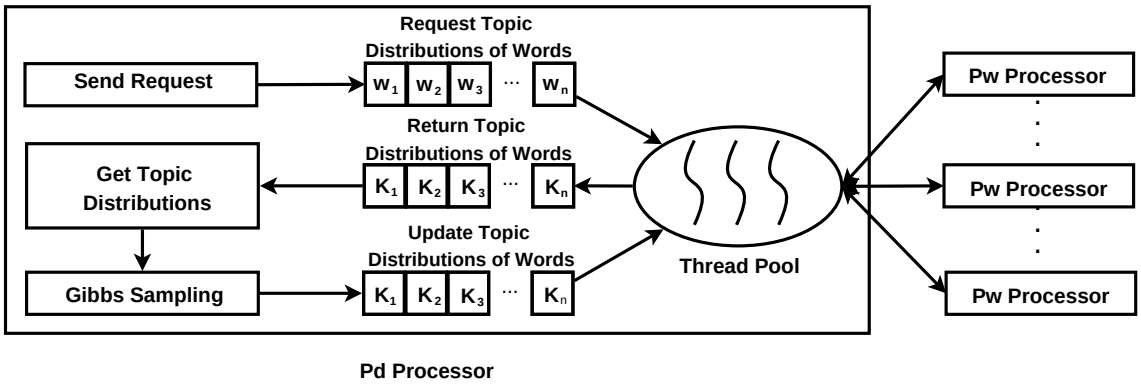


图 3.6　PLDA+的通信机制。

3.2.4.5　容错机制(Fault Tolerance)

PLDA+提供与PLDA类似的容错机制，也就是在P_d机器上对$\boldsymbol{z}_{|pd}$设置检查点。这样做的原因是，一旦出现错误，需要恢复的时候：

1. 在P_d机器上，$\boldsymbol{x}_{|pd}$可以通过数据集合恢复；而$C_{|pd}^{doc}$可以通过$\boldsymbol{z}_{|pd}$恢复。
2. 在P_w机器上，C_{pw}^{word}也可以通过$\boldsymbol{z}_{|pd}$来恢复。

恢复机制的代码放在PLDA+的最开始，如果有检查点信息在硬盘上，说明之前有出错，就加载检查点数据；否则，说明是新运行的PLDA+程序，那么进行随机初始化。

3.2.5　参数和复杂度分析

这一部分将分析PLDA+的主要参数，以及PLDA和PLDA+的复杂度。

3.2.5.1　参数

给定机器数目P，PLDA+的第一个参数是P_w机器和P_d机器数目的比例$\gamma = |P_w|/|P_d|$。γ越大，说明P_d进行吉布斯采样的时间将会增大，因为这部分机器变少了；而同时进行交互的时间将会降低，这是因为更多的P_w机器用来处理来自P_d的请求。因此，需要平衡这两类机器之间的数目，以达到以下两个目的：

1. 同时最小化计算(computing)和通信(communication)的时间。
2. 保证通信时间足够的短，能够被计算时间所覆盖。

一旦知道用来进行吉布斯采样和通信的平均时间，就可以决定这个参数的设置。假设一台机器对整个数据集合进行吉布斯采样的总时间为T_s，两台机器间传输所有词的主题分布的时间为T_t。对于P_d台机器，采样时将为$T_s/|P_d|$。假设同时

向P_w传送主题分布，则通信时间为$T_t/|P_w|$。为了保证采样时间能够覆盖获取和更新的通信时间，需要保证：

$$\frac{T_s}{|P_d|} > \frac{2T_t}{|P_w|}. \tag{3-6}$$

假设$T_s = W\bar{t_s}$，其中$\bar{t_s}$是一个词的所有出现位置的平均采样时间；而$T_t = W\bar{t_f} = W\bar{t_u}$，其中$\bar{t_f}$和$\bar{t_u}$是对一个词的信息进行获取和更新的平均时间，那么，

$$\gamma = \frac{|P_w|}{|P_d|} > \frac{\bar{t_f} + \bar{t_u}}{\bar{t_s}}, \tag{3-7}$$

其中$\bar{t_f}$，$\bar{t_u}$和$\bar{t_s}$可以通过在小数据集合上运行PLDA+来经验性地估计，这样就可以得到一个近似的γ值。在实验中，本章设置$\gamma = 0.6$。

PLDA+的第二个参数是线程池中的线程个数R，这表明可以并行地发送请求的个数。使用线程池是为了避免采样会被某些比较繁忙的P_w机器阻塞，所以R是由网络质量来决定的。R可以在吉布斯采样中经验性地自动调节。也就是说，当等待时间过长时，需要增加线程数。

PLDA+的第三个参数是预先并行发送获取主题分布的请求数F。这个数目与R相关，在实验中设置$F = 2R$。

PLDA+的最后一个参数是获取全局主题分布的间隔$inter_{max}$。这个参数将会影响PLDA+学习主题模型的质量。实验中，设置$inter_{max} = W$可以达到与LDA和PLDA类似的学习质量。

需要注意的是，PLDA+的最优参数设置与分布式环境息息相关，这包括网络带宽和机器计算速度和内存大小等。

3.2.5.2　复杂度分析

表 3.1总结了PLDA+中P_d机器和P_w机器在时间和空间的复杂度。为了进行比较，该表同时列出了LDA和PLDA的复杂度。这里假设$P = |P_w| + |P_d|$来进行PLDA和PLDA+间的比较。在这个表格中，I表示吉布斯采样的迭代次数，c是一个常数，表示从带宽(bandwidth)到每秒浮点计算(flops)之间的转换。

PLDA的预处理包括将文档分配到P台机器上，时间复杂度为$D/|P|$。与PLDA相比，PLDA+的预处理更加复杂，包括三个额外的操作：

1. 每台P_d机器需要建立一个倒排索引，时间复杂度为$O(D/|P_d|)$。
2. 为词项分组，需要对词项按照词频进行快速排序，时间复杂度为$O(W \log W)$。

表 3.1　算法复杂度。在这个表格中，I表示吉布斯采样的迭代次数，c是一个常数，表示从带宽(bandwidth)到每秒浮点计算(flops)之间的转换。

方法	时间复杂度		空间复杂度
	预处理	吉布斯采样	
LDA	-	INK	$K(D+W)+N$
PLDA	$\frac{D}{\|P\|}$	$I(\frac{NK}{P}+cKW\log P)$	$\frac{(N+KD)}{P}+KW$
PLDA+,P_d	$\frac{D}{\|P_d\|}+cW\log W+\frac{WK}{\|P_w\|}$	$\frac{INK}{\|P_d\|}$	$\frac{(N+KD)}{\|P_d\|}$
PLDA+,P_w	-	-	$\frac{KW}{\|P_w\|}$

3. 将P_d机器上的主题分配情况发送给P_w机器以构成初始化的词项-主题矩阵，时间复杂度为$O(WK/|P_w|)$。

在实践中，LDA需要运行数百次迭代，因此以上PLDA+增加的预处理时间与训练时间相比并不明显。

最后考察PLDA+的加速效率(speedup efficiency)。对PLDA+而言，不考虑预处理，假设$\gamma=|P_w|/|P_d|$，那么理想加速效率为:

$$\text{speedup efficiency}=\frac{S/P}{S/|P_d|}=\frac{|P_d|}{P}=\frac{1}{1+\gamma}, \tag{3-8}$$

其中S表示LDA在单机上的运行时间，S/P表示利用P台机器的运行时间，而$S/|P_d|$则表示PLDA+的理想运行时间，这时通信时间完全被计算时间所覆盖。

3.2.6　实验结果

实验通过真实数据比较了PLDA+和PLDA的学习模型质量和加速性能。如研究[90,102]表明，AS-LDA与AD-LDA的加速效果类似，因此这里没有与AS-LDA比较。

3.2.7　数据集合和实验环境

这里采用三个数据集合进行实验，如表 3.2所示。NIPS数据集包含了来自NIPS会议的学术论文，这个数据集合相对较小，用来对算法学习模型的质量和错过截止日期的请求比例。两个Wikipedia数据集是2008年3月份的维基百科英文词条的集合，其中一个集合的词项个数为20,000，另一个是200,000，分别命名为Wiki-20T和Wiki-200T。与Wiki-20T相比，在Wiki-200T会有更多的低频词。但

表 3.2　数据集合详细信息。

	NIPS	Wiki-20T	Wiki-200T
D_{train}	1,540	2,122,618	2,122,618
W	11,909	20,000	200,000
N	1,260,732	447,004,756	486,904,674
D_{test}	200	-	-

是，即使对于排名在200,000左右的词，他们也在维基百科出现在至少24个词条中，这足够让LDA学习他们的主题信息。这两个大规模数据用来检验PLDA+的加速性能。实验采用同步的远端程序调用(synchronous remote procedure call，RPC)实现PLDA+。实验在一个拥有2,048台机器的分布式计算环境中进行，每台机器的配置为2GHz的CPU，3GB内存和超过100GB的硬盘空间。

3.2.8　请求错过截止日期的影响

与AD-LDA论文[89]类似，这里采用测试集合混淆度(test set perplexity)来度量学习模型的质量。混淆度(perplexity)是自然语言处理中用来评价语言模型(language model)的常用指标：

$$Perp(\boldsymbol{x}^{\text{test}}) = \exp\left(-\frac{1}{N^{\text{test}}}\log p(\boldsymbol{x}^{\text{test}})\right), \tag{3-9}$$

其中$\boldsymbol{x}^{\text{test}}$表示测试集合，$N^{\text{test}}$表示测试集合的大小。如果一个模型的混淆度越小，说明它的预测能力越强，该模型的质量越好。对于LDA的混淆度计算，算法对每个测试文档中的单词随机地划分为两部分，其中一部分用来估计文档的主题分布$\boldsymbol{\theta}_j$，另外一部分用来计算混淆度。LDA模型的混淆度的计算与[82]相同，也就是由于吉布斯采样的随机特性，算法需要进行多次不同的吉布斯采样，然后将不同模型的混淆度求平均，这里进行了$S = 40$次不同的训练，然后计算：

$$\log \Pr(\boldsymbol{x}^{test}) = \sum_{j,w} C_{jw}^{\text{test}} \log \frac{1}{S}\sum_{k} \boldsymbol{\theta}_{kj}^{S}\boldsymbol{\phi}_{wk}^{S}, \tag{3-10}$$

其中C_{jw}^{test}是单词w在文档d_j中出现的次数。其中

$$\boldsymbol{\theta}_{kj} = \frac{C_{kj}^{S} + \alpha}{\sum_{k=1}^{K} C_{kj}^{S} + K\alpha} \qquad \boldsymbol{\phi}_{wk} = \frac{C_{wk}^{S} + \beta}{\sum_{w=1}^{W} C_{wk}^{S} + W\beta} \tag{3-11}$$

通过计算NIPS数据集合上的混淆度，可以看到PLDA+的学习质量和收敛速度与单机上的LDA以及PLDA类似，由于这个结论与AD-LDA论文[89]类似，这里就不再赘述。

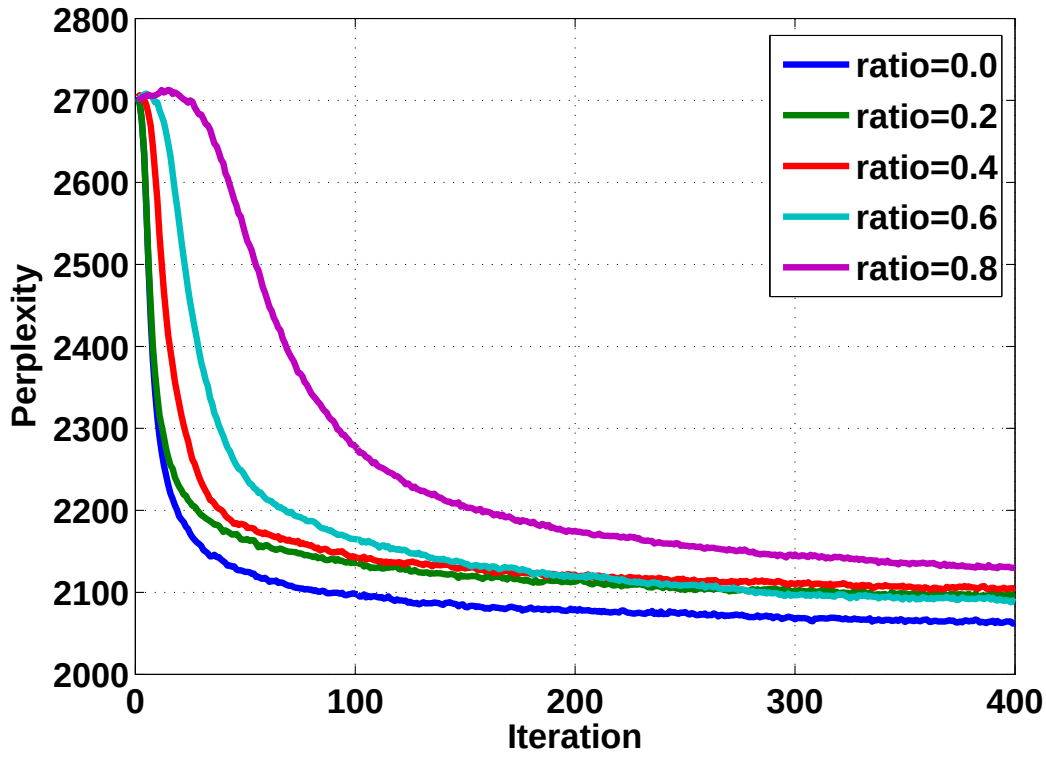


图 3.7　当丢弃比例为0.0，0.2，0.4，0.6和0.8的时候，混淆度与迭代次数之间的关系。

如第 3.2.4.3节的介绍，PLDA+会丢弃那些错过截止时间的请求。本章将在NIPS数据集合上考察丢弃这些请求对学习质量的影响。首先定义丢弃比例(missing ratio)δ为平均每轮吉布斯采样迭代中丢弃的请求数除以请求总数，这个比例在$[0.0, 1.0)$中取值。实验通过随机丢弃δ比例的请求，来考察学习模型质量的变化。在实验中，设置机器数为$P = 50$。图 3.7显示了当主题个数$K = 10$的时候，不同丢弃比例δ下的混淆度随着迭代次数的变化情况。当丢弃比例小于60%的时候，混淆度能够保持较优的情况。而当迭代次数为400的时候，丢弃比例在20%和60%之间的混淆度基本相同。当然，没有丢弃的情况下，混淆度有2%的优势。但是，2%的混淆度的下降不会造成学习模型质量的显著变化。图 3.8显示了不同主题个数的情况下，收敛后的混淆度(400次迭代)与丢弃比例之间的关系。可以看到，主题个数较多的模型会造成混淆度的较大幅度下降。而$\delta = 60\%$仍然是一个比较合理的阈值，只要PLDA+保证丢弃比例不会高于60%，就能够学习到较好的主题模型。而在实际环境中，丢弃比例往往低于1%，远低于这个60%的阈值。虽然丢弃比例很大程度上依赖于计算环境和每台机器的工作量，这个实验的结果证明，PLDA+在丢弃比例δ很大的情况下依然能够学习得到高质量的主题模型。

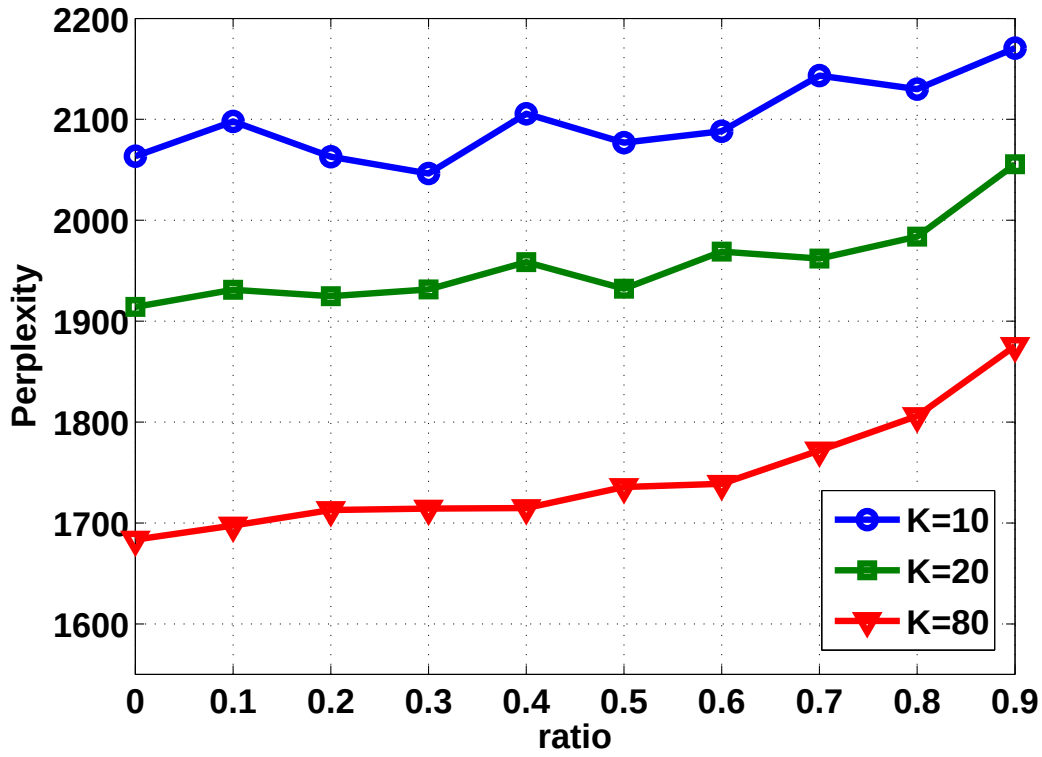


图 3.8　当在不同的主题个数下，混淆度与丢弃比例之间的关系。

3.2.9　加速性能

设计LDA的并行算法的主要目标是获得更好的加速性能。这里将考察PLDA+的加速性能，并与PLDA进行比较。实验将在Wiki-20T和Wiki-200T上进行加速性能实验。设主题个数为$K = 1,000$，在Wiki-20T上考察机器数目为$P = 64, 128, 256, 512$和$1,024$下的加速性能，在Wiki-200T上考察机器数目为$P = 64, 128, 256, 512, 1,024$和$2,048$下的加速性能。由于PLDA+有$P = P_w + P_d$，并设置$\gamma = 0.6$，在线程池中的线程数为$R = 50$。根据第 3.2.5.2节的分析，PLDA+的理想加速比为$\frac{1}{1+\gamma} = 0.625$。

图 3.9比较了在Wiki-20T上的加速性能。加速性能是与机器数$P = 64$的时候进行比较的，因为如此大规模数据在单台机器上会由于内存的限制无法运行。这里假设$P = 64$的加速比为64。从该图可以观察到，当机器数目增加的时候，PLDA+因为能够克服通信瓶颈所以能够获得比PLDA更优的加速性能。图 3.10显示了在Wiki-20T上不同机器数目的情况下，通信时间和采样时间的对比。当机器数$P = 1,024$的时候，PLDA的通信时间为13.38秒，与采样时间几乎相同，这远大于PLDA+的3.68秒。

从这个比较结果可以总结:

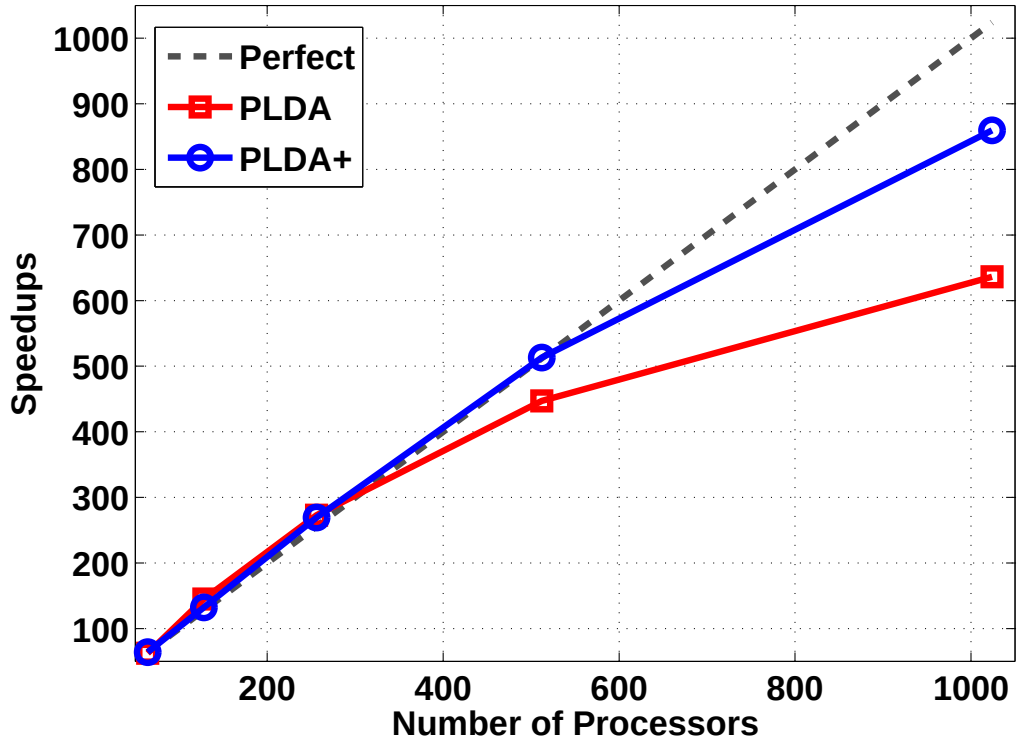


图 3.9　在Wiki-20T数据集合上，从64台机器到1,024台机器的加速情况。

1. 当机器数足够大时，如$P = 512$，PLDA+开始显示出比PLDA更优的加速性能。
2. 而实际上，如果也把PLDA中不同机器用来互相等待的时间算入的话，PLDA的加速性能将大打折扣。例如，在比较繁忙的计算环境中，当机器数为$P = 128$的时候，PLDA大概需要花费70秒进行通信，而其中只有10是用来传输词项-主题矩阵的，而其他绝大部分时间是用来机器之间的互相等待的。

在更大的Wiki-200T数据上，如图 3.11所示，PLDA的加速性能在机器数增长到$P = 512$后没有显著增长，而PLDA+能够保持几乎线性的加速性能。对于PLDA+，并行发送的请求数为$F = 100$，线程数$R = 50$。在词项数$W = 200,000$和主题数$K = 1,000$的时候，词项-主题矩阵占用1.6GB内存，这对通信是一个很大的考验。在图 3.12中显示了这个数据上的通信时间和采样时间的比较。PLDA+在从$P = 64$到$P = 2,048$始终保持了通信时间短于采样时间。当$P = 2,048$的时候，PLDA+只需要20分钟完成100轮迭代，而PLDA要使用160的时间。虽然Amdahl定律(Amdahl's Law)最终会限制无限加速下去的可能，但很明显，PLDA+能够获得比PLDA更显著的加速性能。

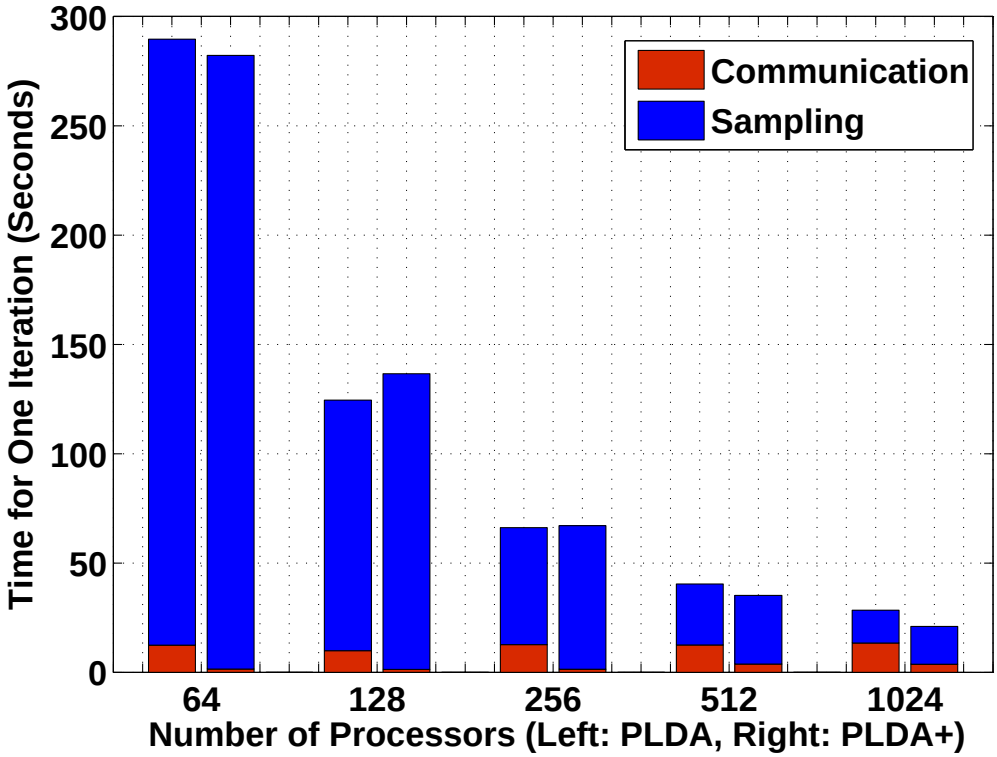


图 3.10　在Wiki-20T数据集合上，从64台机器到1,024台机器的通信和采样时间的比例。

以上的比较没有考虑预处理所花费的时间。实际上，根据第 3.2.5.2节的分析，PLDA+的预处理时间与学习时间相比很短。例如，在$P = 2,048$台机器上用PLDA+学习Wiki-200T的主题模型的预处理时间仅需35秒，而数百次吉布斯采样的迭代，每轮迭代都需要大约13秒时间。

3.2.10　加速算法PLDA+小结

为了解决已有LDA并行算法的瓶颈，本章提出了PLDA+并行算法。通过流水线吉布斯采样算法，算法成功地解决了传统并行算法中的通信瓶颈问题。实验证明，PLDA+算法能够在大规模数据上进行高效的学习。这为利用隐含主题模型研究关键词抽取问题奠定了基础。

3.3　基于隐含主题模型的关键词抽取方法

一旦从大规模数据中学习得到主题模型后，利用隐含主题模型进行关键词抽取的过程非常简单，主要分为两个步骤：

1. 获取文档中的候选关键词，这里仍然通过词性标注选取名词性短语作为候选关键词。

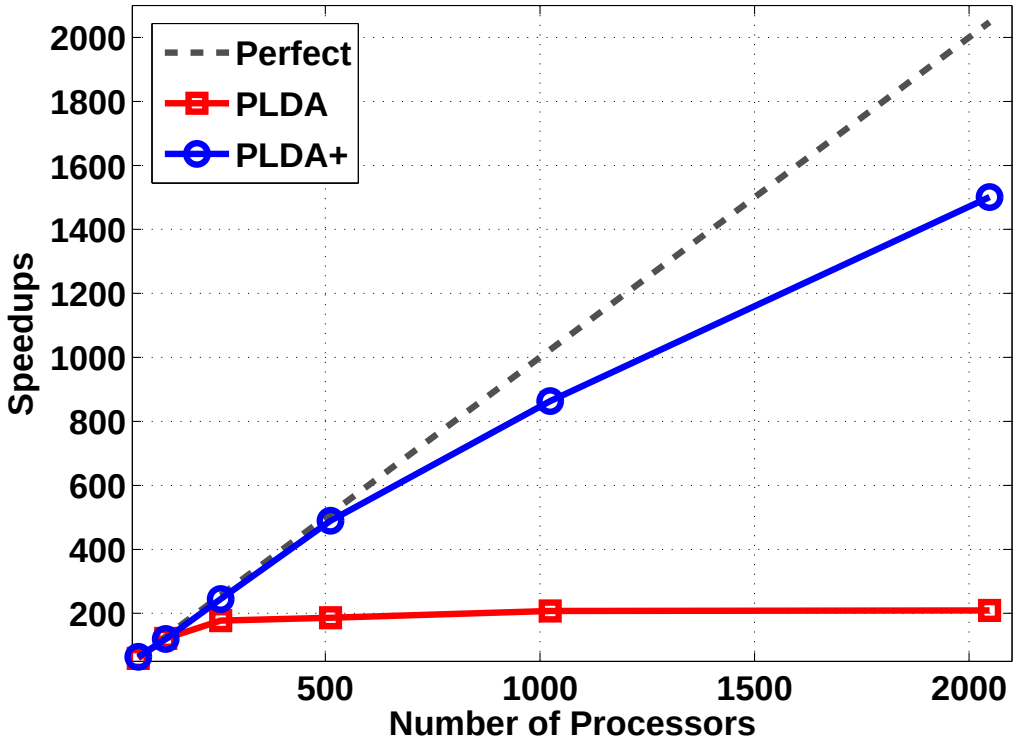


图 3.11　在Wiki-200T数据集合上，从64台机器到1,024台机器的加速情况。

2. 根据从大规模语料学习得到的隐含主题模型，计算获取文档和候选关键词的主题分布。
3. 计算文档和候选关键词的主题相似度，排序并选取最高的几个作为关键词。

由于第1步与上一章类似，这里将对第2和第3步分别详细介绍。

3.3.1　获取文档和候选关键词的主题分布

首先对于一个词w，可以根据公式(3-3)得到它在某个主题k上的概率:

$$\Pr(k|w) = \frac{C_{wk} + \beta}{\sum_{k=1}^{K} C_{wk} + K\beta} \tag{3-12}$$

当然同时也可以得到主题k中某个词w的概率:

$$\Pr(w|k) = \boldsymbol{\phi}_{wk} = \frac{C_{wk} + \beta}{\sum_{w=1}^{W} C_{wk} + W\beta} \tag{3-13}$$

这样，给定一个候选关键词p，它可能包括多个词，那么对于这个候选关键词，可以计算:

$$\Pr(p|k) = \prod_{w \in p} \Pr(w|k) \tag{3-14}$$

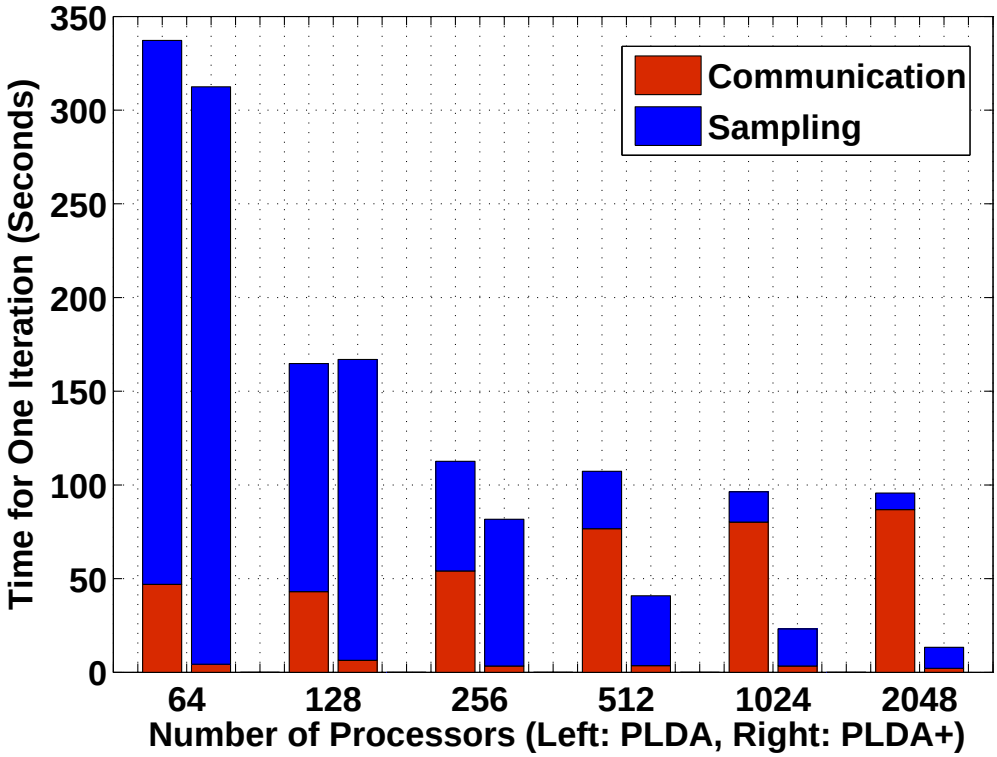


图 3.12　在Wiki-200T数据集合上，从64台机器到1,024台机器的通信和采样时间的比例。

以及

$$\Pr(k|p) = \frac{\Pr(p|k)\Pr(k)}{\Pr(p)} = \frac{n_k}{n_t}\prod_{w\in p}\Pr(w|k) \tag{3-15}$$

其中$\frac{\Pr(k)}{\Pr(p)}$由训练集合中主题k出现的次数除以p中的词项个数来近似。

而对于文档d_j，类似于学习的过程，可以利用吉布斯采样，对其中的每个词利用公式(3-2)赋予其主题，通过反复迭代直到收敛，然后根据文档中词的主题赋予情况，根据公式(3-4)得到该文档的主题分布：

$$\Pr(k|d_j) = \frac{C_{kj} + \alpha}{\sum_{k=1}^{K} C_{kj} + K\alpha} \tag{3-16}$$

与学习的过程不同之处在于：在这里，已经学习的到的词项-主题矩阵是固定的，也就是说这里并不需要更新词项-主题矩阵，而只需要不断更新文档-主题分布即可。

3.3.2　计算文档和候选关键词相似度

这里可以采取以下几种方式来对文档中的关键词进行排序。

首先，给定一个文档的主题分布Pr($k|d$)和候选关键词的主题分布Pr($k|p$)，可以通过余弦相似度、KL距离等来计算两个主题分布之间的相似度来对候选关键词进行排序。其中余弦相似度可以表示为:

$$cos(p;d) = \frac{\sum_{k=1}^{K} \Pr(k|p)\Pr(k|d)}{\sqrt{\sum_{k=1}^{K} \Pr(k|p)\Pr(k|p)} \times \sqrt{\sum_{k=1}^{K} \Pr(k|d)\Pr(k|d)}} \tag{3-17}$$

而KL-divergence是一个非对称的距离度量方法:

$$KL(t;d) = \sum_{k=1}^{K} \Pr(k|p) log \frac{\Pr(k|p)}{\Pr(k|d)} \tag{3-18}$$

此外，还有一种方法是采用预测似然度(predictive likelihood)[103]:

$$PL(p;d) = \sum_{k=1}^{K} \Pr(p|k)\Pr(k|d) \tag{3-19}$$

这度量了给定文档d后，通过隐含主题产生某个关键词p的概率。

3.4　实验结果与分析

3.4.1　实验数据和评价指标

本章选取两个数据进行关键词抽取实验。其中一个数据来自Wan等人①，最初用在论文[19,20]中。这个数据包含了来自DUC2001[104]的308篇新闻文档，手工标注了2,488个关键词，平均每篇文档最多标注10个关键词，实验命名这个数据集为NEWS。另外一个数据集来自Hulth，最初用在论文[12]中，该数据包含2,000个论文摘要，手工标注了19,254个关键词。在实验中称这个数据集为RESEARCH。

实验采用两种学习隐含主题模的方式：(1)利用NEWS和RESEARCH数据集合学习主题模型；(2)利用维基百科学习主题模型。采用第二种方式的原因是NEWS和RESEARCH包含文档数较少，可能不足以训练隐含主题模型，因此需要利用上面提出的并行算法在维基百科英文词条上训练主题模型，这里选用了2008年3月份的集合②来学习主题模型。在去除非词条页面，以及长度小于100个词的词条后，得到2,122,618篇词条，然后根据词项的文档频度选取最高的20,000个作为词汇表。这里将每篇维基百科词条作为一篇文档，训练隐含主题模型。实验尝试了主题个数从50到1,500的各种可能。当遇到主题模型中没有出现的词的时候，算法就返回一个均匀分布。

① http://wanxiaojun1979.googlepages.com。

② http://en.wikipedia.org/wiki/Wikipedia_database。

表 3.3　在NEWS数据集合上不同方法推荐$M = 10$个关键词时的效果比较。

方法	Precision	Recall	F_1-Measure
TFIDF	0.239	0.295	0.264
TextRank	0.242	0.299	0.267
NEWS,PL,$K = 50$	0.258	0.318	0.285
Wiki,PL,$K = 1500$	0.267	0.329	0.294

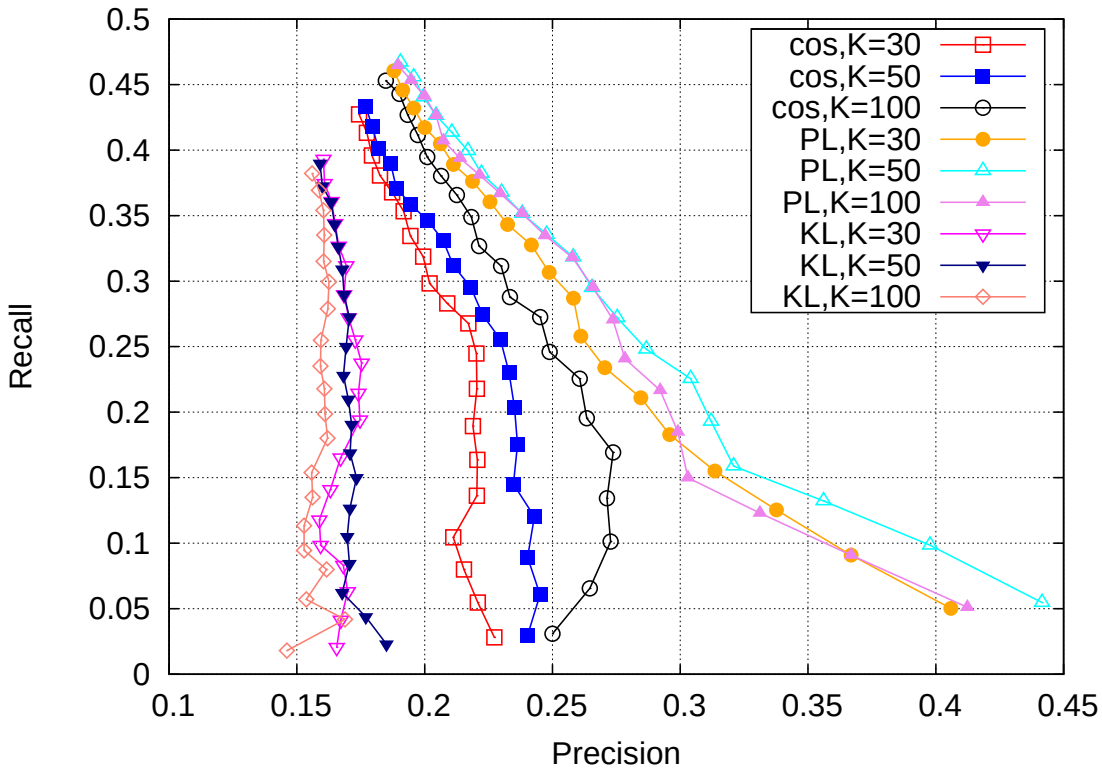
实验采用准确率、召回率和F_1值(precision/recall/F_1-Measure)来评价关键词抽取的效果，如公式(2-8)所示。

3.4.2　实验结果

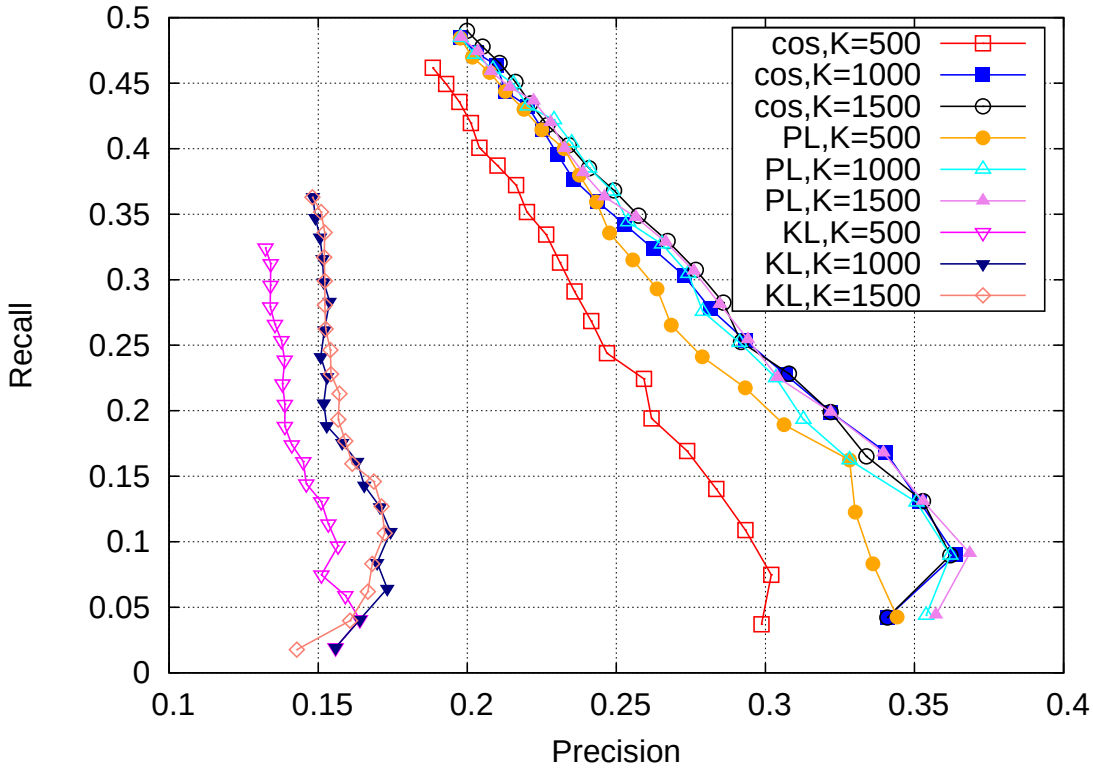
首先图 3.13和图 3.14分别展示了NEWS和RESEARCH数据集合上，利用不同数据学习隐含主题模型、采用不同相似度计算方法、设置不同的主题个数的效果比较。其中，“NEWS”，“Wiki”表示训练的文档集合，PL等表示度量相似度的方法，K表示设置的主题个数。这里，当在NEWS数据和RESEARCH数据集合上学习LDA模型的时候，由于数据集合较小，所以设置的主题个数相对较少；而当在Wikipedia集合上学习LDA模型的时候，由于数据规模较大，所以设置的主题个数相对较多。从图中可以观察到:

1. 在Wkipedia上学习的主题模型比在NEWS/RESEARCH数据上学习的主题模型表现相对较好，但是优势比较有限。这说明在大规模数据上学习的主题模型在某个特定数据集合上应用的时候，面临着知识迁移的问题。
2. 预测似然度方法除了在RESEARCH数据上利用自身数据学习主题模型的效果较差外，相对表现较好；余弦相似度方法除了在个别情况下，也能保持较好的推荐效果；KL方法相对表现较差。
3. 大多数情况下，主题个数的设置在合理范围内对关键词抽取的影响较小。

为了进一步比较关键词抽取效果的有效性，在 3.3和表 3.4比较了TFIDF、TextRank和两种学习主题模型方式的关键词抽取效果。从这里，可以看到利用主题模型能够比较TFIDF和TextRank更有效地抽取关键词。由于本章与下章算法有较大相关性，所以这里不给出示例，而在下章一并给出例子并作分析。这里没有列出上章聚类算法的效果，是因为该算法在NEWS数据集合上的评价结果较差，这是由于在这种较长的文档上聚类算法倾向于推荐出较多的关键词，但又没有办法有效地判断他们的重要性。

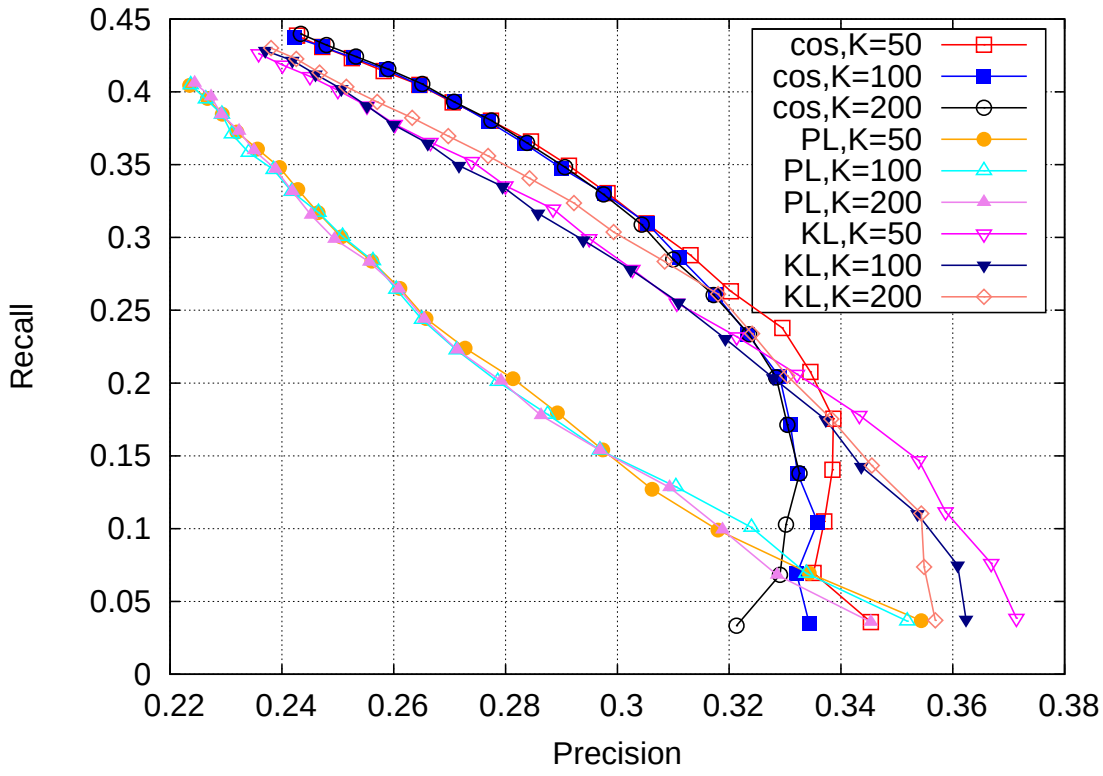


(a) 在NEWS数据集合学习LDA模型

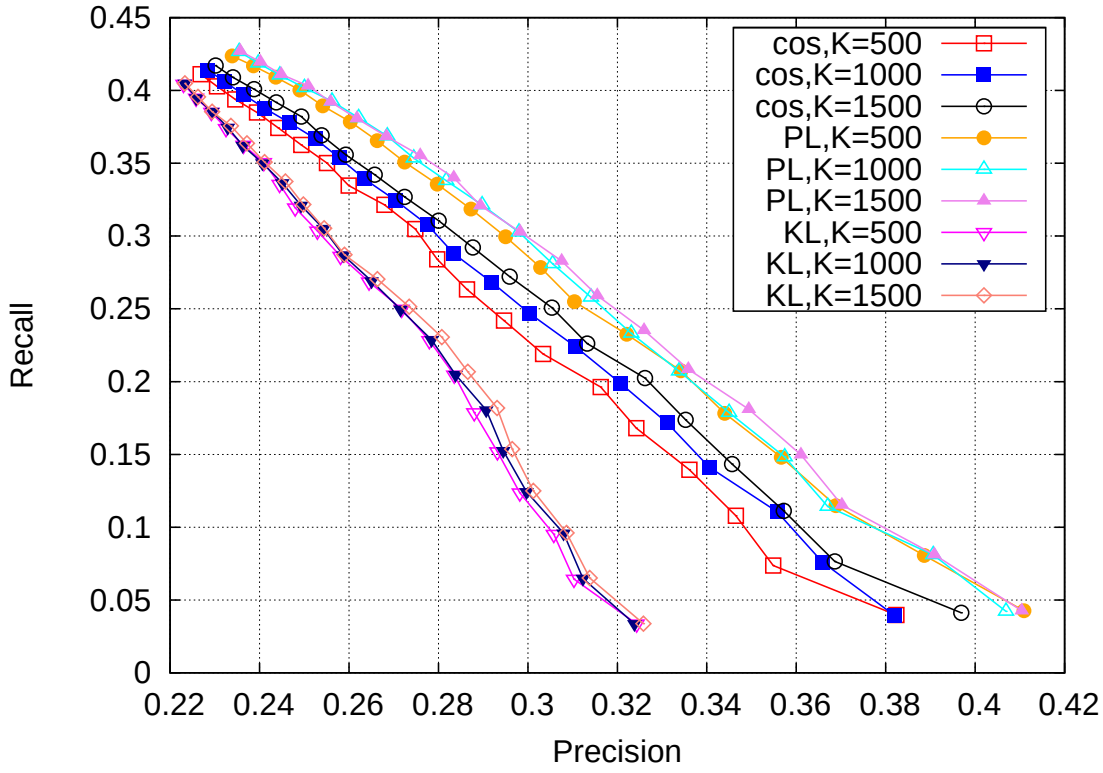


(b) 在Wikipedia数据集合学习LDA模型

图 3.13　在NEWS数据集合上，在两种不同数据学习的隐含主题模型下的效果比较。



(a) 在RESEARCH数据集合学习LDA模型



(b) 在Wikipedia数据集合学习LDA模型

图 3.14　在RESEARCH数据集合上，在两种不同数据学习的隐含主题模型下的效果比较。

表 3.4　在RESEARCH数据集合上不同方法推荐$M = 5$个关键词时的效果比较。

方法	Precision	Recall	F_1-Measure
TFIDF	0.333	0.173	0.227
TextRank	0.330	0.171	0.225
RESEARCH,cos,$K = 50$	0.343	0.178	0.234
Wiki,PL,$K = 1500$	0.349	0.181	0.238

3.5　本章小结

本章探索了利用文档外部的大规模文档集合，通过隐含主题模型构建文档主题并进行关键词抽取。本章首先针对隐含主题模型训练速度较慢的瓶颈，提出了一种高效的并行隐含主题模型，主要的思路是采用流水线的思想并行吉布斯采样中的通信和计算部分，该算法在LDA的并行方面取得了巨大进展。然后，本章系统考察了训练主题模型的文档集合、度量文档和候选关键词的主题相似度的不同方法对关键词抽取的影响。实验证明，该方法能够更好地构建文档主题，并有效抽取覆盖文档主题的关键词。

第 4 章　利用隐含主题模型和文档结构的关键词抽取方法 ①

上一章主要介绍了利用隐含主题模型进行关键词抽取的方法。但是这个方法存在一个重要问题，就是没有考虑文档的结构信息。本章将提出一个综合利用隐含主题模型和文档结构的关键词抽取方法。

4.1　基于隐含主题模型和基于文档结构方法的问题

第一章已经提到，进行关键词抽取主要有两种路线。一种是将关键词抽取当作分类问题，即对每个候选关键词判断是否为关键词的二分类问题[4]。这类方法因为需要人工标注训练集合，费时费力，因此不适合网络上的大规模应用。

另外一个路线是无监督方法，采取各种手段对候选关键词进行排序，然后选取最高的几个作为推荐关键词。在这个路线中，最流行的方法是TextRank[16]，一种基于PageRank的图方法。该方法由于考虑了文档中词与词之间的同现关系，因此比传统的TFIDF方法表现要优。

相比起TFIDF而言，图方法可以看作是对文档结构的一种建模。而上一章提出利用隐含主题模型进行关键词抽取，虽然能够通过隐含主题考虑文档主题信息，但是在另外一个方面，却没有考虑一篇文章本身中词与词之间的结构关系。

从图方法方面来看这个问题，已有的图方法如TextRank等，只为每个词维护一个排序值。而实际上一篇文档可能包含多个不同的主题，例如本章包括“关键词抽取”、“随机游走”和“隐含主题模型”三个主题。每个词可能在不同的主题上表现出不同的重要性。例如，词“短语”和“抽取”会在“关键词抽取”这个主题中占有重要地位，而单词“图”和“PageRank”会在“随机游走”这个主题上占有重要位置，等等。

因此，本章提出一个综合考虑隐含主题模型和文档结构的关键词抽取方法。该方法将隐含主题模型和基于随机游走的图方法结合在了一起。该算法将传统的PageRank分解成在不同主题上的带偏好的PageRank。这样，每个词在不同的主题上会有不同的PageRank排序值。最后根据文档的主题分布，可以计算得到每个候选关键词的最终排序值，用以推荐关键词。这种基于隐含主题的PageRank算

① 本章主要内容以“Automatic Keyphrase Extraction via Topic Decomposition”为题作为学术论文发表在2010年的国际学术会议“The Conference on Empirical Methods in Natural Language Processing (EMNLP'10)”上。

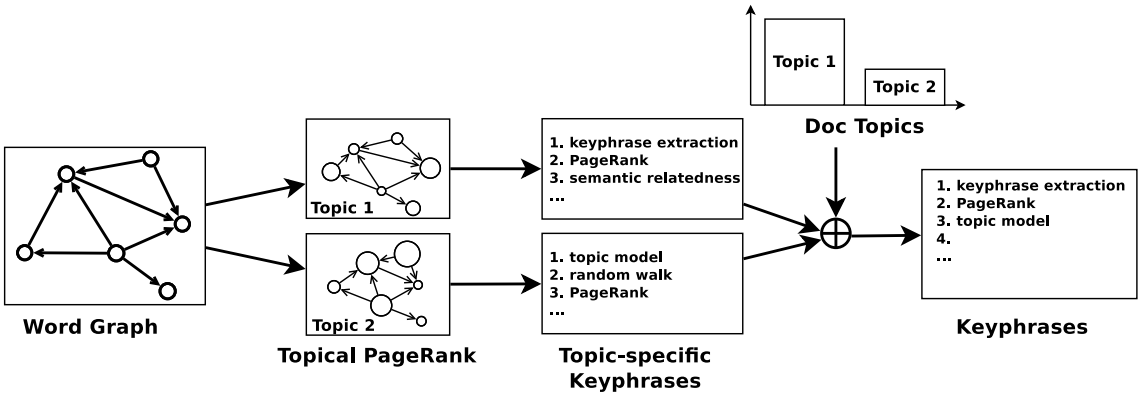


图 4.1　主题随机游走的关键词抽取方法流程。

法被称为Topical PageRank(TPR)。该方法能够使抽取关键词具有更好的主题覆盖度。

4.2　基于隐含主题模型的图方法

基于隐含主题模型的图方法(Topical PageRank，TPR)通过两个阶段进行关键词抽取：

1. 构建一个主题解释器(topic interpreter)来得到给定单词和文档的主题。
2. 运行TPR算法，从文档中抽取关键词。

其中第一步已经在上一章进行了比较详细的介绍。这里主要介绍TPR算法。

给定文档d，利用TPR进行关键词抽取主要包括四个步骤，如图 4.1所示：

1. 根据文档d中单词的同现关系，构建文档对应的单词图；
2. 在图上运行TPR算法，得到每个单词在不同主题上的PageRank值；
3. 在不同的主题上，根据单词在该主题上的PageRank值得到候选关键词在该主题上的值；
4. 获得文档d的主题分布，综合不同主题上候选关键词的重要性，得到对候选关键词的最终排序，选取排序最高的若干作为推荐关键词。

4.2.1　构建单词图

这里需要根据给定文档中单词的同现关系来构建同现图，这表示了在该文档中词与词之间的语义关系。文档被当作一个单词序列，然后通过移动滑动窗口不断向图中增加边。TextRank论文[81]报告单词图是否有向对关键词抽取的影响不大。这里选择构建一个有向图：在每个滑动窗口中，将该窗口中的第一个词指向剩余的其他词。由于关键词一般都是名词短语，所以在构建图的时候，仅考虑形

容词和名词。

4.2.2　Topical PageRank (TPR)

在介绍TPR之前，这里首先给出一些定义和符号。文档的单词图可以表示为: $G = (V, E)$，其中的节点是文档中的名词和形容词$V = \{w_1, w_2, \cdots, w_N\}$，其中的每条边$(w_i, w_j) \in E$表示一条从单词$w_i$到单词$w_j$的链接。在一个单词图中，每个节点表示一个单词，每条边表示单词之间的语义上的关联程度。边(w_i, w_j)上的权重可以表示为$e(w_i, w_j)$，而单词w_i的出度(out degree)表示为$O(w_i) = \sum_{j:w_i \to w_j} e(w_i, w_j)$。

TPR是基于PageRank[15]的算法。PageRank作为Google成功的重要法宝而在研究领域闻名，用于度量网络中节点的重要性。PageRank的基本思想是一个节点的重要性取决于有多少重要的节点指向它，可以看作是网络中节点之间的推荐。在PageRank中，一个词w_i的重要性$R(w_i)$定义为:

$$R(w_i) = \lambda \sum_{j:w_j \to w_i} \frac{e(w_j, w_i)}{O(w_j)} R(w_j) + (1 - \lambda) \frac{1}{|V|} \tag{4-1}$$

其中λ是一个衰减因子(damping factor)，取值范围是$[0, 1]$，$|V|$表示网络中的节点个数。衰减因子表示每个节点有$(1 - \lambda)$的概率随机跳转到网络中的其他节点，而有λ的概率随边跳转到该节点的邻居节点。PageRank值是通过迭代运行公式(4-1)直到收敛后得到的。在公式(4-1)中的第二部分，也就是随机跳转部分，可以看作是对PageRank的一种平滑，可以使图满足非周期和不可约的条件，从而可以使PageRank能够收敛到稳定状态。在PageRank中，不同节点的第二部分经常设为相同的值$\frac{1}{|V|}$，这表示每个节点等概率地随机跳转到其他节点，没有任何偏好。

实际上，PageRank在公式(4-1)的第二部分也可以设置为非均匀分布(non-uniformed)。假如对某些节点设置一个较大的概率，最终的PageRank将更加偏向于这些节点。这种PageRank被称为带偏好的PageRank(Biased PageRank)。

Topical PageRank (TPR)的思想就是为每个隐含主题单独的运行带偏好的PageRank[105,106]。每个主题相关的PageRank都会偏好那些与该主题有较大语义相关度的单词。这个偏好就是通过设置随机跳转概率来实现的。

对每个主题z，可以为每个词设置偏好值(preference value)$\Pr_z(w)$作为该词的随机跳转概率，并保证$\sum_{w \in V} p_z(w) = 1$。这样那些与主题$z$相关的词将被赋予较大的偏好，在运行PageRank时，将有更大的概率被访问。对于主题z，带偏好的PageRank值为:

$$R_z(w_i) = \lambda \sum_{j:w_j \to w_i} \frac{e(w_j, w_i)}{O(w_j)} R_z(w_j) + (1 - \lambda) \Pr_z(w_i) \tag{4-2}$$

图 4.1显示了一个包含两个主题的例子。该图用节点的大小表示这个词与该主题的相关程度。在两个主题的PageRank中，不同的词被赋给了较大的偏好值。最后，这些词在不同主题下的PageRank中也得到了不同的PageRank值。

偏好值$\Pr_z(w)$的设置会比较大地影响TPR的效果。这里尝试采用三种不同的方式设置偏好值:

- $\Pr_z(w) = \Pr(w|z)$，表示给定主题z后，单词w的出现概率，这表示w在主题z中所占的比例，也表示主题z有多大程度集中在单词w上。
- $\Pr_z(w) = pr(z|w)$，表示给定单词w后，主题z的出现概率，这表示单词w有多大程度集中在z上。
- $\Pr_z(w) = pr(w|z) \times pr(z|w)$，是以上两种概率的乘积，综合了两个方面的考虑，这个度量受到了[107]的影响。

PageRank和TPR都是迭代式的算法。终止算法的条件是当迭代次数达到100次，或者两次相邻的迭代每个词的PageRank值的差别小于0.001。

4.2.3　利用主题相关的重要性进行关键词抽取

在获取每个词的排序情况后，对候选关键词进行排序。根据已有工作[80]的报告，大部分关键词都是名词短语，所以选取文档中的名词性短语作为候选关键词。

文档候选关键词的获取方法如下：首先对文档进行预处理，并标注词性(part-of-speech，POS)。在实验中采用Stanford POS Tagger① 进行标注，所采用的标注模型是left3words-distsim-wsj。然后根据正则表达式`(adjective)*(noun)+`选取关键词，这个正则表达式意义是由0个或者多个形容词跟随1个或者多个名词组成的词串。本章将这些词串看作候选关键词。

当选取了这些候选关键词后，算法根据TPR得到的单词的重要性对这些候选关键词进行排序。在PageRank中，一个候选关键词p的重要性是其中每个词的PageRank值之和$R(p) = \sum_{w_i \in p} R(w_i)$[16,19,20]。然后选取排序最高的$M$个作为关键词。

TPR首先需要计算每个主题下，候选关键词的重要性:

$$R_z(p) = \sum_{w_i \in p} R_z(w_i) \tag{4-3}$$

然后，考虑文档的主题分布，将候选关键词在不同主题上的重要性总和为它最终

① http://nlp.stanford.edu/software/tagger.shtml。

的重要性。设文档d在主题z上的概率为$\Pr(z|d)$，那么对于每个候选关键词p，可以计算它在文档中的重要性：

$$R(p) = \sum_{z=1}^{K} R_z(p) \times \Pr(z|d) \tag{4-4}$$

根据每个候选关键词的重要性进行排序后，算法选取排序最高的M个作为关键词。

4.3　实验结果与分析

与上一章类似，本章在NEWS和RESEARCH两个数据上进行关键词抽取实验。这里也用两种学习隐含主题模的方式：(1)利用NEWS和RESEARCH数据集合学习主题模型；(2)利用维基百科学习主题模型。

4.3.1　评价指标

实验将标准答案和抽取关键词都抽取它们的词干(stemming)后进行比较，抽取词干的方法是利用Porter Stemmer①，实验采用准确率、召回率和F_1值(precision/recall/F_1-Measure)和另外两种评测指标。可以发现，抽取关键词的排序顺序也表示了关键词抽取效果。一个关键词抽取方法如果能够将正确答案排得越高，那么说明它的效果越好。但是上述的准确率、召回率和F_1值没有办法考虑抽取关键词的顺序。因此，这里额外采用两种评价指标。

其中一个指标称为*binary preference measure* (Bpref)[108]。Bpref是考虑排序顺序的评测指标。对于一个文档，如果在M个抽取的关键词中有R个是标准答案，其中的准确抽取的用r表示，错误抽取的用n表示，那么Bpref通过以下公式计算：

$$\text{Bpref} = \frac{1}{R} \sum_{r \in R} 1 - \frac{|n \text{ ranked higher than } r|}{M} \tag{4-5}$$

另外一个方法称为*mean reciprocal rank* (MRR)[109]，用来度量每个文档第一个被准确推荐的关键词的排名情况，对于一个文档d，用$rank_d$来表示第一个被准确推荐关键词的排名位置，那么MRR定义为：

$$\text{MRR} = \frac{1}{|D|} \sum_{d \in D} \frac{1}{rank_d} \tag{4-6}$$

其中D是进行关键词抽取测试的文档集合。

① http://tartarus.org/~martin/PorterStemmer。

表 4.1　当推荐$M = 10$个关键词时，滑动窗口大小W对NEWS上关键词抽取的影响。

W	Precision	Recall	F_1-Measure	Bpref	MRR
5	0.280	0.345	0.309	0.213	0.636
10	0.282	0.348	0.312	0.214	0.638
15	0.282	0.347	0.311	0.214	0.646
20	0.284	0.350	0.313	0.215	0.644

4.3.2　参数对于TPR的影响

TPR中有四个参数可能会对关键词抽取的效果产生影响，分别是：

- 滑动窗口大小W；
- LDA的主题个数K；
- 不同的设置偏好值的方法$\Pr_z(w)$；
- TPR中的衰减因子λ。

这里将分别考察这些参数对TPR进行关键词抽取结果的影响。除了被考察的参数，其他参数设置为$W = 10, K = 1,000, \lambda = 0.3$以及$\Pr_z(w) = \Pr(z|w)$，这些都是TPR在NEWS和RESEARCH上获得最好结果的参数设置。

4.3.2.1　滑动窗口大小W

在NEWS的实验中，在表 4.1中展示TPR进行关键词抽取的效果随滑动窗口W从5到20的变化，可以发现效果的变化与论文[19]中的情形类似，也就是当滑动窗口变化的时候，算法效果变化不大，而且当W适中的时候，如$W = 10$或$W = 15$，算法效果达到最优。

类似的，在RESEARCH数据上，当W从2到10变化时，TPR效果也变化不大。但是可以发现TPR在RESEARCH上当$W = 20$的时候效果变差。这个现象的原因在于RESEARCH本身较短，每篇摘要平均仅有121个词，要远短于NEWS的704个词。如果在RESEARCH上将滑动窗口大小W设置得过大，建立的图将成为全连通图，因为任意两个节点都有可能相连，这将不能够有效捕捉文档的结构信息。

4.3.2.2　隐含主题模型的主题个数K

图 4.2显示了TPR当隐含主题个数K不同，学习LDA模型的数据集合不同时，在NEWS上的准确率-召回率曲线。可以发现抽取效果并没有随着主题个数变化有明显波动，也没有随着学习LDA模型的训练集合的变化而有剧烈变化。在RESEARCH上表现类似的趋势，这表明TPR能够非常鲁棒地利用LDA获取文档和单词的主题分布，进行关键词抽取。

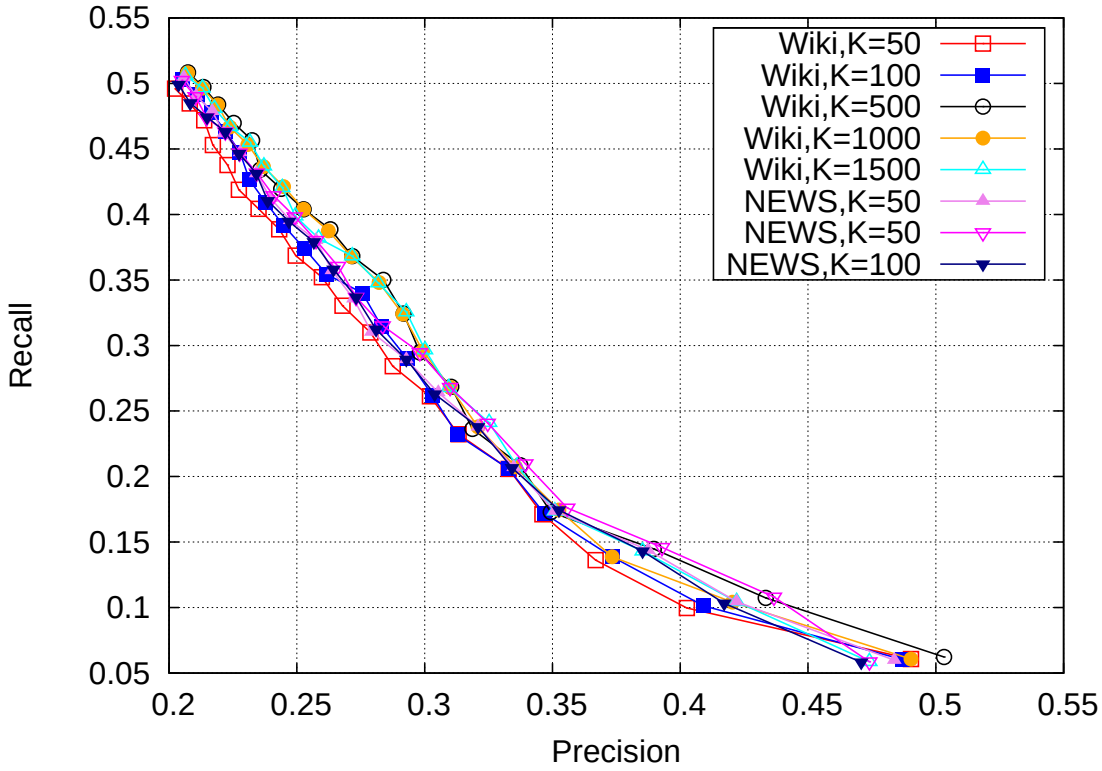


图 4.2　当隐含主题个数K不同，学习LDA模型的数据集合不同时，在NEWS上的准确率-召回率曲线，M取值范围是1到20。

表 4.2　当推荐M = 10个关键词时，LDA主题个数K对NEWS上关键词抽取的影响。

K	Precision	Recall	F_1-Measure	Bpref	MRR
50	0.268	0.330	0.296	0.204	0.632
100	0.276	0.340	0.304	0.208	0.632
500	0.284	0.350	0.313	0.215	0.648
1000	0.282	0.348	0.312	0.214	0.638
1500	0.282	0.348	0.311	0.214	0.631

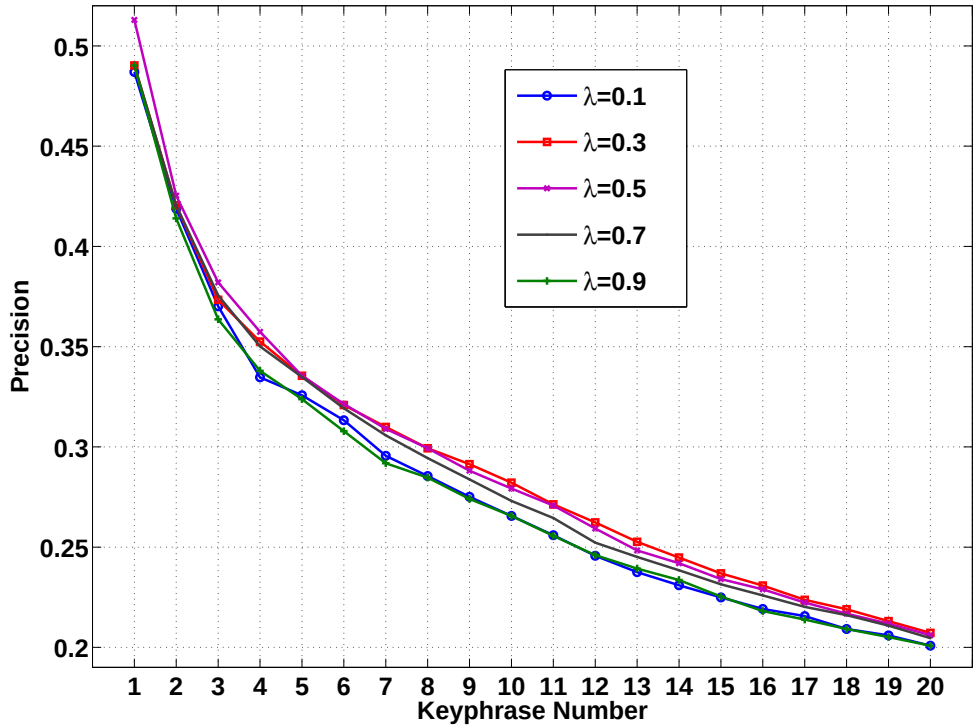


图 4.3　当$\lambda = 0.1, 0.3, 0.5, 0.7, 0.9$，在NEWS上推荐$M$从1到20的准确率变化。

4.3.2.3　衰减因子λ

TPR的衰减因子λ会影响随机游走的过程中，随着边跳转(公式(4-2)中的第一部分)和随机跳转也就是偏好部分(公式(4-2)中的第二部分)的比例。在图 4.3、图 4.4和图 4.5中，实验展示了当$\lambda = 0.1, 0.3, 0.5, 0.7, 0.9$时，推荐1到20个关键词的时候，准确率、召回率和$F_1$值的变化情况。从该图可以看到，当$\lambda$取值为0.2到0.7之间的时候，TPR的效果不错。Bpref和MRR也能够随着λ的变化保持稳定。

4.3.2.4　偏好值$\Pr_z(w)$

最后，实验考察不同的偏好值设置$\Pr_z(w)$对TPR进行关键词抽取的影响，偏好值的设置如公式(4-2)所示。表 4.3展示了在NEWS上推荐$M = 10$的时候，不同偏好值对抽取效果的影响。从该表可以发现$\Pr(z|w)$表现最佳，这与在RESEARCH上的观察类似。

在关键词抽取任务中，需要发现能够准确表示文档主题的关键词。因此，并不需要在许多主题中都出现的那种常用词。而$\Pr(w|z)$通常会根据一个词在这个主题下出现的频度给予偏好值，因此，那些常用词会在很多主题上都得到较高的值，

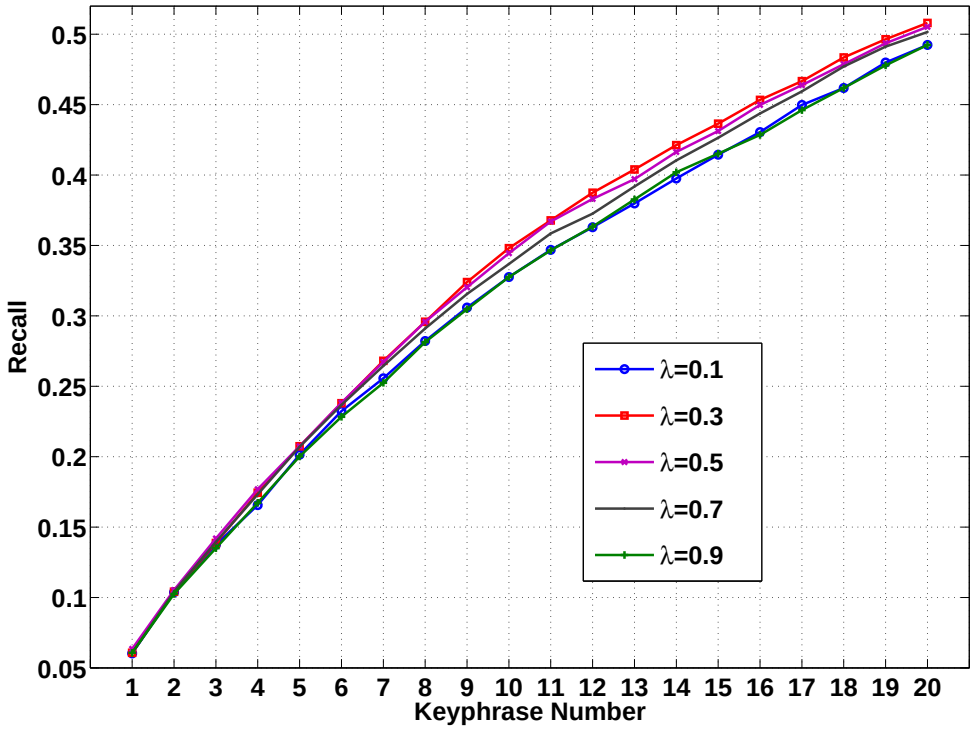


图 4.4　当$\lambda = 0.1, 0.3, 0.5, 0.7, 0.9$，在NEWS上推荐$M$从1到20的召回率变化。

表 4.3　在NEWS上推荐$M = 10$的时候，偏好值设置对TPR关键词抽取的影响。

偏好值	Precision	Recall	F_1-Measure	Bpref	MRR
Pr(w|z)	0.256	0.316	0.283	0.192	0.584
Pr(z|w)	0.282	0.348	0.312	0.214	0.638
prod	0.259	0.320	0.286	0.193	0.587

从而在最终的排序中拔得头筹。与此不同，Pr(z|w)则偏好那些专注于该主题的单词，使用Pr(z|w)来设置偏好值，能够抽取出那些与主题密切相关的关键词。

4.3.3　与其他方法的比较

通过在NEWS和RESEARCH上系统考察过不同参数对TPR进行关键词抽取的影响后，这里开始与TFIDF，TextRank和LDA比较关键词抽取的效果。

TFIDF根据每个词在文档中的TFIDF值来计算其重要性，即$R(w) = tf_w \times \log(idf_w)$。而在TextRank中，每个词的重要性是通过公式(4-1)来度量的，这两种方式都没有采用主题信息。

LDA则是通过文档和单词的主题分布相似度来计算单词的重要性。给定文档d和单词w，有很多方法计算他们之间的相似度，如余弦相似度、预测似然度以

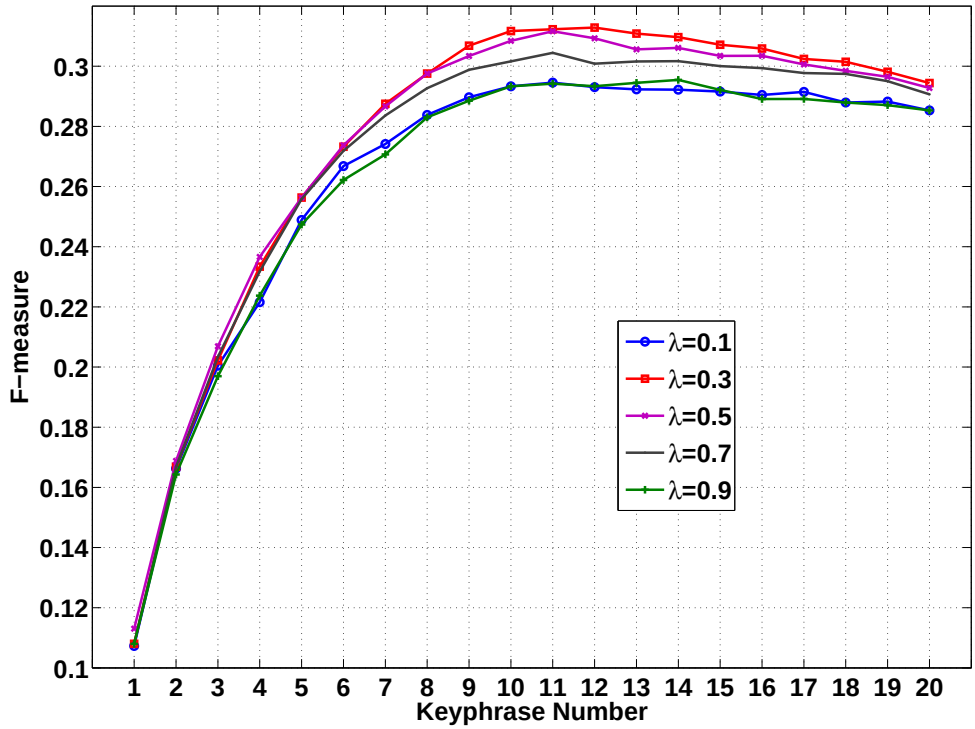


图 4.5　当$\lambda = 0.1, 0.3, 0.5, 0.7, 0.9$，在NEWS上推荐$M$从1到20的$F_1$值变化。

表 4.4　在NEWS上推荐$M = 10$个关键词的时候的不同方法的效果。

方法	Precision	Recall	F_1-Measure	Bpref	MRR
TFIDF	0.239	0.295	0.264	0.179	0.576
TextRank	0.242	0.299	0.267	0.184	0.564
LDA	0.267	0.329	0.294	0.200	0.558
TPR	**0.282**	**0.348**	**0.312**	**0.214**	**0.638**

及KL距离[103]等。这里仅展示在上一章中分别在NEWS和RESEARCH表现最优的结果，即在Wikipedia上训练LDA模型、设置$K = 1500$、并采用预测似然相似度的结果。

表 4.4和表 4.5比较了在NEWS和RESEARCH上不同方法进行关键词抽取的效果。由于在NEWS上平均每篇文档人工标注的关键词个数要多于RESEARCH，因此对NEWS设置抽取$M = 10$个关键词，而对RESEARCH设置抽取$M = 5$个关键词。在NEWS和RESEARCH的参数设置已经在第 4.3.2节描述过。

从这两个表格，实验有以下观察：

1. 首先，TPR在两个数据集合上都优于其他所有的方法。该提高通过了95%置信度的显著性检验。这表明TPR进行关键词抽取的有效性和鲁棒性。

表 4.5　在RESEARCH上推荐$M = 5$个关键词的时候的不同方法的效果。

方法	Precision	Recall	F_1-Measure	Bpref	MRR
TFIDF	0.333	0.173	0.227	0.255	0.565
TextRank	0.330	0.171	0.225	0.263	0.575
LDA	0.349	0.181	0.238	0.225	0.571
TPR	**0.354**	**0.183**	**0.242**	**0.274**	**0.583**

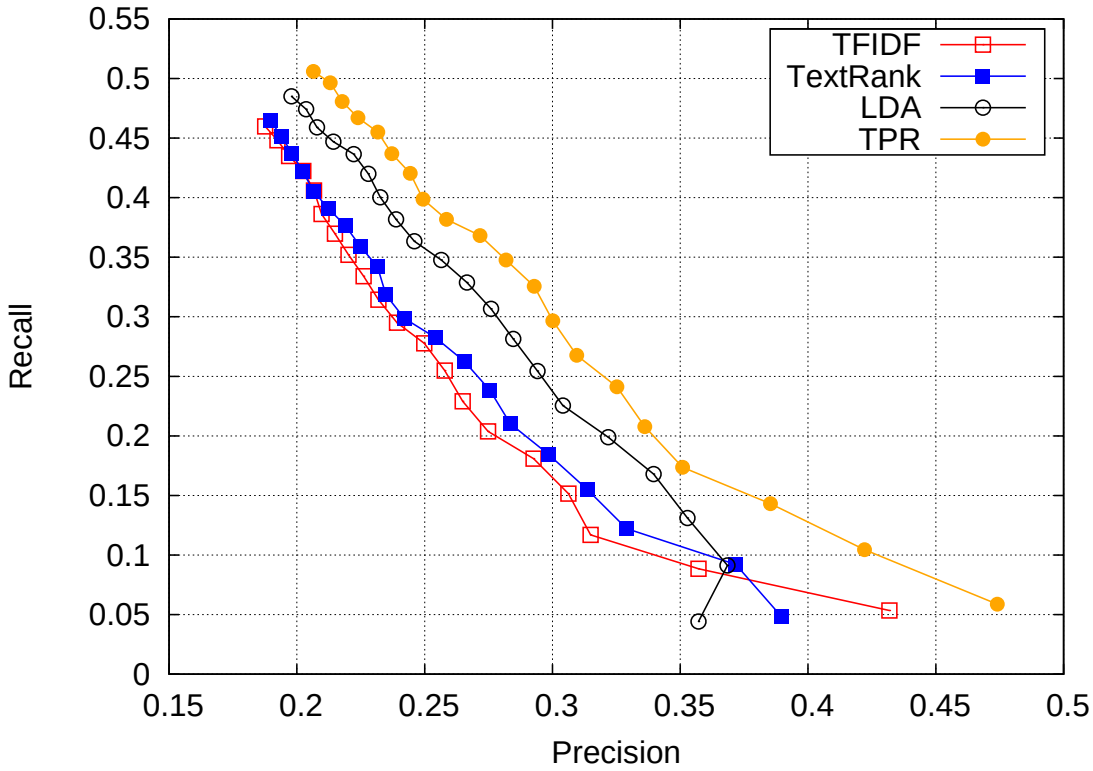


图 4.6　在NEWS上的准确率-召回率曲线，M取值范围是1到20。

2. 其次，在评价指标准确率、召回率和F_1值上，LDA的表现优于TFIDF和TextRank。但是LDA的MRR值则要低于TFIDF和TextRank，这说明LDA没有办法很好地准确抽取第一个关键词，原因主要在于：(1) LDA没有考虑文档的结构信息，如TextRank那样；(2)LDA也没有考虑单词在文档中的词频信息，如TFIDF那样。而TPR则同时具有LDA和TFIDF/TextRank的优势。

更重要的是，图 4.6和 4.7显示了四个方法在NEWS和RESEARCH上的准确率-召回率曲线，每条曲线上的一个点代表推荐不同的关键词个数M时的评价结果，这条曲线接近于右上方，该方法性能越高。实验证明TPR具有较大的优势。

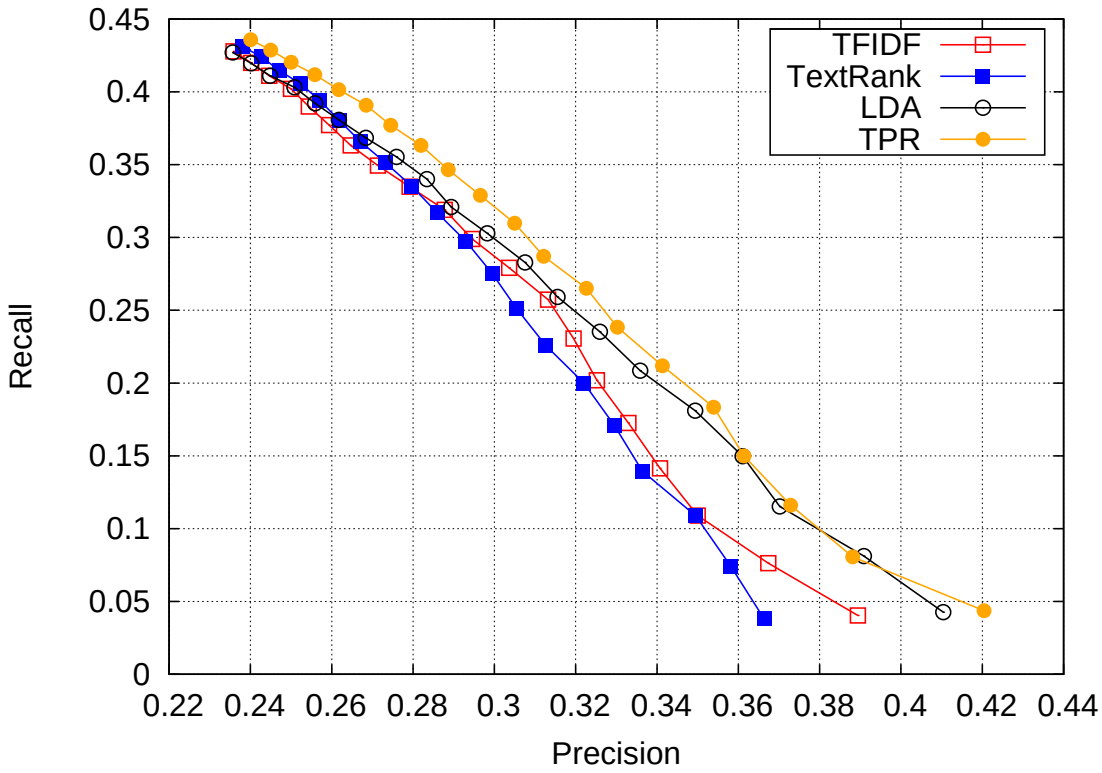


图 4.7　在RESEARCH上的准确率-召回率曲线，M取值范围是1到20。

4.3.4　抽取结果示例

最后，表 4.6给出利用TPR进行关键词抽取的例子。该文档是一篇新闻，标题为“Arafat Says U.S. Threatening to Kill PLO Officials”(DUC2001中的文档号为AP880510-0178)，全文参考附录 A。这里仅列出前10个关键词，其中推荐正确的用“(+)”标识。表中同时还在每个方法后列出它推荐对了多少个关键词，如TPR后的“(+7)”，表示TPR正确推荐了7个关键词。此外还列出了该文档的排名前三的主题以及在该主题下的关键词。很显然，这些主题分别是关于巴勒斯坦(“Palestine”)，以色列(“Israel”)和恐怖主义(“terrorism”)的，这也表明算法推荐的关键词对文档主题有比较好的覆盖度，同时也能够保证关键词之间具有一定的差异性。通过将这些主题相关的关键词融合在一起，算法得到了这篇文章的关键词。

为了更直观地展示TPR算法抽取关键词的性能，下面利用Wordle技术[110]① 对上述示例进行可视化，如图 4.8所示，是TPR对上述示例新闻文档推荐关键词的可视化，图中的关键词的大小与关键词的TPR值相关，TPR值越大的关键词在可视化中越大。图 4.9、图 4.10和图 4.11分别展示了该文档的三个最重要的主题下关键

① 相关可视化工具可以通过http://wordle.net/访问。

表 4.6　TPR抽取关键词示例，文档全文参考附录 A。

TPR (+7)
PLO leader Yasser Arafat(+), Abu Jihad, Khalil Wazir(+), slaying Wazir, political assassination(+), Palestinian guerrillas(+), particulary Palestinian circles, Israeli officials(+), Israeli squad(+), terrorist attacks(+)
TPR，文档主题排名第1的“Palestine”
PLO leader Yasser Arafat(+), United States(+), State Department spokesman Charles Redman, Abu Jihad, U.S. government document, Palestine Liberation Organization leader, political assassination(+), Israeli officials(+), alleged document
TPR，文档主题排名第2的“Israel”
PLO leader Yasser Arafat(+), United States(+), Palestine Liberation Organization leader, Israeli officials(+), U.S. government document, alleged document, Arab government, slaying Wazir, State Department spokesman Charles Redman, Khalil Wazir(+)
TPR，文档主题排名第3的“terrorism”
terrorist attacks(+), PLO leader Yasser Arafat(+), Abu Jihad, United States(+), alleged document, U.S. government document, Palestine Liberation Organization leader, State Department spokesman Charles Redman, political assassination(+), full cooperation

词的排名情况。由这些可视化图例可以直观地看到，TPR不仅可以有效地推荐关键词，而且还能够按照不同的主题对关键词进行排序。这样，如果用户对某个主题的关键词特别感兴趣，TPR还可以推荐某个特定主题下的关键词。

表 4.7显示了其他几个方法抽取的结果。对于TFIDF，它仅仅考虑文档的词频信息，因此会将含有“PLO”的候选关键词排得特别高，因为这个“PLO”出现16次之多，也因此忽略率与主题“Israel”相关的关键词。LDA由于仅通过主题相似度选取关键词，没有考虑文档主题，因此可能会忽略词频较高的关键词，如LDA没有抽取“political assassination”，而单词“assassination”在文档中出现8次之多。

4.4　本章小结

本章针对仅利用文档结构信息进行关键词抽取(如TextRank)和仅利用隐含主题模型进行关键词抽取存在的问题，提出一种综合利用隐含主题模型和文档结构信息的关键词抽取方法，Topical PageRank。该方法是一种基于主题的随机游走模型，在每个主题上运行PageRank，计算词在不同主题下的PageRank值。该方法一方面能够通过隐含主题模型构建文档主题，同时能够通过文档图的随机游走模型

图 4.8　TPR附录 A新闻文档抽取关键词的可视化。

图 4.9　TPR对附录 A新闻文档主题“Palestine”下关键词的可视化。

图 4.10　TPR对附录 A新闻文档主题“Israel”下关键词的可视化。

图 4.11　TPR对附录 A新闻文档主题“terrorism”下关键词的可视化。

表 4.7　其他方法抽取关键词示例。

TFIDF (+5)
PLO leader Yasser Arafat(+), PLO attacks, PLO offices, PLO officials(+), PLO leaders, Abu Jihad, terrorist attacks(+), Khalil Wazir(+), slaying wazir, political assassination(+)
TextRank (+3)
PLO leader Yasser Arafat(+), PLO officials(+), PLO attacks, United States(+), PLO offices, PLO leaders, State Department spokesman Charles Redman, U.S. government document, alleged document, Abu Jihad
LDA (+5)
PLO leader Yasser Arafat(+), Palestine Liberation Organization leader, Khalil Wazir(+), Palestinian guerrillas(+), Abu Jihad, Israeli officials(+), particulary Palestinian circles, Arab government, State Department spokesman Charles Redman, Israeli squad(+)

考虑文档结构为关键词抽取提供信息，实验证明，该方法能够综合隐含主题模型和文档结构信息进行关键词抽取的优势，有效抽取关键词，使关键词对文档主题具有更好的覆盖度。

第 5 章　基于文档与关键词主题一致性的关键词抽取方法 ①

前面三章主要探讨了如何利用文档主题结构进行关键词抽取，提高抽取关键词对文档主题的覆盖度。本章将以文档与关键词的主题一致性作为前提，探讨如何解决文档与关键词的词汇差异问题，更好地度量文档与关键词之间的语义相关性。

5.1　词汇差异问题与文档与关键词的主题一致性

词汇差异是文档和关键词之间的常见现象，具体表现在，很多关键词在文档中出现的次数并不多，无法用传统简单的统计方式(如TFIDF和TextRank)抽取出来。而更严重的情况是，当文档较短的时候，很多关键词甚至都没有在文档中出现过。

那应当如何解决这个问题呢? 已经有一些方案可以在一定程度上缓解词汇差异问题，其大致思路是依赖该文档外部的资源提供更丰富的信息，帮助建立文档与关键词之间的语义映射。

一个典型的方法是ExpandRank[19,20]。这是基于TextRank的算法，基本思想是，当需要给某个文档抽取关键词的时候，在构建单词图的时候，除了考察该文档中词与词的同现关系外，还寻找文档集合中与该文档最相似的k个文档，也将他们中词与词的同现关系加入到该图中。这样可以通过近邻文档中的单词之间的关系，在词网中添加更加丰富和精确的语义信息，从而在一定程度上缓解词汇差异问题。

另外一个典型的方法是采用隐含主题模型进行关键词抽取。这类方法由于是通过主题相似度推荐关键词，避免了传统的基于统计信息的方式推荐关键词的弊端，也能够一定程度上缓解词汇差异问题。

但是这两种方法各自有不可克服的问题。这里可以将ExpandRank看作是在文档层次引入外部信息。当文档集合中含有与该文档高度相关的信息的时候，ExpandRank可以取得较好的效果。但是正是由于引入的过程是以文档为单位的，所以往往会引入噪音。例如当邻居文档虽然也在描述与给定文档相关的主题，但

① 本章主要内容以“Automatic Keyphrase Extraction by Bridging Vocabulary Gap”为题作为学术论文发表在2011年的国际学术会议“The 15th Conference on Computational Natural Language Learning (CoNLL'11)”上。

是也涉及了一些给定文档并未涉及的主题，这些主题也会被当作外部信息引入到词图中，这往往会导致推荐关键词发生主题漂移(topic drift)的现象。而对于隐含主题模型，则可以看作是在主题层次引入外部信息，这种方法往往倾向于推荐在主题中经常出现的词，这样往往会导致推荐的关键词倾向于常用词。

本章提出一种在词汇层次上引入外部信息的方法，也就是利用统计机器翻译的词对齐技术在大规模文档集合上学习文档中的词与关键词之间的语义关系，从而在给定文档和关键词之间建立精确的语义映射，能够推荐更相关的关键词。

作为解决词汇差异问题的方法，机器翻译模型已经被广泛应用于信息检索(information retrieval)[111,112]，图像视频标注(image and video annotation)[113]，问答系统(question answering)[114–118]，查询扩展和重写(query expansion and rewriting)[119–121]，文档摘要(document summarization)[122]，搭配提取(collocation extraction)[123,124]和复述生成(paraphrasing)[125,126]。

本章将考察机器翻译模型的词对齐技术在关键词抽取中的效果，验证其解决文档与关键词的词汇差异问题的有效性。

5.2　基于统计机器翻译词对齐技术的关键词抽取

首先，设文档集合为D，其中每篇文档$d \in D$。关键词抽取的目的是按照给定文档d的似然度对候选关键词进行排序，也就是对所有候选关键词$p \in P$，计算$\Pr(p|d)$，其中P是候选关键词集合。然后，选择前M个作为文档关键词，这里M可以是事先给定的，也可以由系统自己确定。文档d可以看作是一个词的序列$\mathbf{w}_d = \{w_i\}_1^{N_d}$，其中$N_d$是文档$d$的长度。

图 5.1展示了利用词对齐进行关键词抽取的算法流程。本章将算法划分为三个步骤：准备翻译对，训练翻译模型和对给定文档抽取关键词。接下来将分别详细介绍这三步。

5.2.1　准备翻译对

词对齐模型的训练集合需要包含大量的句子级别的翻译对。在关键词抽取任务中，需要准备足够的翻译对来表达文档和关键词之间的翻译关系。这里提出两种为关键词抽取准备翻译对的方式：“文档-标题”翻译对和“文档-摘要”翻译对。

输入： 文档集合D。

1. **准备翻译对：** 对每个文档$d \in D$，有两种准备翻译对的方式：
   - **“文档-标题”翻译对**。使用文档的标题t_d准备翻译对，表示为$\langle D, T\rangle$。
   - **“文档-摘要”翻译对**。使用文档的摘要s_d准备翻译对，表示为$\langle D, S\rangle$。
2. **训练翻译模型：** 给定翻译对，例如$\langle D, T\rangle$，使用词对齐模型训练词到词的翻译模型$\Pr_{\langle D,T\rangle}(t|w)$，其中$w$是文档中的词，$t$是标题中的词。
3. **关键词抽取：** 对给定文档d，根据翻译模型如$\Pr_{\langle D,T\rangle}(t|w)$等进行关键词抽取。
   (a) 度量文档中每个词w的重要性$\Pr(w|d)$；
   (b) 计算每个候选关键词p的重要性：

$$\Pr(p|d) = \sum_{t \in p} \sum_{w \in d} \Pr_{\langle D,T\rangle}(t|w) \Pr(w|d) \tag{5-1}$$

   (c) 根据$\Pr(p|d)$排序结果，选择最高的M个作为文档d的关键词。

图 5.1　利用词对齐模型进行关键词抽取的算法流程。

5.2.1.1　“文档-标题”翻译对

文档的标题往往是对文档内容的简短摘要。大多数情况下的文档，如学术论文、新闻报道等都会有标题。因此，可以利用标题作为对关键词语言的近似，与文档一起构建“文档-标题”翻译对。

词对齐模型假设每个翻译对的长度基本一致。然而，“文档-标题”对的长度往往差别很大，一篇文档的长度要远长于标题。如果直接将这些长度不平衡的翻译对送给词对齐模型学习，往往会造成学习的翻译模型效果较差。因此，这里提出两种方法来解决这种长度不平衡的问题：抽样方法(sampling method)和分割方法(split method)。

抽样方法会对文档中的词进行抽样，从而得到一个新的“文档”，使得它的长度与标题长度相当。假设文档和标题长度分别为N_d和N_t。那么对于文档d，首先建立它的词袋模型$\mathbf{b}_d = \{(w_i, e_i)\}_{i=1}^{W_d}$，其中$W_d$是文档$d$中出现的所有词项(word type)的个数，也就是每个词不管重复出现多少次，只算一次，而e_i表示文档d中词w_i的权重。

本章采用TFIDF(term frequency－inverse document frequency)作为词在文档中的权重。根据$\mathbf{b}_d$进行N_t次带放回的抽样，最后得到一个新的包含N_t个词的词袋来代表文档d。在抽样的结果中，保留了文档d中最重要的词，这样可以在保证不丢失文档主要语义的前提下实现翻译对在长度上的平衡性。在图 5.2中展示了抽样

1: **输入：** 文档d和它的标题t_d。文档长度为L_d，标题长度为L_t。
2: 将文档表示为词袋模型$\mathbf{w}_d = \{(w_i, e_i)\}_{i=1}^{N_d}$，其中$e_i$是词$w_i$在文档$d$中的权重。
3: **for** i from 1 to L_t **do**
4: 　从$\mathbf{w}_d$中根据词的权重带放回地抽取一个词w;
5: 　将词w放入词袋$\mathbf{w'}_d$中。
6: **end for**

图 5.2　抽样方法算法流程。

1: **输入：** 文档d和它的标题t_d。
2: 将文档d分割为多个句子$\mathbf{s}_d = \{s_i\}_{i=1}^{S_d}$，$S_d$表示句子个数。
3: **for** i from 1 to S_d **do**
4: 　计算句子s_i与标题t_d之间的语义相似度$sim(s_i, t_d)$。
5: 　**if** $sim \geq \delta$ **then**
6: 　　将翻译对(s_i, t_d)放入训练集合。
7: 　**else**
8: 　　丢弃s_i。
9: 　**end if**
10: **end for**

图 5.3　分割方法算法流程。

方法算法流程。

分割方法将文档分割为句子，这样每个句子与标题能够保持较为相似的长度。但是并不是所有的句子都与标题存在翻译关系。因此，对于每个句子，计算它与标题之间的语义相似度。有多种方法来计算标题和句子之间的语义相似度。这里选用向量空间模型(vector space model)表示句子和标题，然后利用余弦相似度(cosine similarity)来度量他们之间的相似度。如果相似度大于某个预先定义好的阈值δ，就将其作为翻译对放入训练集合，否则就将这对“句子-标题”丢弃。在图 5.3中，展示了分割方法算法流程。

抽样方法和分割方法各有特点。与分割方法相比，抽样方法会在抽样过程中

丢失一些词汇信息。但是分割方法会产生比抽样方法更多的翻译对(抽样方法的翻译对个数为$O(D)$，而分割方法的翻译对个数为$O(D \times \overline{S_d})$，其中$\overline{S_d}$表示文档集合中平均每个文档的句子个数)，这将使词对齐训练过程更长，在接下来的实验环节，将比较两种方法的效果和效率。

5.2.1.2　“文档-摘要”翻译对

对于大部分学术论文，作者通常会同时提供摘要来概括文档主题。许多新闻文档也会有编辑者提供的摘要。假如每篇文档有摘要信息，算法可以利用摘要和文档构建翻译对进行词对齐学习翻译模型。这里同样可以使用抽样方法或者分割方法来让翻译对的长度具有平衡性。不同之处在于摘要往往也包含多个句子，因此分割方法需要同时对摘要和文档进行分割，选取那些相似度超过阈值的句子对作为翻译对放入训练集合。

5.2.2　机器翻译的词对齐技术

本章主要采用IBM Model-1[127]进行词对齐。IBM Model-1是最为广泛应用的词对齐算法。该模型不需要任何语言学知识，仅仅根据大规模翻译对中的同现关系来学习翻译概率。实验也尝试过使用更复杂高级的词对齐模型，如IBM Model-3等，但是发现这些高级模型并没有得到更好的关键词抽取效果，因此在本章将只介绍和展示IBM Model-1的结果。

不失一般性，在接下来介绍IBM Model-1的时候所使用的翻译对是“文档-标题”对$\langle \mathbf{w}_d, \mathbf{w}_t \rangle$。在IBM Model-1中，每个翻译对$\langle \mathbf{w}_d, \mathbf{w}_t \rangle$中，文档语言$\mathbf{w}_d = \{w_i\}_{i=0}^{L_d}$和标题语言$\mathbf{w}_t = \{t_i\}_{i=0}^{L_t}$之间的翻译关系是通过隐含变量$\mathbf{a} = \{a_i\}_{i=1}^{L_d}$来描述的。该变量代表了从文档中的词到标题中的词的映射关系：

$$\Pr(\mathbf{w}_d|\mathbf{w}_t) = \sum_{\mathbf{a}} \Pr(\mathbf{w}_d, \mathbf{a}|\mathbf{w}_t) \tag{5-2}$$

例如$a_j = i$表示文档$\mathbf{w}_d$中在位置j的词w_j被对齐到了标题$\mathbf{w}_t$中在位置i的词$\mathbf{w}_t$。对齐变量$\mathbf{a}$还包括一个对到“空词”(empty word)的对齐$a_j = 0$，表示文档中在位置j的词w_j被对其到一个空的词上。IBM Model-1可以利用期望最大化算法(Expectation-Maximization，EM)[128]进行无监督训练。利用IBM Model-1可以得到两种语言词汇的翻译概率，也就是$\Pr(w_t|w_d)$和$\Pr(w_d|w_t)$，其中w_d是文档词汇，而w_t是标题词汇。IBM Model-1的训练过程如图 5.4所示。

IBM Model-1将会产生从一种语言到另外一种语言的“一对多”的对齐，所以学习得到的翻译模型是非对称的。也就是说利用“文档-标题”对进行学习和利

输入： 翻译句对d和t
输出： 翻译概率$\Pr(w_t|w_d)$
按照均匀分布初始化$\Pr(w_t|w_d)$
while 没有满足收敛条件 **do**
　对所有的t和w设$count(w_t|w_d) = 0$
　对所有的w设$count(w_d) = 0$
　for 每个翻译对(d, t) **do**
　　for t中的每个词w_t **do**
　　设$s - total(t) = 0$
　　for d中的每个词w_d **do**
　　设$s - total(w_t) += t(w_t|w_d)$
　　end for
　　end for
　　for t中的每个词w_t **do**
　　for d中的每个词w_d **do**
　　设$count(w_t|w_d) += \frac{\Pr(w_t|w_d)}{s-total(w_t)}$
　　设$count(w_d) += \frac{\Pr(w_t|w_d)}{s-total(w_t)}$
　　end for
　　end for
　end for
　for 文档中的每个词w_d **do**
　　for 标题中的每个词w_t **do**
　　计算 $\Pr(w_t|w_d) = \frac{count(w_t|w_d)}{w_d}$
　　end for
　end for
end while

图 5.4　词对齐模型IBM Model-1的训练过程[129]。

用“标题”-文档”对进行学习，得到的翻译概率是不同的。因此可以利用同一个翻译对语料库学习得到两种翻译概率，一种是利用(文档→标题)学习得到的，另外一种是利用(标题→文档)学习得到的。这里将前一种翻译模型表示为$\mathrm{Pr}_{\mathrm{d2t}}$，将后一种翻译模型表示为$\mathrm{Pr}_{\mathrm{t2d}}$。算法定义$\mathrm{Pr}_{\langle D,T\rangle}(t|w)$为两种翻译模型的调和平均：

$$\mathrm{Pr}_{\langle D,T\rangle}(t|w) \propto \left(\frac{\lambda}{\mathrm{Pr}_{\mathrm{d2t}}(t|w)} + \frac{(1-\lambda)}{\mathrm{Pr}_{\mathrm{t2d}}(t|w)}\right)^{-1}, \tag{5-3}$$

其中λ是调和系数，用来融合两种翻译模型，当$\lambda = 1.0$或者 $\lambda = 0.0$的时候，这个翻译模型将仅使用$\mathrm{Pr}_{\mathrm{d2t}}$或者$\mathrm{Pr}_{\mathrm{t2d}}$。利用这个翻译模型$\mathrm{Pr}(t|w)$，能够将存在词汇差异的文档和关键词建立语义上的映射。

5.2.3　基于词对齐技术的关键词抽取方法

给定文档d，通过计算似然度$\mathrm{Pr}(p|d)$来对候选关键词p进行排序。每个候选关键词p可能包含多个词。如[12]所示，大多数关键词是名词短语。因此，跟随大部分已有工作[16,19,20]，利用词性标注信息选取文档中的名词短语作为候选关键词。对于每个词t，可以计算它给定文档d的似然度$\mathrm{Pr}(t|d) = \sum_{w\in d}\mathrm{Pr}(t|w)\,\mathrm{Pr}(w|d)$，其中$\mathrm{Pr}(w|d)$是文档$d$中词$w$的权重，这里使用归一化的TFIDF值；而$\mathrm{Pr}(t|w)$是词对齐模型学习得到的翻译概率。这里将候选关键词中每个词的值相加得到候选关键词的值，也就是$\mathrm{Pr}(p|d) = \sum_{t\in p}\mathrm{Pr}(t|d)$。总之，候选关键词的值可以通过图 5.1中的公式(5-1)计算。根据排名，选取最高的M个作为关键词。

5.3　实验结果与分析

为了验证方法效果，实验从163.com上抓取了13,702篇中文新闻作为文档集合。这些新闻包括了多个主题，如科学、技术、政治、体育、文化、社会和军事等。所有的新闻文档都被编辑者手工标记了关键词，而这些关键词均来自文档正文。每篇新闻也同时提供标题和名为“核心提示”的摘要。

在这个文档集合中，文档有72,900个词项，关键词中有12,405个词项。每篇文档、标题和摘要的平均长度分别为971.7个词、11.6个词和45.8个词。每篇文档平均有2.4个关键词。实验将采用标题和摘要来构建翻译对。

实验选用GIZA++ ① [130]来训练IBM Model-1。GIZA++是最为广泛应用的词对齐工具，它实现了IBM Model-1到IBM Model-5以及一个HMM词对齐模型。

① http://code.google.com/p/giza-pp/

为了验证方法的效果，实验采用163.com为新闻标注的关键词作为标准答案。如果一个系统推荐的关键词与标准答案完全匹配，则认为该关键词推荐正确。这里同样采用准确率、召回率和F_1进行评价。实验结果都是采用5等分交叉检验(5-fold cross validation)得到的。

5.3.1　关键词抽取效果评价

5.3.1.1　比较与分析

这里采用四种典型的无监督方法与基于词对齐的关键词抽取方法进行比较。这四种方法分别是：TFIDF，TextRank[81]，ExpandRank[19,20]和LDA[65]。为了进行比较，接下来将基于词对齐的方法表示为WAM。

图 5.5展示了不同关键词抽取方法的准确率-召回率曲线图，包括：TFIDF，TextRank，ExpandRank，LDA，基于“文档-标题”对和抽样方法的WAM(Title-Sa)，基于“文档-标题”对和分割方法的WAM(Title-Sp)，基于“文档-摘要”对和抽样方法的WAM(Summ-Sa)，以及基于“文档-摘要”对和分割方法的WAM(Summ-Sp)。对于WAM，设置调和系数$\lambda = 1.0$，阈值$\delta = 0.1$，这是得到最优结果的参数。这些参数对关键词抽取效果的影响将稍后展示。对于TextRank，LDA和ExpandRank，实验均汇报其经参数调整后的最优结果。例如LDA中隐含主题个数为$K = 400$，ExpandRank中近邻文档数设置为$k = 5$。

在一条准确率-召回率曲线上，每个点分别表示推荐不同数目关键词时的评价结果，从右下方法的$M = 1$到左上方的$M = 10$。一条准确率-召回率曲线越靠近右上方，就说明这个方法的效果总体越好。表 5.1同时展示了不同方法在推荐$M = 2$个关键词的时候的准确率、召回率和F_1值。这里选择$M = 2$是因为这个文档集合上平均每篇文档的关键词数目是2.4，而且WAM在$M = 2$的时候F_1值最高。表 5.1还在F_1值±后面展示了5等分之间的方差。从图 5.5和表 5.1可以得到以下结论：

首先，基于词对齐的关键词抽取方法要优于其他所有方法，这表明从翻译的角度看待文档-关键词的关系是可行的。当面对词汇差异问题的时候，TFIDF和TextRank基本束手无策。ExpandRank采取的策略是从文档层面利用外部信息，这会引入许多噪音。LDA则是从主题的层面利用外部信息，隐含主题往往粒度较粗。与这些方法不同，基于词对齐的方法在词语的层次上利用外部信息在存在差异的词汇之间建立映射关系，可以有效地避免主题漂移问题。因此，基于词对齐的关键词抽取方法能够更好地解决关键词的词汇差异问题。

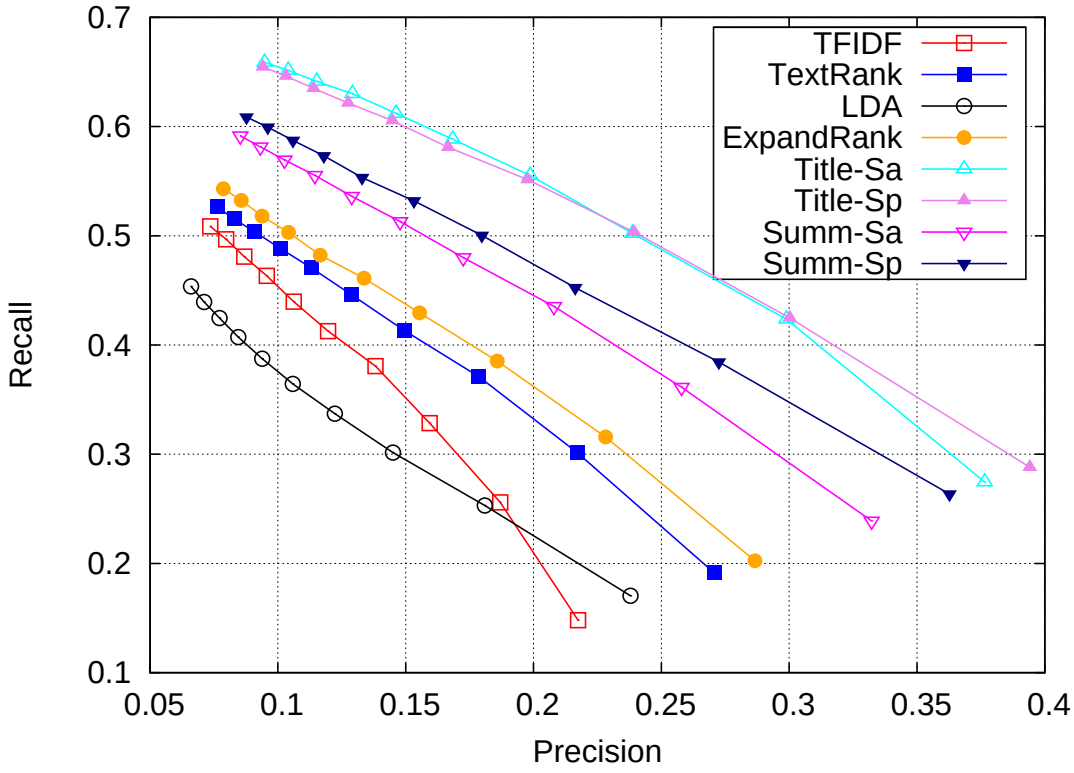


图 5.5　不同关键词抽取方法的准确率-召回率曲线。

表 5.1　当抽取$M = 2$个关键词时，不同方法的准确率、召回率和F_1值。

方法	Precision	Recall	F_1-Measure
TFIDF	0.187	0.256	0.208±0.005
TextRank	0.217	0.301	0.243±0.008
LDA	0.181	0.253	0.203±0.002
ExpandRank	0.228	0.316	0.255±0.007
Title-Sa	0.299	0.424	0.337±0.008
Title-Sp	**0.300**	**0.425**	**0.339±0.010**
Summ-Sa	0.258	0.361	0.289±0.009
Summ-Sp	0.273	0.384	0.307±0.008

第二，无论是基于抽样方法还是分割方法，“文档-标题”翻译对学习的翻译模型比“文档-摘要”翻译对学习的翻译模型效果要好。这说明标题比摘要更接近关键词语言，这也与人们的直观相符合，因为标题往往比摘要更重要和精炼。同时，利用“文档-标题”对也能够节省翻译模型的训练时间。

最后，分割方法比抽样方法效果要优。这个原因在于分割方法能够产生比抽样方法更多的有效翻译对。

5.3.1.2　参数的影响

现在考察一下参数对基于词对齐的关键词抽取的影响。这里以取得最好效果的基于“文档-标题”翻译对和利用分割方法为例进行考察。基于词对齐的关键词抽取的主要参数包括：调和因子λ(参见公式(5-3)和阈值δ。

调和因子λ控制了两个方向的翻译对训练的翻译模型的贡献比例，也就是公式(5-3)中的$\Pr_{d2t}(t|w)$和$\Pr_{t2d}(t|w)$。而阈值δ则是用来去除相似度较小的句子对。

图 5.6展示了不同调和因子下的准确率-召回率曲线，调和因子的取值为从0.0到1.0，步长是0.2，从该图可以看到，翻译模型$\Pr_{d2t}(t|w)$ (也就是当$\lambda = 1.0$时)的效果要明显好于$\Pr_{t2d}(t|w)$ (也就是当$\lambda = 0.0$时)。这说明，仅仅采用一个方向上的翻译模型 $\Pr_{d2t}(t|w)$就可以达到最优的效果。

图 5.7展示了阈值δ从0.01到0.90变化时的准确率-召回率曲线。在利用分割方法将“文档-标题”翻译对构成训练集合时，如果没有过滤任何“句子-标题”对(也就是$\delta = 0$)，翻译对总数是$347,188$。而当$\delta = 0.01$，0.10和0.90时，保留下来的翻译对分别是$165,023$，$148,605$和$41,203$。从图 5.7可以看到更多的翻译对会得到更好的关键词抽取效果。但是，更多的翻译对也会造成更长的词对齐训练时间。幸运的是，可以看到，即使过滤较多的对，抽取的效果并没有因此下降很多。即使当$\delta = 0.9$，WAM依然能够达到准确率0.277，召回率0.391，F_1值是0.312的较好效果(当推荐关键词个数$M = 2$时的评价结果)。但这时相比起$\delta = 0.01$，算法过滤掉了大约50%的翻译对。

综上对两个参数的分析，说明基于词对齐进行关键词抽取能够有效地建立文档与关键词之间的映射，并且对参数具有较高的鲁棒性。

5.3.1.3　当标题或摘要不存在的情况

在某些特殊情况下，文档的标题和摘要可能无法获取，那么应当如何构建翻译对呢? 受到基于句子抽取的文档摘要研究的启发[81,131]，可以在文档内抽取一句或者多句重要的句子来与文档构建翻译对。基于重要句子抽取的无监督文档摘

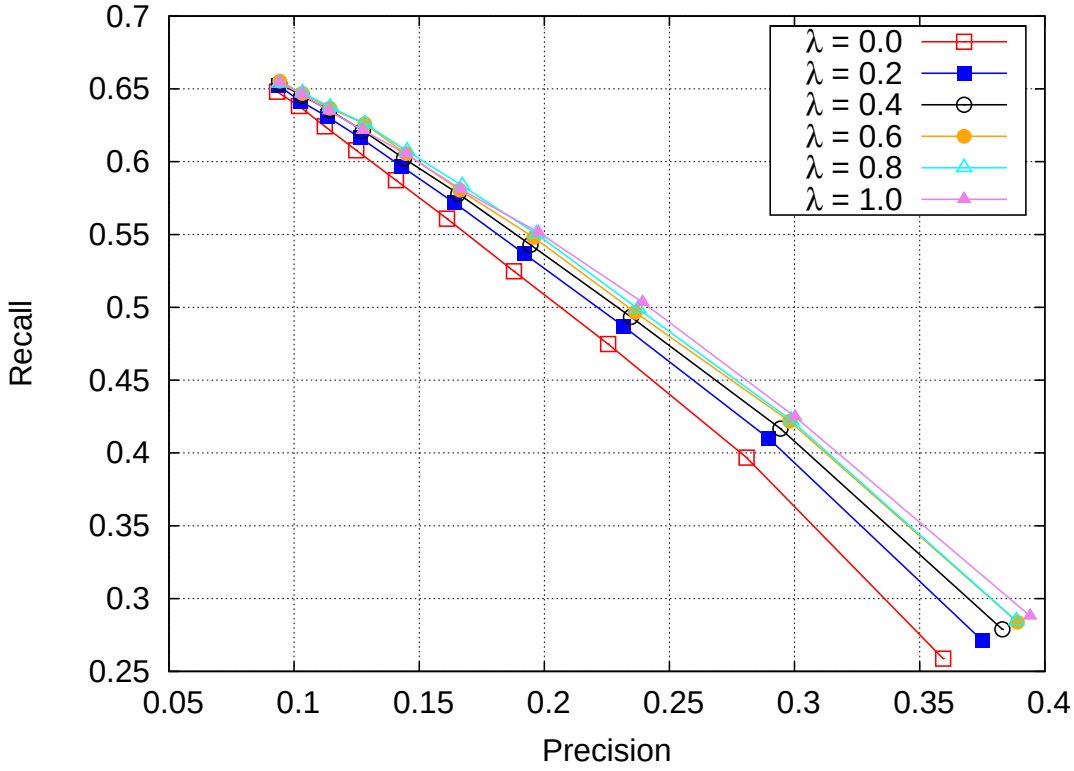


图 5.6　调和因子λ从0.0到1.0变化时的准确率-召回率曲线。

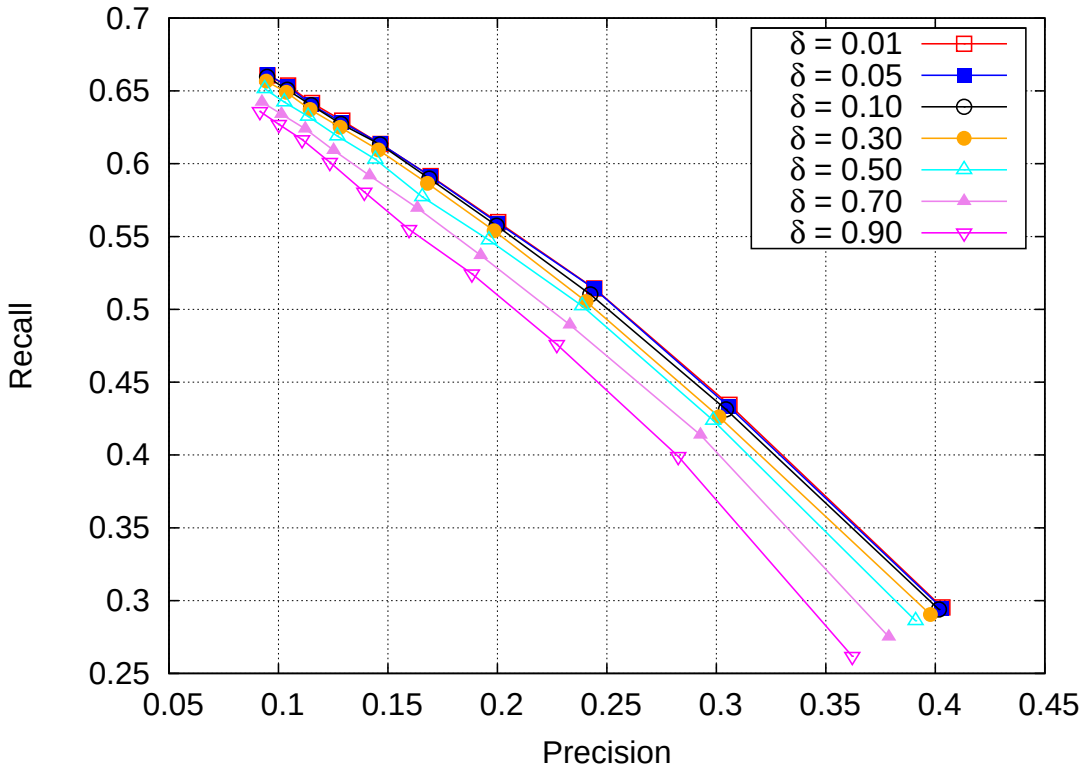


图 5.7　阈值δ从0.01到0.90变化时的准确率-召回率曲线。

表 5.2　当推荐$M = 2$个关键词时，使用两种方法抽取文档一句话来构建翻译对进行基于词对齐的关键词抽取的准确率、召回率和F_1值。

方法	Precision	Recall	F_1-Measure
First	0.290	0.410	0.327±0.013
Importance	0.260	0.367	0.293±0.010

要是自然语言处理中的重要研究任务。在本章，如表 5.2所示，仅采用两种最基本同时也是最鲁棒的方法抽取关键句子:

1. 选择文档的第一句作为文档的关键句子(在表格中表示为“First”);
2. 计算每句话与文档的余弦相似度(在表格中表示为“Importance”)。

实验发现一个有趣的现象：利用文档的第一句的效果与前面利用标题的效果非常接近。这是受到新闻文档一般会在新闻的第一句话概括整个新闻的习惯的影响。虽然第二种方法的效果没有前者好，但是它的效果依然要比其他的已有方法要好。更重要的是，如果能够寻找更有效的方式来寻找文档中的关键句子，第二种方法的效果应该会进一步提高。

5.3.2　关键词生成效果评价

第 5.3.1节在关键词抽取上评价了各种方法的效果，分析了基于词对齐方法的有效性。实际上，基于词对齐的方法有能力推荐并没有在文档出现的关键词。这种能力在短文档上尤其有效，因为短文档所包含的文本信息往往极为有限。这里称这种任务为**关键词生成**(keyphrase generation)。为了评价关键词生成上的效果，接下来选择根据文档的标题来推荐关键词，并且不限制关键词一定要在标题中出现。该实验的训练过程与 5.3.1一样，都是利用文档与标题构建翻译对进行翻译模型的训练，但是在测试的时候，仅能够根据文档的标题进行关键词生成。需要注意的是，LDA和ExpandRank与词对齐方法类似，都能够推荐没有在文档正文出现的关键词。这里依然使用标题所在的文档所对应的关键词作为标准答案对不同方法进行评价。在这种情况下，大约有59%的标准关键词没有在标题中出现。

如表 5.3所示，是不同方法在推荐$M = 2$个关键词的时候的评价结果。对于词对齐方法，这里仅展示利用“文档-标题”对和分割方法的结果。从该表格可以得到以下结论:

1. 基于词对齐的方法在关键词生成上的效果明显优于其他方法。更重要的是，大约有10%的准确推荐的关键词没有在标题中出现，这说明了词对齐方法进行关键词生成的有效性。
2. TFIDF和TextRank的效果要远逊于表 5.1中关键词抽取的效果，这时因为标

表 5.3　当推荐$M = 2$个关键词时，基于词对齐的关键词生成的准确率、召回率和F_1值。

方法	Precision	Recall	F_1-Measure
TFIDF	0.105	0.141	0.115±0.004
TextRank	0.107	0.144	0.118±0.005
LDA	0.180	0.256	0.204±0.008
ExpandRank	0.194	0.268	0.216±0.012
WAM	**0.296**	**0.420**	**0.334±0.009**

表 5.4　LDA、ExpandRank和WAM对附录 B新闻文档推荐的排名最高的$M = 5$个关键词。

LDA	伊朗，美国，谈判，以色列，制裁
ExpandRank	伊朗，以色列，黎巴嫩，美国，以军
WAM	伊朗，动武，以军，以色列，核武器
标准答案	核武器，以色列，伊朗

题太短，没有提供足够多的关键词，也没有提供足够多的选择关键词的统计信息。

3. LDA，ExpandRank和基于词对齐的方法都与他们关键词抽取的效果保持基本一致，如表 5.1所示。其中ExpandRank的效果稍有下降。这说明这三种方法都能够进行关键词生成，但也再次证明基于词对齐方法的优越性。

为了展示词对齐方法进行关键词生成的特点，在表 5.4中列出了LDA、ExpandRank和词对齐方法对某篇新闻推荐的前5个关键词。该新闻标题是“以军方称伊朗能造核弹可能据此对伊朗动武”，全文参考附录 B。对该表可以做出以下结论：

1. LDA倾向于推荐比较常用的词，如“谈判”，“制裁”等与这个主题相关的常用词。
2. ExpandRank则会推荐出与此主题无关的词，如“黎巴嫩”等。这是由于引入邻居文档时会同时引入噪音造成的。
3. 基于词对齐的方法能够产生恰当的关键词，比较有效地处理了主题偏移的问题。更重要的是，该方法能够发现“核武器”这样没有出现在标题中的关键词。

5.4　利用词对齐技术的社会标签推荐

以上在关键词抽取和关键词生成任务上验证了词对齐技术解决词汇差异的有效性。不过这两种方法都是在无监督情况下进行的，也就是假设没有人工标注关键词的训练集合，通过学习文档和标题/摘要的翻译关系无监督地对关键词进行排

表 5.5　《基督山伯爵》的描述文档和用户标注的标签(括号内是标注该标签的用户个数)。

描述文档
标题：《基督山伯爵》 简介：《基督山伯爵》是法国小说家大仲马（1802－1870）的名著。一部洋洋一百多万字的小说，居然能让人读得津津有味而不觉冗长，真不容易。一部表现复仇这一不知重复过多少遍的旧主题的通俗小说，居然能历时一百多年长销不衰，更不容易。而这两个“不容易”《基督山伯爵》都做到了，我们不能不叹服大仲马高超的小说艺术，对这部小说刮目相看。《基督山伯爵》写的是水手邓蒂斯即基督山伯爵对迫害他的三个仇人——维尔福、邓格拉斯和弗南，这三人后来分别是司法、金融、政界的头面人物——复仇的故事。由于小说的情节曲折离奇，险象环生，出乎想象之外，又在情理之中，因此它扣人心弦，让读者难以释卷，实在不足为怪。另外，三个各异其趣的复仇故事，写的都是对作恶多端的大人物的胜利，是正义对邪恶的胜利，当然能让经常受气却又经常无奈的普通百姓颇觉舒畅解气。这部小说除了能为读者提供一个极好的猎奇机会，更能让他们在白日梦的逍遥中获得了某种复仇的快感。这样的小说怎能不叫人喜欢呢? 社会学色彩浓厚的评论家会说，这部小说通过写邓蒂斯悲惨经历揭露了法国当时的司法界的黑暗，是一部进步小说。这当然没错。更看重小说艺术本身的评论家会发现，小说的情节安排得曲折离奇、跌宕起伏同时又繁而不乱、环环相扣，充分显示了大仲马作为杰出小说家和剧作家的想象天才和结构能力。从文章学角度看，这是一部营造得天衣无缝的巨匠之作。
标注标签
大仲马(690)，基督山伯爵(474)，小说(439)，外国文学(395)，法国(298)，名著(248)，经典(240)，外国名著(143)

序。

为了验证主题一致性假设的通用性，这里进一步考察利用词对齐技术进行社会标签推荐。在社会标签推荐中，会有较多的标注文档，但是由于社会标签系统中的标注噪音较大，因此存在更严重的词汇差异问题。

表 5.5展示一本书的描述文档和用户标注的标签。社会标签推荐任务如下：给定一个标签标注集合，需要训练一个标签推荐模型，可以对给定新的文档推荐相关标签。

5.4.1　利用词对齐技术进行社会标签推荐

利用词对齐进行标签推荐分为三个主要步骤：

1. **准备文档-标注对**。给定一个标注集合，首先准备文档-标注对，通过词对齐技术训练翻译概率模型。
2. **训练翻译模型**。给定一个文档-标注对的集合，利用IBM Model-1训练翻译模型，学习文档中词与标签之间的翻译概率。
3. **对文档推荐标签**。当学习得到翻译模型后，给一个新的文档，可以根据该文档中的词，通过翻译模型对标签进行排序，推荐排序最高的标签。

可以看到与关键词抽取和关键词生成的不同之处在于如何构建文档-标注对来建立翻译模型：这里面对的是文档和它的标注标签。

在一个标注系统中，对一个对象(如书籍等)所标注的标签有可能多达成千上万个，也有可能只有寥寥几个。这样，就与对象的描述文档的长度有较大的差别，因此，准备文档-标注对的任务就是构建长度均衡的文档-标注对，以保证词对齐模型能够有效地学习翻译概率。

由于一个文档标注的标签没有语法结构等信息，因此，需要对标注标签进行采样来实现与文档的长度相当。采样是由两个参数来控制的。

第一个参数是采样所以来的**标签权重**。这里采用两种方式来衡量标签的权重，一种是标签的频度TF_t，另外一种是标签的TFIDF_t。另外一个参数是采样后，文档与标注标签的**长度比**$\eta = \frac{|\mathbf{w}_d|}{|\mathbf{w}_t|}$，其中$|\mathbf{w}_r|$表示文档中的词的个数，而$|\mathbf{w}_t|$表示所标注标签的个数。

5.4.2　强调文档中出现标签的WAM模型(EWAM)

在上面关键词抽取和关键词生成任务中，大部分关键词还是来自于待分析的文档的。而在社会标签推荐系统中，由于文档和标签的对应关系更加复杂，因此学习得到的翻译模型会更加倾向于推荐没在文档出现过的标签。而实际上，可能在文档出现的标签有更高的概率被用户选中作为标签标注，因此，这里提出一种强调(emphasizing)文档中出现的标签的WAM模型(EWAM)，该方法对每个标签所计算的重要性如以下公式所示：

$$\Pr(t|d) = \sum_{w \in \mathbf{w}_d} (\gamma I_t(w) + (1-\gamma)\Pr(t|w))\Pr(w|d), \tag{5-4}$$

其中$I_t(w)$是个示性函数(indicator function)，当$t = w$时，也就是文档中的这个词与这个标签相同的时候，示性函数取值为1，而当$t \neq w$的时候，示性函数取值为0，而γ则是一个平滑因子，取值范围是$[0.0, 1.0]$。可以看到，当$\gamma = 1.0$的时候，该方法相当于关键词抽取方法，仅仅依赖这些词在文档中的权重来进行排序；而当$\gamma = 0.0$的时候，该方法就是普通的WAM模型。而当γ取$(0.0, 1.0)$之间的某个值

表 5.6　两个数据集合的统计信息。D，W，T，$\bar{N}_w$和$\bar{N}_t$分别表示文档数目，文档中的词项数目，标签中的词项数目，平均每个文档的单词数目以及平均每个标注的标签数。

Data	D	W	T	$\bar{N}_w$	$\bar{N}_t$
BOOK	70,000	174,748	46,150	211.6	3.5
BIBTEX	158,924	91,277	50,847	5.8	2.7

时，该方法就在WAM的基础上强调那些在文档中出现的标签，γ越大，对文档中的标签的强调程度越高。

5.4.3　实验和讨论

5.4.3.1　数据集合和评价指标

实验选取两个真实世界的数据集合验证WAM进行标签推荐的效果。在表 5.6中给出了这两个数据集合的统计信息。

第一个数据集合，命名为BOOK，是从中国最大的图书评论网站豆瓣网[①]抓取的数据，包含图书的介绍文档以及用户标注的标签。第二个数据集合，命名为BIBTEX，是从英文的科学文献网站Bibsonomy[②]获取的数据。这个数据包含了学术论文的介绍文档(主要是标题和简介)以及用户标注的标签。与BOOK不同的是，BIBTEX没有提供一个文档打过某个标签的用户数量信息。

实验采用准确率、召回率和F_1值(precision/recall/F_1-Measure)来评价不同方法的性能。实验结果都是在5交叉检验得到的。在实验中，推荐标签的个数M设置在1到10之间。

5.4.3.2　比较方法

实验选择四种代表算法与WAM进行比较，分别是朴素贝叶斯(Naive Bayes，NB)[13]，k近邻(k nearest neighborhood，kNN)[13], Content Relevance Model(CRM)[69]和Tag Allocation Model(TAM)[132]。

NB和kNN是两个分类算法，一般被用来将标签推荐转换为分类方法来解决，也就是将社会标签看作是分类类别。NB是一个简单的生成模型(generative model)，用来对给定一个文档d产生某个标签t的概率进行建模$\Pr(t|d) \propto \Pr(t)\prod_{w\in d}\Pr(w|t)$，其中$\Pr(t)$可以用标注了标签$t$的比例来估计，而$\Pr(w|t)$则用词$w$在标注了$t$的文档中的频度来估计。$k$NN则是对给定文档，根据向量空间模型(vector space model)下的

① http://www.douban.com/。

② http://www.bibsonomy.org/。该数据可以从http://www.kde.cs.uni-kassel.de/bibsonomy/dumps获取。

余弦相似度寻找数据集合中最相似的k个文档，获取这些文档所标注的标签，按照这些标签被标注的频繁程度排序，然后选择排序最高的推荐给给定的文档[13]。

CRM和TAM是利用主题模型解决标签推荐问题的代表方法。CRM是一个基于LDA的模型，该方法的核心参数是主题个数K。实验测试了不同主题模型个数下的CRM的效果，最终选取它表现最好的$K = 1,024$下的结果展示。TAM也是一个产生式模型，将文档中的词作为主题分配给所标注的标签，实验设置的参数与论文[132]的相同。

这里比较一下这些方法的复杂度。将CRM，TAM和WAM中的训练的迭代次数表示为I，CRM中的主题个数表示为K，那么在训练过程中，NB的复杂度为$O(R\bar{N}_w\bar{N}_t)$，kNN为$O(1)$(因为不需要训练)，TAM为$O(IR\bar{N}_w\bar{N}_t)$，CRM为$O(IKR\bar{N}_w\bar{N}_t)$，而WAM为$O(IR\bar{N}_w\bar{N}_t)$。更精确的，WAM的训练过程包括两部分，准备文档-标注对的复杂度为$O(R\bar{N}_t)$，而词对齐的学习过程复杂度为$O(IR\bar{N}_w\bar{N}_t)$，其中$\bar{N}_t$表示采样后每个文档的平均标签数。而在为一个给定的文档推荐标签时，假设该文档含有N_w个不同的词项，那么NB复杂度为$O(N_wT)$，kNN为$O(R\bar{N}_w\bar{N}_t)$，CRM为$O(IKN_wT)$，TAM为$O(IN_wT)$，而WAM为$O(N_wT)$。从复杂度分析可以看到，WAM的训练和测试的复杂度都相对较小。稍后将会看到WAM的标签推荐效果也非常好，这更加凸显了WAM复杂度较小的难能可贵。

5.4.3.3　实验结果和分析

实验仍然采用GIZA++来作为IBM Model-1的实现训练词对齐模型。与其他方法比较的时候，WAM的参数是，采样时的标签权重为TF-IDF$_t$，长度比为$\eta = 1$，调和因子为$\lambda = 0.5$，文档中词的权重为TF-IDF$_w$。稍后分别考察参数对WAM进行标签推荐的影响。

图 5.8显示了NB，kNN，CRM，TAM和WAM等方法的准确率-召回率曲线图，曲线上的每个点分别对应推荐不同标签数目时的评价结果。从该图可以观察到：

1. 在两个数据集合上，WAM比其他所有方法表现都好。这说明WAM除了在关键词抽取、关键词生成上表现优秀外，在社会标签推荐上也表现出了较大的优势。这说明基于“文档-关键词主题一致性”的假设在标签推荐上也是成立了，也证明了词对齐模型的有效性。
2. WAM在BOOK数据集合上的优势更加明显，这说明WAM能够更好地利用标签的标注次数信息，而其他方法则不能。此外，BIBTEX的文档长度较短，也在一定程度上限制了不同方法能力的发挥。但即使是在BIBTEX上，WAM依然能够表现优良，尤其是当推荐前几个标签的时候，这往往是标签

表 5.7　NB，*k*NN，CRM，TAM和WAM在BOOK数据集合上推荐$M = 3$个标签时的效果比较。

方法	Precision	Recall	F_1-Measure
NB	0.271	0.302	0.247 ± 0.004
*k*NN	0.280	0.314	0.258 ± 0.002
CRM	0.292	0.323	0.266 ± 0.004
TAM	0.310	0.344	0.283 ± 0.001
WAM	0.368	0.452	0.355 ± 0.002

表 5.8　对例子 5.5，NB，*k*NN，CRM，TAM和WAM推荐的标签情况。

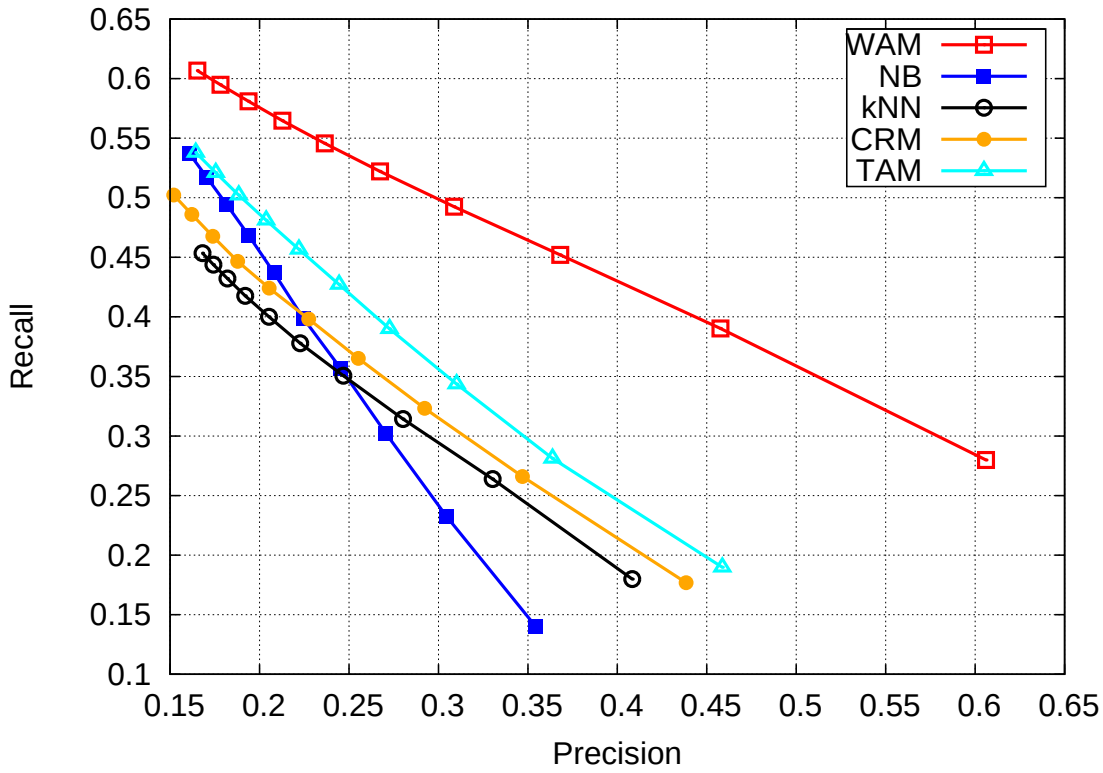
*k*NN	小说，刘恒，文化生活译丛，记忆，最小说，小言，柯艾，青春，围棋
NB	外国文学，文学，历史，日本，经典，法国，哲学，美国，传记
CRM	外国文学，文学，传记，哲学，文化，法国，英国，漫画，历史
TAM	小说，社会学，金融，外国文学，法国，文学，传记，法国文学，漫画，中国
WAM	小说，大仲马，历史，基督山伯爵，外国文学，传记，悬疑，漫画，美国，法国

推荐更加看重的部分。

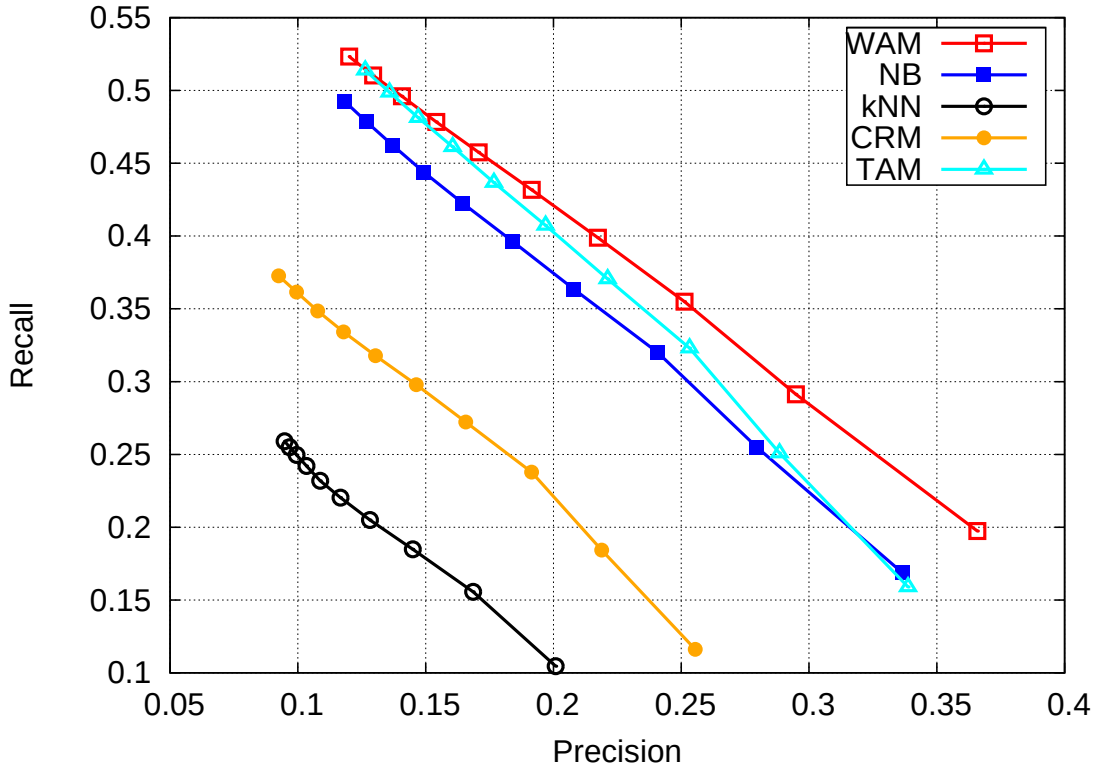
为了更好地展示WAM的效果，在表 5.7中列出了不同方法在BOOK数据集合上推荐$M = 3$个标签时的准确率、召回率和F_1值。之所以选择这个数值，是因为在BOOK数据集上每个文档平均的标签数为3。而实际上WAM在推荐$M = 2$的时候表现最优，此时的F_1值为0.370，远高于CRM的0.263和TAM的0.277。

在表 5.8中给出了NB，*k*NN，CRM，TAM和WAM所推荐的前10个标签。可以看到*k*NN由于无法找到适合的最近邻，因此推荐出来的标签基本无法理解；NB可以推荐出一些相关的标签，但基本都是特别常见的标签；而CRM和TAM与NB的效果类似。WAM的效果最好，可以发现WAM既能够推荐比较粗粒度的标签，如“小说”，“历史”，“外国文学”等，也会推荐出“大仲马”和“基督山伯爵”这样与这本书相关的细粒度的标签。这能够保证WAM更好地满足用户标注的需求。

在表 5.9中列出了表 5.5的文档中最重要的四个词及其他们对应的翻译概率最高的若干标签。在每个标签后的括号中的数值是该标签的翻译概率$\Pr(t|w)$，对于每个词，仅列出翻译概率高于0.1的标签。可以看到这些翻译概率能够有效地按照词与标签的语义相似度将文档和标签建立映射关系。



(a) BOOK



(b) BIBTEX

图 5.8　NB，*k*NN，CRM，TAM和WAM在两个数据上的效果比较。

表 5.9　在表 5.5中的文档中最重要的四个词及其对应的翻译概率最高的若干标签示例。

基督山伯爵	基督山伯爵(0.728)，大仲马(0.270)，...
大仲马	大仲马(0.966)，...
复仇	外国文学(0.168)，经典(0.130)，传记文学(0.123)，大仲马(0.122)，...
法国	法国(0.99)，...

表 5.10　当在BOOK上推荐$M = 3$个标签的时候，采样采用不同的标签权重的影响。

Weighting	Precision	Recall	F_1-Measure
TF_t	0.356	0.437	0.342 ± 0.002
$TF\text{-}IRF_t$	0.368	0.452	0.355 ± 0.002

5.4.3.4　参数的影响

这里将考察不同参数对WAM进行标签推荐的影响。这些参数包括：调和因子，文档-标注长度比，抽样标签权重类型，等。当考察某个参数的时候，设其他的参数都取最优结果时的设置。最后，还将考察不同大小训练集合对算法效果的影响。实验发现WAM在两个数据上表现出类似的趋势，因此这里仅展示在BOOK数据集合上的参数变化的影响。

图 5.9展示了在BOOK数据上，调和因子λ从0.1到1.0变化的时候，WAM的F_1值随着推荐标签数的变化情况。如在关键词抽取中的公式(5-3)所示，调和因子控制了Pr_{d2a}和Pr_{a2d}的影响。

从图 5.9可以观察到，在社会标签推荐任务上，无论是单单哪个方向上的翻译模型都无法取得最好的效果。只有当两者经过调和之后，当调和因子在0.2和0.6之间的时候，标签推荐能够达到最好的效果。这虽然与关键词抽取上观察的效果有所不同，但也看到，即使是利用单方向上的翻译模型进行标签推荐，仍然能够达到较好的效果。

图 5.10显示了文档-标注长度比在BOOK数据集合上的影响。从这个图可以看到，标签推荐的性能对长度比的变化较为鲁棒，除了当$\eta = 10$的时候，因为GIZA++默认接受的长度比范围是$[\frac{1}{9}, 9]$，所以当$\eta = 10$的时候，GIZA++会切断过长的部分从而导致推荐效果变差。

表 5.10展示了当在BOOK上推荐$M = 3$个标签的时候，采样采用不同的标签权重的影响。由于$TF\text{-}IDF_t$倾向于选择语义专注于该文档的词，而不仅仅是常用词，所以它的表现要比TF_t优秀。

实验还考察了不同大小的训练集合对算法效果的影响。图 5.11显示了

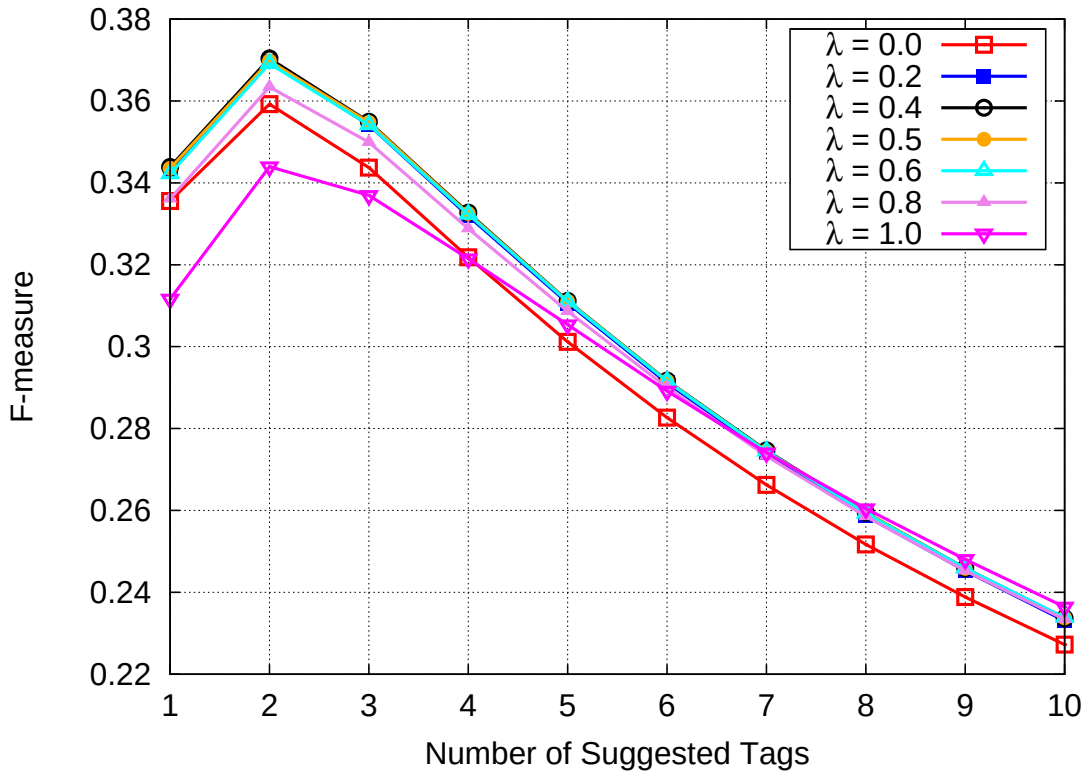


图 5.9　当在BOOK数据上，调和因子λ从0.1到1.0变化的时候，WAM的F_1值随着推荐标签数的变化。

在BOOK数据集合上，当训练集合从8,000增长到56,000时的准确率-召回率曲线图的变化情况。从该图可以观察到:

1. 当训练集合较小的时候(如8,000)，WAM就能够获得不错的推荐效果。
2. 随着训练集合的增大，推荐效果也逐渐变好，但是变好的速度逐渐放慢。这表明WAM并不需要太大的训练集合就能够达到不错的效果。

5.4.3.5　EWAM的效果

最后，实验考察了EWAM对标签推荐的效果。EWAM在BOOK和BIBTEX上表现了不同的变化。在BOOK数据集合上，EWAM的准确率、召回率和F_1值分别为0.385，0.472和0.371 ± 0.001，比WAM的效果有较大提高。而在BIBTEX上，EWAM的效果则发生了下降，其F_1值只有0.229。这表明，EWAM不是一个普适的策略。在BIBTEX上效果之所以变差的原因可能是，BIBTEX的文档较短，无法有效地确定应当强调哪些标签，这样会导致强调了错误的标签，从而把本来正确推荐的标签排除在外了。

该实验的结论是：必须在合适的情形下才能强调文档出现的标签。目前来看，

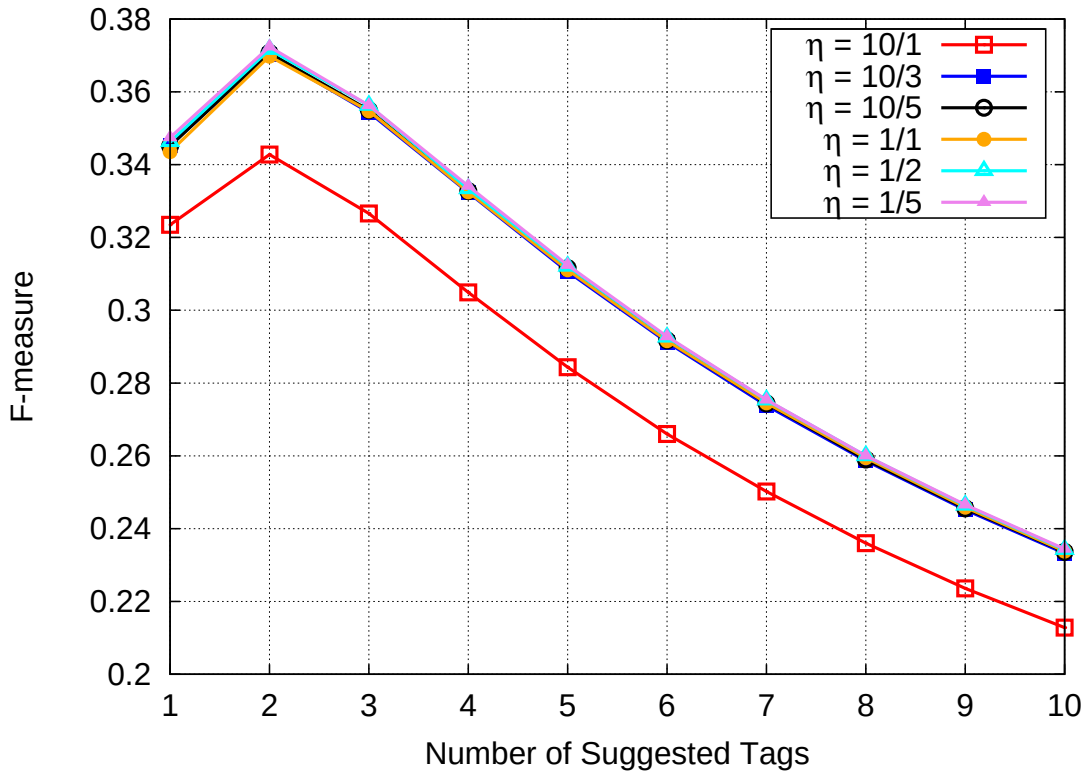


图 5.10　当在BOOK数据上，文档-标注长度比η从10/1到1/5变化的时候，WAM的F_1值随着推荐标签数的变化。

至少需要文档能够提供足够的信息正确决定要强调的标签。

5.5　本章小结

本章提出文档和关键词之间的词汇差异问题。针对这个挑战，本章提出了基于文档与关键词主题一致性的假设，并以此为前提，提出采用统计机器翻译中的词对齐模型来建立文档中的词和关键词之间的语义关系，并以此为基础进行关键词抽取、关键词生成和标签推荐。在关键词抽取、关键词生成和社会标签推荐等任务上的实验验证了该假设，并证明了基于词对齐模型能够有效地在文档和关键词之间建立语义映射，有助于解决词汇差异问题。

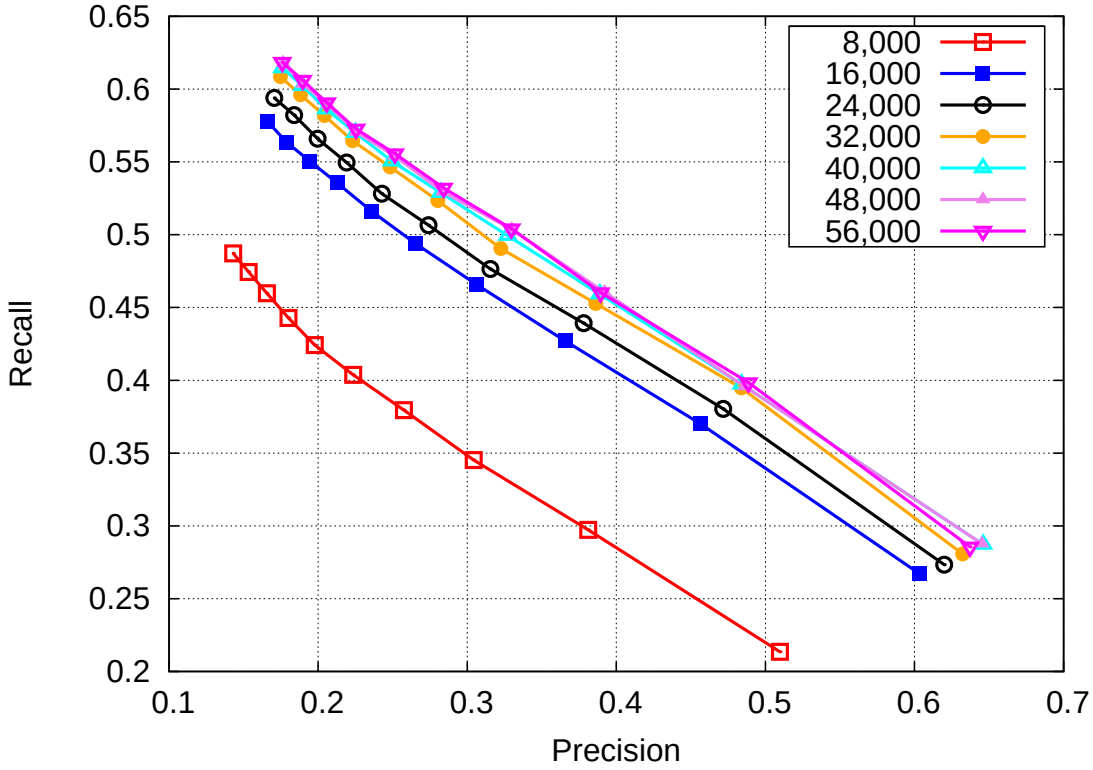


图 5.11　在BOOK数据集合上，当训练集合从8,000增长到56,000时的准确率-召回率曲线图的变化情况。

第 6 章　微博关键词抽取原型系统设计与实现

在前面几章，本文深入研究了基于文档主题结构的关键词抽取方法，在相应的通用测试数据集合（新闻或者学术论文）上取得了较好效果。本章将综合已有研究经验和成果，在具体的应用环境中设计并实现一个实用的关键词抽取原型。

最近，以Twitter(http://twitter.com)和新浪微博(http://t.sina.com.cn)为代表的微型博客(micro-blog，简称微博)成为热门Web2.0应用。相比起博客(blog)，用户发表的每一条微博都有长度限制，如Twitter要求140个英文字母，新浪微博要求140个汉字。由于发表微博快捷便利，微博逐渐成为用户分享和获取信息、发表观点和交流思想的重要方式。微博贴近人们生活，因此微博内容也在一定程度上反映了用户的兴趣。

本章提出微博关键词抽取(keyword extraction for micro-blogs)的任务。该任务的输入是一个查询，这个查询可能是用户ID、主题词或者一个用户关注的好友列表等；输出则是这个查询所返回的微博集合的关键词。其应用价值在于：

- 当输入是一个用户的ID时，可以获取这个用户最近发表的N条微博，这些微博反映了该用户在最近一段时间的兴趣。通过对这些微博进行关键词抽取，就可以得到该用户的关键词，在一定程度上能够表达用户的兴趣。利用这些关键词，可以为用户推荐新闻、好友甚至广告等信息，从而提高用户访问微博的体验。
- 当输入是一个主题词时，可以获取微博系统中与该主题相关的N条微博，这些微博反映了用户在讨论该主题的时候主要讨论什么内容。通过对这些微博进行关键词抽取，就可以得到这些讨论的主要方面和角度，方便用户快速获取他们感兴趣的内容。
- 当输入是一个用户关注的好友列表时，可以对这些用户最近发表的N条微博进行关键词抽取，从而方便用户快速了解他/她的好友最近都在讨论什么话题。

总之，利用关键词抽取可以有效地发现用户兴趣，并以此为基础研制各种应用，提高用户使用的体验和获取信息的效率。

6.1　微博关键词抽取原型系统

6.1.1　系统框架和设计思路

与其他“用户原创内容”(user generated content)的应用类似，微博内容具有海量、异构和多变等重要特点。这些特点要求一个实用的微博关键词抽取系统应当具有较强的处理新词的能力。同时，由于微博具有很强的时效性，因此需要在系统中考虑到每条微博会随时间有不同的权重。

由于新浪微博是目前国内最大的微博系统，并且为开发者提供了丰富的API接口获取用户微博，因此本章选择新浪微博作为微博关键词抽取原型系统的开发平台。图 6.1展示了微博关键词抽取原型系统的框架。如图 6.1所示，微博关键词抽取流程由以下几步组成：

1. 根据输入查询通过新浪API获取微博集合；
2. 利用中文分词系统对微博进行分词；
3. 利用微博权重分析系统和单词权重分析系统计算微博中每个词的权重；
4. 采用第5章的方法，利用翻译概率模型产生微博集合的关键词列表；
5. 利用可视化系统将关键词列表进行可视化，并输出为关键词可视化图片呈现给用户。

在该原型系统中，有两部分进行了特殊设计：

- 中文分词系统需要有实时更新分词词表的机制。这样分词系统就能够较快地、准确地将新词分出来，而不会将其割裂开来从而造成关键词抽取错误。这里选取搜狗输入法网络新词词典(http://pinyin.sogou.com/dict/)、百度热门查询词榜单(http://top.baidu.com/)和新浪微博热门标签(http://weibo.com/pub/top)等作为获取新词的来源，不断更新分词词表。
- 翻译概率模型需要不断增加新的翻译对，训练能够及时反映用户兴趣的翻译模型。这里以各新闻网站的新闻“标题-正文”对和搜索引擎的查询日志中的“查询词-点击网页”对作为新增翻译对来源。

这两个模块的实时更新机制保证了微博关键词抽取系统能够“与时俱进”，抽取更能代表用户兴趣的关键词。下面分别介绍系统中的几个重要模块。

6.1.2　新浪微博API

新浪微博为开发者提供了丰富的API接口(http://open.weibo.com/)获取用户信息和微博内容。微博关键词抽取系统主要使用新浪微博API接口中的以下两组接口：

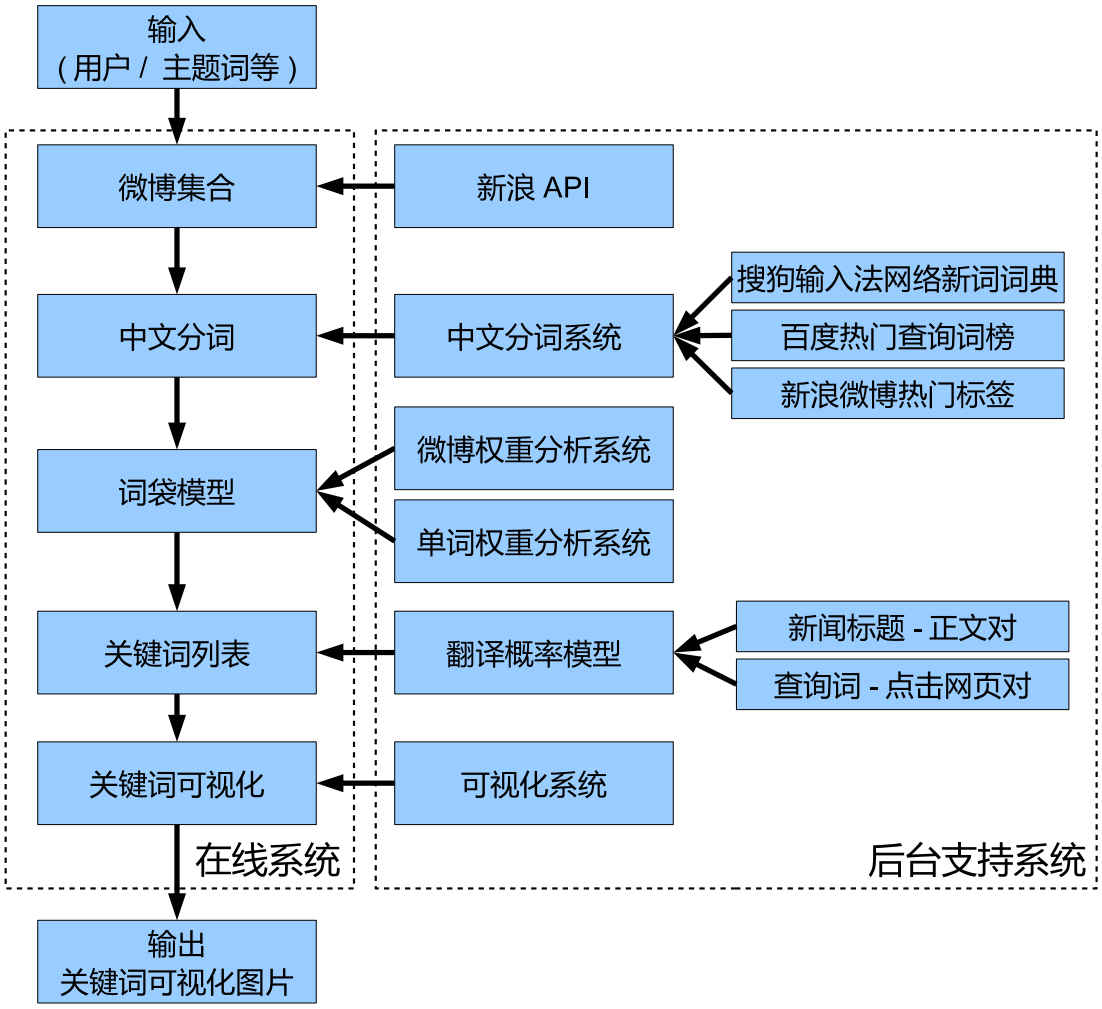


图 6.1　微博关键词抽取原型系统框架。

- 下行数据集(Timeline)接口。用来获取给定用户(user_timeline)，给定主题词(trends)，或者给定用户的好友列表(friends_timeline)所发表的微博集合。
- 微博访问(Statuses)接口。用来将产生的关键词可视化图片作为一条微博发布到用户微博中(upload)。

6.1.3　中文分词系统

这里采用的中文分词系统以清华大学自然语言处理组开发的高效中文分词系统[133]为基础，增加了新词词表的动态更新模块。该分词系统采用简单的产生式模型，在保证分词精度的同时(在SIGHAN 2005 MSRA语料库上F值只比评测中最好的结果低0.001)，分词速度实现较大提高，单机单线程分词速度达到每秒100到200万字。

该系统能够从外部加载新词词表，这里新词来源分别为：搜狗输入法网络新词词典、百度热门查询词榜单和和新浪微博热门标签。未来还可以增加更多的新词来源，如百度百科词条和社会化标签系统中的标签，等。

6.1.4　微博权重分析系统

这里微博权重分析方法将主要考虑时间因素。一个直观的假设是用户当前的兴趣与最近发表的微博更为相关，而与较为久远的微博相关程度较低。给定一个查询q，设返回微博集合为$D_q = d_1, \ldots, d_N$，其中每条微博d_n都有一个时间戳s_n。假设查询发生时间为s，可以定义微博d_n的权重为：

$$\Pr(d_n|q) = \frac{1}{\exp(s - s_n)} \tag{6-1}$$

其物理意义是，给定查询q及其查询时间s，距离当前时间越远的微博权重越小。

除了时间因素之外，一条微博的权重还应与该微博被转发的次数等有密切关系。更细致地考虑微博权重将在未来工作中逐渐完善。

6.1.5　单词权重分析系统

根据第5章方法，需要估计文档中每个词的权重$\Pr(w|d)$，这部分工作由“单词权重分析系统”来完成。第5章采取了归一化TFIDF来度量$\Pr(w|d)$。在实验中，也尝试过TextRank、LDA以及第4章的TPR来度量单词权重，但是与TFIDF相比优势并不显著。考虑到TPR等技术计算复杂度较高，因此这里仍然采用归一化TFIDF计算单词权重。

6.1.6　翻译概率模型

如第5章，这里的翻译概率模型采用IBM Model-1进行训练。考虑到微博关键词抽取的实时性要求，这里对IBM Model-1进行改进，使之允许随时加入新翻译对来进行训练，从而使翻译模型能够“与时俱进”。这里以各新闻网站的新闻“标题-正文”对和搜索引擎查询日志中的“查询词-点击网页”对作为新增翻译对来源。

综上，给定查询q，可以按照以下公式计算关键词的重要性，并对其进行排序：

$$\Pr(p|q) = \sum_{t \in p} \sum_{d_n \in D_q} \sum_{w \in d_n} \Pr(t|w) \Pr(w|d_n) \Pr(d_n|q) \tag{6-2}$$

接下来，需要将关键词排序列表输入到可视化系统进行可视化。

6.1.7　可视化系统

为了提高用户使用微博关键词抽取系统的用户体验，本章使用Wordle技术[110]① 对抽取关键词进行可视化。本章选用了开源工具PyTagCloud② 实现Wordle技术。

6.2　系统应用效果

本节将主要介绍微博关键词原型系统的应用效果。目前该系统提供四个功能：

- “我的微博关键词”，用来产生使用该系统的用户自身的微博关键词，通过抓取用户发表的最近200条微博进行分析。
- “Ta的微博关键词”，允许用户指定想要了解用户的昵称，通过抓取输入昵称的用户发表的最近200条微博进行分析。
- “主题的微博关键词”，允许用户输入主题词，通过抓取与该主题相关的最近50条微博进行分析。
- “好友的微博关键词”，用来产生使用该系统的用户好友们的微博关键词，通过抓取用户好友在一段时间内(如1个小时内，由用户指定)的微博进行分析。

① 可以通过http://wordle.net/访问。
② 该开源软件可通过https://github.com/PaulKlinger/PyTagCloud访问。

6.2.1　微博关键词可视化示例

这里几个微博关键词可视化图片示例来展示关键词抽取效果。图 6.2、图 6.3和图 6.4分别显示了该博士论文作者刘知远、清华大学马少平老师以及该博士论文导师孙茂松老师的微博关键词。马少平老师的主要研究方向是信息检索和搜索引擎，而孙茂松老师的主要研究方向是自然语言处理。可以看到，系统能够较好地反映用户的兴趣，包括个人爱好和研究兴趣等。

图 6.2　刘知远(http://weibo.com/zibuyu9)的微博关键词。

图 6.5和图 6.6则显示了微软亚洲研究院官方微博和清华大学百年校庆官方微博的关键词。从图中也可以看到所抽取的关键词能够较好地反映微博主题。

图 6.7和图 6.8显示了刘知远的好友分别在2011年5月25日18:40前1个小时内和2011年5月22日11:07前1个小时内的微博关键词。可以看到，微博关键词抽取系统能够较好地获取多变的微博内容主题。

图 6.9显示了主题词“IBM”的微博关键词。能够看出这些关键词与对应的主题词有较大联系。不过由于新浪微博API为主题词提供的最大返回微博数仅为50，因此相比起其他几个功能，“主题的微博关键词”可视化效果相对较差。

6.2.2　系统应用统计数据

微博关键词抽取系统以新浪微博应用(Sina Weibo App, SWA)的形式上线，用户通过授权给该应用，可以利用系统进行关键词分析。该应用的名称为“围脖关键词”。

图 6.3　马少平老师(http://weibo.com/thumsp)的微博关键词。

图 6.4　孙茂松老师(http://weibo.com/1970879995)的微博关键词。

该应用在2011年5月25日通过新浪微博应用审核。从通过审核到5月29日，新浪微博提供的应用使用情况数据如表 6.1所示。从该表格可以看到，从5月26日到5月29日，应用以每天2万用户左右的速度快速增长。到5月30日凌晨，应用的累积用户数已达8万人，接口调用总次数超过100万次。考虑到目前运行该系统服务器有限的计算能力，这些统计数据充分表明微博关键词抽取原型系统能够有效抽取关键词，受到广大微博用户的认可。

图 6.5　微软亚洲研究院(http://weibo.com/msra)的微博关键词。

图 6.6　清华大学百年校庆(http://weibo.com/tsinghua100)的微博关键词。

6.3　本章小结

本章根据已有研究工作经验和方法(主要是第5章方法)，在新浪微博平台上设计并实现了微博关键词抽取原型系统。该系统的设计考虑了微博数据海量、异构、多变的特点，能够动态更新用于中文分词的新词词表和用于训练翻译概率模型的翻译对。另外，该系统也初步考虑了微博应当随时间而有不同的权重，使抽取的关键词能够更好地表达用户当前的兴趣。该系统上线后的用户统计数据表明，本

图 6.7　刘知远的好友在2011年5月25日18:40前1个小时内的微博关键词。

图 6.8　刘知远的好友在2011年5月22日11:07前1个小时内的微博关键词。

文所进行的基于文档主题结构的关键词抽取研究能够较好地适应Web用户需求。

目前该系统仍然是一个原型系统，距离成为一个成熟的技术产品还有许多需要完善的地方。很多微博用户也对该系统提出了各种有益的意见和建议。例如，目前的可视化是以图片的形式输出的，缺乏用户交互机制。作为未来工作，需要设计一个具有良好交互能力的可视化界面，如利用HTML5技术进行可视化，使用户能够点击感兴趣的关键词，搜索和浏览包含该关键词的微博。

图 6.9　主题词“IBM”的微博关键词。

表 6.1　微博关键词原型系统使用情况。

时间	使用用户数	新增用户数	接口调用数
2011年5月25日	526	501	15,201
2011年5月26日	24,047	23,964	274,089
2011年5月27日	22,826	21,043	260,023
2011年5月28日	19,599	17,448	237,878
2011年5月29日	24,273	21,381	303,315
总结	91,271	84,337	1,090,506

第 7 章　总结与展望

关键词是快速获取文档主题的重要方法，在信息检索和自然语言处理等领域均有重要应用。传统的方法仅依靠词汇的统计信息进行推荐，没有考虑文档主题结构对关键词抽取的影响。本文主要研究考虑文档主题结构的关键词抽取方法。本文针对文档主题结构在关键词抽取中的重要作用，从四个方面提出考虑文档主题结构的关键词抽取方法：基于文档内部信息构建主题的关键词抽取，基于隐含主题模型构建主题的关键词抽取，综合利用隐含主题模型和文档结构的关键词抽取，以及基于文档与关键词主题一致性的关键词抽取。因此，本文的主要贡献可以从这四个方面来总结。

7.1　论文的主要贡献

首先，提出基于文档内部信息，利用文档的词聚类算法构建文档主题，进行关键词抽取。该方法仅依靠文档内部信息，通过度量文档中词与词之间的相似度，利用聚类的方法构建文档主题，并根据不同主题在文档中的重要性，进行关键词抽取。实验证明，该方法能够在一定程度上发现文档主要话题，并抽取出与文档主题相关的关键词，提高了关键词对文档主题的覆盖度。

其次，提出基于文档外部信息，利用隐含主题模型构建文档主题，进行关键词抽取。该方法针对基于文档内部信息通过聚类算法进行关键词抽取受限于文档提供信息不足的缺点，提出利用机器学习算法中广泛使用的隐含主题模型构建文档主题，进行关键词抽取。并针对隐含主题模型训练速度较慢的瓶颈，提出了一种高效的并行隐含主题模型。实验证明，该方法能够更好地构建文档主题，并有效抽取关键词。

再次，提出综合利用隐含主题模型和文档结构信息，进行关键词抽取。该方法针对隐含主题模型无法考虑文档结构信息的缺点，提出综合利用隐含主题模型和文档结构信息的方法：基于主题的随机游走模型，进行关键词抽取。该方法一方面能够通过隐含主题模型构建文档主题，同时能够通过文档图的随机游走模型考虑文档结构为关键词抽取提供信息，实验证明，该方法能够综合隐含主题模型和文档结构信息进行关键词抽取的优势，有效抽取关键词。

最后，基于文档与关键词主题一致性的前提，提出基于机器翻译模型的关键词抽取方法。该方法针对文档和关键词之间存在较大词汇差异的问题，基于文档

和关键词主题一致性的前提，提出利用机器翻译中的词对齐模型计算文档中的词到关键词的翻译概率，然后进行关键词抽取。实验证明该方法能够有效的建立文档词汇与关键词之间的语义联系，有效的推荐关键词。

7.2 工作展望

展望未来，关键词抽取研究还有很多工作需要完成。这里总结以下亟待探索的研究方向和路线:

1. 在中文关键词抽取方面还有很多中文独特的问题需要处理，如中文自动分词等。针对典型Web应用环境（如社会新闻、科技新闻、博客、论文、评论等），研究在自动分词必然存在一定错误的条件下从中文文本中比较准确地提取候选关键词的算法，综合考虑候选关键词内部结合的紧密程度及其上下文决定的独立程度相结合，分词与字的N-gram相结合，局部统计量和全局统计量相结合的“三结合”因素, 并探讨词法、句法、语义等多层次信息主要以统计形式应用于Web文本候选关键词提取中的现实性。
2. 研究词语粒度对关键词提取的影响及其分析算法，考察多个不同粒度的词语关于此任务的一般配组规律与取舍原则。
3. 从篇章连贯性、主题分布等角度考察并分析关键词与文本结构（如文本标题、首段落、尾段落、句群等）之间的联系，研究融入了“显性”文本结构与“隐性”文本结构的关键词提取算法, 以及所提取关键词的数量及构成与文本长短、类型、内容之间的关系。
4. 研究Web环境下受控词表的及时扩充和更新算法，一方面充分利用维基百科之类的开放资源，另一方面充分利用用户日志、社会标注等资源。
5. 对社会标签集合的歧义现象进行大规模的考察，研究在关键词自动标注条件下的标签歧义消解算法。
6. 研究适合于大规模受控词表的关键词分配算法，在高频词的高覆盖性与低频词的高区分性之间寻找合适的平衡点。
7. 研究关键词抽取与关键词分配相结合的关键词自动标注算法，能够根据文本类型及其文本结构信息，自动确定两种来源的候选关键词的混合方式及混合比例。
8. 从社会网络计算的角度对社会标签集合的各种性质进行全面考察，通过自动发现社会标签之间的各种关联，研究有效组织大规模社会标签的算法，进一步地，将“标签云”表示演变为“标签森林”表示，从而有效提高社会标签

集合的整体表达能力。

9. 研究中文社会标签推荐的动力学演化过程，从多个角度(如关于空间、时间的动态稳定性、变化率等)对其进行定量描述与分析，并试图揭示隐含于这个动力学过程中的语言学、社会学等方面的规律，提出一种基于时间动力学模型的中文关键词标注算法。
10. 在集成和融合上述研究成果的基础上，设计并实现一个针对大规模Web中文文本的关键词自动标注系统。

总之，关键词和标签反映了人们的兴趣，随着时间有着剧烈的演化。因此关键词标注和标签推荐是发现用户兴趣、分析趋势、发现热点的重要工具。通过解决上述问题，最终可以建立一个完整的关键词标注和标签推荐系统，并以此为基础构建兴趣发现、趋势和热点检测、以及关键词和标签可视化系统。这无论对信息社会的发展、互联网商业模式，还是对研究人类文化演变都具有重要而深远的意义。本文认为，这是综合运用并检验关键词标注和标签推荐有效性的非常合适的应用任务。由上可见，针对关键词标注和标签推荐的研究，无论是在理论研究方面，还是在关键应用技术研究方面，都极具前瞻性的探索空间，本文只是对其中的两个重要挑战进行了研究，还有很多研究问题值得进行深入而富有开创性的探索。

参考文献

[1] 金观涛, 刘青峰. 观念史研究: 中国现代重要政治术语的形成. 北京: 法律出版社, 2009.

[2] Williams R. Keywords: A vocabulary of culture and society. USA: Oxford University Press, 1985.

[3] 雷蒙威廉斯. 关键词: 文化与社会的词汇. 北京: 北京: 三联书店, 2005.

[4] Turney P D. Learning Algorithms for Keyphrase Extraction. Information Retrieval, 2000, 2(4):303–336.

[5] Manning C D, Schütze H. Foundations of statistical natural language processing. Cambridge, MA, USA: MIT Press, 1999.

[6] Smadja F. Retrieving collocations from text: Xtract. Computational Linguistics, 1993, 19(1):143–177. 972458.

[7] Church K W, Hanks P. Word association norms, mutual information, and lexicography. Computational Linguistics, 1990, 16(1):22–29. 89095.

[8] Church K, Gale W, Hanks P, et al. Using Statistics in Lexical Analysis. Lexical acquisition: exploiting on-line resources to build a lexicon, 1991. 115–164.

[9] Dunning T. Accurate methods for the statistics of surprise and coincidence. Computational Linguistics, 1993, 19(1):61–74.

[10] Tomokiyo T, Hurst M. A Language Model Approach to Keyphrase Extraction. Proceedings of ACL 2003 workshop on Multiword expressions, 2003. 33–40.

[11] Silva J, Lopes G. Towards Automatic Building of Document Keywords. Proceedings of COLING, 2010. 1149–1157.

[12] Hulth A. Improved Automatic Keyword Extraction Given More Linguistic Knowledge. Proceedings of EMNLP, 2003. 216–223.

[13] Manning C, Raghavan P, Schtze H. Introduction to information retrieval. New York, NY, USA: Cambridge University Press, 2008.

[14] Frank E, Paynter G W, Witten I H, et al. Domain-specific Keyphrase Extraction. Proceedings of IJCAI, 1999. 668–673.

[15] Page L, Brin S, Motwani R, et al. The pagerank citation ranking: Bringing order to the web. Technical report of Stanford Digital Library Technologies Project, 1998..

[16] Mihalcea R, Tarau P. TextRank: Bringing Order into Texts. Proceedings of EMNLP, 2004. 404–411.

[17] Litvak M, Last M. Graph-Based Keyword Extraction for Single-Document Summarization. Proceedings of Workshop Multi-source Multilingual Information Extraction and Summarization, 2008. 17–24.

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

[19] Wan X, Xiao J. CollabRank: Towards a Collaborative Approach to Single-Document Keyphrase Extraction. Proceedings of COLING, 2008. 969–976.

[20] Wan X, Xiao J. Single Document Keyphrase Extraction Using Neighborhood Knowledge. Proceedings of AAAI, 2008. 855–860.

[21] Zha H. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. Proceedings of SIGIR, 2002. 113–120.

[22] Wan X, Yang J, Xiao J. Towards an Iterative Reinforcement Approach for Simultaneous Document Summarization and Keyword Extraction. Proceedings of ACL, volume 45, 2007. 552–559.

[23] Li D, Li S, Li W, et al. A semi-supervised key phrase extraction approach: learning from title phrases through a document semantic network. Proceedings of ACL, 2010. 296–300.

[24] Ercan G, Cicekli I. Using lexical chains for keyword extraction. Information Processing Management, 2007, 43(6):1705–1714.

[25] Huang C, Tian Y, Zhou Z, et al. Keyphrase Extraction Using Semantic Networks Structure Analysis. Proceedings of ICDM, 2006. 275–284.

[26] Tsoumakas G, Katakis I. Multi-label classification: An overview. International Journal of Data Warehousing and Mining, 2007, 3(3):1–13.

[27] Tsoumakas G, Katakis I, Vlahavas I. Mining multi-label data. Data Mining and Knowledge Discovery Handbook, 2010. 667–685.

[28] Medelyan O. Automatic keyphrase indexing with a domain-specific thesaurus[D]. Breisgau, Germany: University of Freiburg, 2005.

[29] Medelyan O, Witten I H. Thesaurus based automatic keyphrase indexing. Proceedings of JCDL, 2007. 296–297.

[30] Medelyan O, Witten I H. Domain-independent Automatic Keyphrase Indexing with Small Training Sets. Journal of the American Society for Information Science and Technology, 2008, 59(7):1026–1040.

[31] Xue G R, Xing D, Yang Q, et al. Deep classification in large-scale text hierarchies. Proceedings of SIGIR, 2008. 619–626.

[32] Milne D, Witten I H. Learning to link with wikipedia. Proceedings of CIKM, 2008. 509–518.

[33] Medelyan O, Witten I H, Milne D. Topic Indexing with Wikipedia. Proceedings of AAAI WikiAI workshop, 2008.

[34] Zheng Y, Liu Z, Sun M, et al. Incorporating user behaviors in new word detection. Proceedings of IJCAI, 2009. 2101–2106.

[35] Eisenstein J, O'Connor B, Smith N A, et al. A latent variable model for geographic lexical variation. Proceedings of EMNLP, 2010. 1277–1287.

[36] Wang R C, Cohen W W. Language-Independent Set Expansion of Named Entities Using the Web. Proceedings of ICDM, 2007. 342–350.

[37] Wang R C, Cohen W W. Iterative Set Expansion of Named Entities Using the Web. Proceedings of ICDM, 2008. 1091–1096.

[38] Wang R C, Cohen W W. Automatic set instance extraction using the web. Proceedings of ACL-IJCNLP, 2009. 441–449.

[39] Pasca M. Organizing and searching the world wide web of facts–step two: harnessing the wisdom of the crowds. Proceedings of WWW, 2007. 101–110.

[40] Pasca M. Weakly-supervised discovery of named entities using web search queries. Proceedings of CIKM, 2007. 683–690.

[41] Pasca M, Van Durme B. Weakly-supervised acquisition of open-domain classes and class attributes from web documents and query logs. Proceedings of ACL, 2008. 19–27.

[42] Talukdar P P, Reisinger J, Pa M, et al. Weakly-supervised acquisition of labeled class instances using graph random walks. Proceedings of EMNLP, 2008. 582–590.

[43] 邹纲, 刘洋, 刘群, et al. 面向Internet的中文新词语检测. 中文信息学报, 2004, 18(6):1–9.

[44] 刘华. 一种快速获取领域新词语的新方法. 中文信息学报, 2006, 20(5):17–23.

[45] Zhu X, Goldberg A B. Introduction to semi-supervised learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 2009, 3(1):1–130.

[46] Madani O, Connor M, Greiner W. Learning When Concepts Abound. Journal of Machine Learning Research, 2009, 10:2571–2613.

[47] Crammer K, Dekel O, Keshet J, et al. Online Passive-Aggressive Algorithms. Journal of Machine Learning Research, 2006, 7:551–585.

[48] Zhang M L, Zhang K. Multi-label learning by exploiting label dependency. Proceedings of KDD, 2010. 999–1008.

[49] Xu Z, Fu Y, Mao J, et al. Towards the semantic web: Collaborative tag suggestions. Proceedings of Collaborative Web Tagging Workshop at WWW2006, 2006.

[50] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms. Proceedings of WWW, 2001. 285–295.

[51] Jaschke R, Marinho L, Hotho A, et al. Tag Recommendations in Folksonomies. Proceedings of ECML/PKDD, 2007. 506–514.

[52] Rendle S, Schmidt-Thieme L. Pairwise interaction tensor factorization for personalized tag recommendation. Proceedings of WSDM, 2010. 81–90.

[53] Ohkura T, Kiyota Y, Nakagawa H. Browsing System for Weblog Articles based on Automated Folksonomy. Proceedings of Workshop on the Webloging Ecosystem at WWW, 2006.

[54] Lee S O K, Chun A H W. Automatic Tag Recommendation for the Web 2.0 Blogosphere Using Collaborative Tagging and Hybrid ANN Semantic Structures. Proceedings of The 6th WSEAS International Conference on Applied Computer Science, 2007. 88–93.

[55] Katakis I, Tsoumakas G, Vlahavas I. Multilabel Text Classification for Automated Tag Suggestion. Proceedings of ECML PKDD Discovery Challenge, 2008. 75–83.

[56] Shepitsen A, Gemmell J, Mobasher B, et al. Personalized Recommendation in Collaborative Tagging Systems Using Hierarchical Clustering. Proceedings of ACM conference on Recommender systems, 2008. 259–266.

[57] ECML/PKDD Discovery Challenge 2008 Result Page, 2008. [2010年10月访问].

[58] Mishne G. AutoTag: a collaborative approach to automated tag assignment for weblog posts. Proceedings of WWW, 2006. 953–954.

[59] Fujimura S, Fujimura K O, Okuda H. Blogosonomy: Autotagging Any Text Using Bloggers' Knowledge. Proceedings of WI, 2007. 205–212.

[60] Sood S C, Owsley S H, Hammond K J, et al. TagAssist: Automatic Tag Suggestion for Blog Posts. Proceedings of ICWSM, 2007.

[61] Tatu M, Srikanth M, D Silva T. RSDC'08: Tag recommendations using bookmark content. Proceedings of ECML/PKDD Discovery Challenge, volume 2008, 2008. 96–107.

[62] Lipczak M, Hu Y, Kollet Y, et al. Tag sources for recommendation in collaborative tagging systems. Proceedings of ECML/PKDD Discovery Challenge, 2009.

[63] Landauer T K, Foltz P W, Laham D. An Introduction to Latent Semantic Analysis. Discourse Processes, 1998, 25:259–284.

[64] Hofmann T. Probabilistic latent semantic indexing. Proceedings of SIGIR, 1999. 50–57.

[65] Blei D M, Ng A Y, Jordan M I. Latent Dirichlet Allocation. Journal of Machine Learning Research, 2003, 3:993–1022.

[66] Krestel R, Fankhauser P, Nejdl W. Latent dirichlet allocation for tag recommendation. Proceedings of ACM conference on Recommender systems, 2009. 61–68.

[67] Si X, Sun M. Tag-LDA for Scalable Real-time Tag Recommendation. Journal of Computational Information Systems, 2009, 6(2):23–31.

[68] Bundschus M, Yu S, Tresp V, et al. Hierarchical Bayesian Models for Collaborative Tagging Systems. Proceedings of ICDM, 2009. 728–733.

[69] Iwata T, Yamada T, Ueda N. Modeling Social Annotation Data with Content Relevance using a Topic Model. Proceedings of NIPS, 2009. 835–843.

[70] Miller G A, Beckwith R, Fellbaum C, et al. WordNet: An on-line lexical database. International Journal of Lexicography, 1990, 3:235–244.

[71] Barzilay R, Elhadad M. Using lexical chains for text summarization. Proceedings of The ACL Workshop on Intelligent Scalable Text Summarization, volume 17. Madrid, spain, 1997.

[72] Silber H, McCoy K. Efficiently computed lexical chains as an intermediate representation for automatic text summarization. Computational Linguistics, 2002, 28(4):487–496.

[73] Elbeltagy S, Rafea A. KP-Miner: A keyphrase extraction system for English and Arabic documents. Information Systems, 2009, 34(1):132–144.

[74] Gabrilovich E, Markovitch S. Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. Proceedings of IJCAI, 2007. 6–12.

[75] Cilibrasi R L, Vitanyi P M B. The Google Similarity Distance. IEEE Transactions on Knowledge and Data Engineering, 2007, 19(3):370–383.

[76] Han J, Kamber M. Data Mining: Concepts and Techniques, second edition. San Francisco, CA, USA: Morgan Kaufmann, 2005.

[77] Luxburg U. A tutorial on spectral clustering. Technical report, Max Planck Institute for Biological Cybernetics, 2006.

[78] Chen W Y, Song Y, Bai H, et al. PSC: Paralel Spectral Clustering. Submitted, 2008.

[79] Frey B J J, Dueck D. Clustering by Passing Messages Between Data Points. Science, 2007..

[80] Hulth A. Improved automatic keyword extraction given more linguistic knowledge. Proceedings of EMNLP, 2003. 216–223.

[81] Mihalcea R, Tarau P. TextRank: Bringing Order into Texts. Proceedings of EMNLP, 2004.

[82] Griffiths T, Steyvers M. Finding scientific topics. Proceedings of the National Academy of Sciences of the United States of America, 2004, 101(90001):5228–5235.

[83] Rosen-Zvi M, Chemudugunta C, Griffiths T, et al. Learning author-topic models from text corpora. ACM Transactions on Information Systems, 2010, 28(1):1–38.

[84] Li W, McCallum A. Pachinko Allocation: DAG-Structured Mixture Models of Topic Correlations. Proceedings of ICML, 2006.

[85] Chemudugunta C, Smyth P, Steyvers M. Modeling general and specific aspects of documents with a probabilistic topic model. Proceedings of NIPS, 2007. 241–248.

[86] Newman D, Asuncion A, Smyth P, et al. Distributed Algorithms for Topic Models. Journal of Machine Learning Research, 2009, 10:1801–1828.

[87] Liu J. The Collapsed Gibbs Sampler in Bayesian Computations with Applications to a Gene Regulation Problem. Journal of the American Statistical Association, 1994, 89(427).

[88] Mimno D M, McCallum A. Organizing the OCA: learning faceted subjects from a library of digital books. Proceedings of ACM/IEEE Joint Conference on Digital Libraries, 2007. 376–385.

[89] Newman D, Asuncion A, Smyth P, et al. Distributed Inference for Latent Dirichlet Allocation. Proceedings of NIPS, 2007. 1081–1088.

[90] Asuncion A, Smyth P, Welling M. Asynchronous Distributed Learning of Topic Models. Proceedings of NIPS, 2008. 81–88.

[91] Yan F, Xu N, Qi Y. Parallel Inference for Latent Dirichlet Allocation on Graphics Processing Units. Proceedings of NIPS, 2009. 2134–2142.

[92] Gomes R, Welling M, Perona P. Memory bounded inference in topic models. Proceedings of ICML, 2008. 344–351.

[93] Porteous I, Newman D, Ihler A, et al. Fast Collapsed Gibbs Sampling For Latent Dirichlet Allocation. Proceedings of KDD, 2008. 569–577.

[94] Wang Y, Bai H, Stanton M, et al. PLDA: Parallel latent dirichlet allocation for large-scale applications. Proceedings of Algorithmic Aspects in Information and Management, 2009. 301–314.

[95] Chen W, Chu J, Luan J, et al. Collaborative filtering for orkut communities: discovery of user latent behavior. Proceedings of WWW, 2009. 681–690.

[96] Dean J, Ghemawat S. MapReduce: Simplified Data Processing on Large Clusters. Proceedings of OSDI, 2004. 137–150.

[97] Chu C T, Kim S K, Lin Y A, et al. MapReduce for Machine Learning on Multicore. Proceedings of NIPS, 2006.

[98] Graham S, Snir M, Patterson C. Getting up to speed: The future of supercomputing. Washington, D.C., USA: National Academies Press, 2005.

[99] Shen J P, Lipasti M H. Modern Processor Design: Fundamentals of Superscalar Processors. USA: McGraw-Hill Higher Education, 2005.

[100] Blinn J. A trip down the graphics pipeline: Line clipping. IEEE Computer Graphics and Applications, 1991, 11(1):98–105.

[101] Berenbrink P, Friedetzky T, Hu Z, et al. On weighted balls-into-bins games. Theoretical Computer Science, 2008, 409(3):511–520.

[102] Asuncion A, Smyth P, Welling M. Asynchronous distributed estimation of topic models for document analysis. Statistical Methodology, 2010..

[103] Heinrich G. Parameter Estimation for text analysis. Technical report, Vsonix GmbH and University of Leipzig, 2008.

[104] Over P, Liggett W, Gilbert H, et al. Introduction to DUC-2001: An intrinsic evaluation of generic news text summarization systems. Proceedings of Proceedings of DUC2001, 2001.

[105] Haveliwala T. Topic-sensitive PageRank. Proceedings of WWW, 2002. 517–526.

[106] Haveliwala T. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. IEEE transactions on knowledge and data engineering, 2003. 784–796.

[107] Cohn D, Chang H. Learning to Probabilistically Identify Authoritative Documents. Proceedings of Proceedings of ICML, 2000. 167–174.

[108] Buckley C, Voorhees E. Retrieval evaluation with incomplete information. Proceedings of Proceedings of SIGIR, 2004. 25–32.

[109] Voorhees E. The TREC-8 question answering track report. Proceedings of Proceedings of TREC, 2000. 77–82.

[110] Viegas F, Wattenberg M, Feinberg J. Participatory visualization with wordle. IEEE Transactions on Visualization and Computer Graphics, 2009. 1137–1144.

[111] Berger A, Lafferty J. Information retrieval as statistical translation. Proceedings of SIGIR, 1999. 222–229.

[112] Karimzadehgan M, Zhai C. Estimation of statistical translation models based on mutual information for ad hoc information retrieval. Proceedings of SIGIR, 2010. 323–330.

[113] Duygulu P, Barnard K, Freitas J F G, et al. Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. Proceedings of ECCV, 2002. 97–112.

[114] Berger A, Caruana R, Cohn D, et al. Bridging the lexical chasm: statistical approaches to answer-finding. Proceedings of SIGIR, 2000. 192–199.

[115] Echihabi A, Marcu D. A noisy-channel approach to question answering. Proceedings of ACL, 2003. 16–23.

[116] Murdock V, Croft W. Simple translation models for sentence retrieval in factoid question answering. Proceedings of SIGIR, 2004.

[117] Soricut R, Brill E. Automatic question answering using the web: Beyond the factoid. Information Retrieval, 2006, 9(2):191–206.

[118] Xue X, Jeon J, Croft W. Retrieval models for question and answer archives. Proceedings of SIGIR, 2008. 475–482.

[119] Riezler S, Vasserman A, Tsochantaridis I, et al. Statistical machine translation for query expansion in answer retrieval. Proceedings of ACL, 2007. 464–471.

[120] Riezler S, Liu Y, Vasserman A. Translating queries into snippets for improved query expansion. Proceedings of COLING, 2008. 737–744.

[121] Riezler S, Liu Y. Query rewriting using monolingual statistical machine translation. Computational Linguistics, 2010, 36(3):569–582.

[122] Banko M, Mittal V, Witbrock M. Headline generation based on statistical translation. Proceedings of ACL, 2000. 318–325.

[123] Liu Z, Wang H, Wu H, et al. Collocation extraction using monolingual word alignment method. Proceedings of EMNLP, 2009. 487–495.

[124] Liu Z, Wang H, Wu H, et al. Improving Statistical Machine Translation with monolingual collocation. Proceedings of ACL, 2010. 825–833.

[125] Quirk C, Brockett C, Dolan W. Monolingual machine translation for paraphrase generation. Proceedings of EMNLP, volume 149, 2004.

[126] Zhao S, Wang H, Liu T. Paraphrasing with Search Engine Query Logs. Proceedings of COLING, 2010. 1317–1325.

[127] Brown P, Pietra V, Pietra S, et al. The mathematics of statistical machine translation: Parameter estimation. Computational linguistics, 1993, 19(2):263–311.

[128] Dempster A, Laird N, Rubin D, et al. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 1977, 39(1):1–38.

[129] Koehn P. Statistical machine translation. New York, NY, USA: Cambridge University Press, 2010.

[130] Och F, Ney H. A systematic comparison of various statistical alignment models. Computational linguistics, 2003, 29(1):19–51.

[131] Goldstein J, Mittal V, Carbonell J, et al. Multi-document summarization by sentence extraction. Proceedings of NAACL-ANLP 2000 Workshop on Automatic summarization, 2000. 40–48.

[132] Si X, Liu Z, Sun M. Modeling Social Annotations via Latent Reason Identification. IEEE Intelligent Systems, 2010, 25(6):42 – 49.

[133] Zhang K, Sun M, Xue P. A Local Generative Model for Chinese Word Segmentation. Information Retrieval Technology, 2010. 420–431.

致　谢

感谢我的导师孙茂松教授对本人的精心指导和悉心培养，您严谨治学的态度让我受益终生。感谢谷歌研究院张智威博士对本人的指导和培养。感谢我的父亲刘树江和母亲崔秀玲，在你们的宽容、支持和鼓励下，我能够专心致志奋斗到今天。感谢实验室的各位老师。感谢实验室同学们的热情帮助和支持。特别感谢郑亚斌、司宪策、李鹏、陈新雄、黄文溢和张开旭等同学在我博士生涯的不同时期给予的帮助和支持。感谢计算机系学生组伙伴们的帮助和支持。感谢所有曾经帮助、指导过我的同学和朋友。

本课题承蒙中国高技术研究发展计划、国家自然科学基金和清华-谷歌联合研究项目的资助，特此致谢。

声　明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签　名：_____ 日　期：_____

附录 A NEWS数据中文档号为AP880510-0178的新闻全文

标题

Arafat Says U.S. Threatening to Kill PLO Officials

正文

Yasser Arafat on Tuesday accused the United States of threatening to kill PLO officials if Palestinian guerrillas attack American targets.

The United States denied the accusation.

The State Department said in Washington that it had received reports the PLO might target Americans because of alleged U.S. involvement in the assassination of Khalil Wazir, the PLO's second in command.

Wazir was slain April 16 during a raid on his house near Tunis, Tunisia. Israeli officials who spoke on condition they not be identified said an Israeli squad carried out the assassination.

There have been accusations by the PLO that the United States knew about and approved plans for slaying Wazir.

Arafat, the Palestine Liberation Organization leader, claimed the threat to kill PLO officials was made in a U.S. government document the PLO obtained from an Arab government. He refused to identify the government.

In Washington, Assistant Secretary of State Richard Murphy denied Arafat's accusation that the United States threatened PLO officials.

State Department spokesman Charles Redman said the United States has been in touch with a number of Middle Eastern countries about possible PLO attacks against American citizens and facilities.

He added that Arafat's interpretation of those contacts was "entirely without foundation."

Arafat spoke at a news conference in his heavily guarded villa in Baghdad, where extra security guards have been deployed. He said security also was being augmented at PLO offices around the Arab world following the alleged threat.

He produced a photocopy of the alleged document. It appeared to be part of a longer document with the word “CONFIDENTIAL” stamped at the bottom.

The document, which was typewritten in English, referred to Wazir by his code name, Abu Jihad. It read:

“You may be aware of charges in several Middle Eastern and particulary Palestinian circles that the U.S. knew of and approved Abu Jihad’s assassination.

“On April 18th (a) State Department spokesman said that the United States ‘condemns this act of political assassination,’ ‘had no knowledge of’ and ‘was not involved in any way in this assassination.

“It has come to our attention that the PLO leader Yasser Arafat may have personally approved a series of terrorist attacks against American citizens and facilities abroad, possibly in retaliation for last month’s assassination of Abu Jihad.

“Any possible targeting of American personnel and facilities in retaliation for Abu Jihad’s assassination would be totally reprehensible and unjustified. We would hold the PLO responsible for any such attacks.”

Arafat said the document “reveals the U.S. administration is planning, in full cooperation with the Israelis, to conduct a crusade of terrorist attacks and then to blame the PLO for them.

“These attacks will then be used to justify the assassination of PLO leaders.”

He strongly denied that the PLO planned any such attacks.

附录 B 新闻“以军方称伊朗能造核弹 可能据此对伊朗动武”全文

标题

以军方称伊朗能造核弹 可能据此对伊朗动武

摘要

核心提示：以色列军方情报负责人Maj-Gen Amos Yadlin称，伊朗目前可以建造一颗核弹，并且不久将有充足的浓缩铀来造第二颗核弹。这与美国中情局和国际原子能机构之前的判断一致。这名负责人的言论十分重要，因为以色列不会排除采取军事行动以防止伊朗开发核武器。

正文

环球时报-环球网11月4日报道以色列军方情报负责人表示，伊朗目前有足够的浓缩铀，可以建造一颗核弹，并且不久将有充足的浓缩铀来造第二颗核弹。

据英国《每日电讯报》3日报道，以色列军方情报负责人Maj-Gen Amos Yadlin在对以色列议会外交关系与国防委员会所作的证词中做出了上述言论，这与美国中情局和国际原子能机构之前的判断一致。

报道称，这名负责人的言论十分重要，因为怀疑伊朗核开发项目的以色列不会排除采取军事行动以防止伊朗开发核武器。

个人简历、在学期间发表的学术论文与研究成果

个人简历

1984 年 11 月 01 日出生于山东省新泰市。

2002 年 9 月考入清华大学计算机科学与技术系计算机科学与技术专业，2006 年 7 月本科毕业并获得工学学士学位。

2006 年 9 月免试进入清华大学计算机科学与技术系攻读博士学位至今。

发表的学术论文

[1] **Zhiyuan Liu**, Xinxiong Chen, Maosong Sun. A Simple Word Trigger Method for Social Tag Suggestion. The Conference on Empirical Methods in Natural Language Processing (EMNLP'11), 2010. Oral presentation.

[2] **Zhiyuan Liu**, Xinxiong Chen, Yabin Zheng, Maosong Sun. Automatic Keyphrase Extraction by Bridging Vocabulary Gap. The 15th Conference on Computational Natural Language Learning (CoNLL'11), 2011. Poster paper.

[3] **Zhiyuan Liu**, Yabin Zheng, Lixing Xie, Maosong Sun, Liyun Ru, Yang Zhang. User Behaviors in Related Word Retrieval and New Word Detection: A Collaborative Perspective. ACM Transactions on Asian Language Information Processing (ACM TALIP) (Special Issue on Chinese Language Processing), 2011.

[4] **Zhiyuan Liu**, Yuzhou Zhang, Edward Y. Chang, Maosong Sun. PLDA+: Parallel Latent Dirichlet Allocation with Data Placement and Pipeline Processing. ACM Transactions on Intelligent Systems and Technology (ACM TIST) (Special Issue on Large Scale Machine Learning), 2010.

[5] **Zhiyuan Liu**, Wenyi Huang, Yabin Zheng, Maosong Sun. Automatic Keyphrase Extraction via Topic Decomposition. The Conference on Empirical Methods in Natural Language Processing (EMNLP'10), 2010. Oral presentation.

[6] **Zhiyuan Liu**, Maosong Sun. Domain-Specific Term Rankings Using Topic Models. The Sixth Asia Information Retrieval Society Conference (AIRS'10), 2010. Full paper. EI Index: 20110213561318.

[7] **Zhiyuan Liu**, Chuan Shi, Maosong Sun. FolkDiffusion: A Graph-based Tag Sug-

gestion Method for Folksonomies. The Sixth Asia Information Retrieval Society Conference (AIRS'10), 2010. Poster paper. EI Index: 20110213561294.

[8] **Zhiyuan Liu**, Peng Li, Yabin Zheng, Maosong Sun. Clustering to Find Exemplar Terms for Keyphrase Extraction. The Conference on Empirical Methods in Natural Language Processing (EMNLP'09), 2009. Regular paper.

[9] **Zhiyuan Liu**, Yabin Zheng, Maosong Sun. Quantifying Asymmetric Semantic Relations from Query Logs by Resource Allocation. The 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'09), 2009. Regular paper. EI Index: 20093012218189.

[10] **Zhiyuan Liu**, Maosong Sun. Asymmetrical Query Recommendation Method Based on Network-resource-allocation Dynamics. The 17th International World Wide Web Conference (WWW'08), 2008. Poster paper. EI Index: 20085111784830.

[11] Yabin Zheng, Lixing Xie, **Zhiyuan Liu**, Maosong Sun, Yang Zhang, Liyun Ru. Why Press Backspace? Understanding User Input Behaviors in Chinese Pinyin Input Method. The 49th Annual Meeting of the Association for Computational Linguistics and Human Language Technologies (ACL-HLT'11), 2011. Short paper.

[12] Xiance Si, **Zhiyuan Liu**, Maosong Sun. Modeling Social Annotations via Latent Reason Identification. IEEE Intelligent Systems, Vol. 25, No. 6, pp. 42-49, 2010. SCI Impact Factor: 3.144.

[13] Xiance Si, **Zhiyuan Liu**, Maosong Sun. Exploring Subsumptioqn Relations in Social Tags. The 23rd International Conference on Computational Linguistics (COLING'10), 2010. Regular paper.

[14] Yabin Zheng, **Zhiyuan Liu**, Lixing Xie. Growing Related Words from Seed via User Behaviors: A Re-ranking Based Approach. ACL Student Research Workshop, 2010. Poster paper.

[15] Xiance Si, **Zhiyuan Liu**, Peng Li, Qixia Jiang, Maosong Sun. Content-based and Graph-based Tag Suggestion. ECML/PKDD Discovery Challenge Workshop (RSDC'09), 2009. Regular paper.

[16] Yabin Zheng, **Zhiyuan Liu**, Shaohua Teng, Maosong Sun. Efficient Text Classification Using Term Projection. The 5th Asia Information Retrieval Symposium (AIRS'09), 2009. Poster paper. EI Index: 11251312.

[17] Yabin Zheng, **Zhiyuan Liu**, Maosong Sun, Liyun Ru, Yang Zhang. Incorporating

User Behaviors in New Word Detection. The 21st International Joint Conference on Artificial Intelligence (IJCAI'09), 2009. Oral paper.

[18] 郑亚斌, 曹嘉伟, **刘知远**. 基于最大匹配和马尔科夫模型的对联系统. 第四届全国学生计算语言学研讨会 (SWCL'08), 2008.

[19] Yabin Zheng, Shaohua Teng, **Zhiyuan Liu**, Maosong Sun. Text Classification Based on Transfer Learning and Self-Training. The 4th International Conference on Natural Computation (ICNC'08), 2008. EI Index: 10398646.

[20] **刘知远**, 郑亚斌, 孙茂松. 汉语依存句法网络的复杂网络性质. 复杂系统与复杂性科学, Vol. 5, No. 2, pp. 37-45, 2008.

[21] **刘知远**, 孙茂松. 汉语词同现网络的小世界效应和无标度特性. 中文信息学报, Vol. 21, No. 6, pp. 52-57, 2007. 中文核心期刊.

[22] **刘知远**, 司宪策, 郑亚斌, 孙茂松. 中文博客标签的若干统计性质. 第七届中文处理国际会议 (ICCC'07), 2007.

[23] 郑亚斌, **刘知远**, 孙茂松. 中文歌词的统计特征及其检索应用. 中文信息学报, Vol. 21, No. 5, pp. 61-67, 2007. 中文核心期刊.

[24] **刘知远**, 孙茂松. 基于WEB的计算机领域新术语的自动检测. 第九届全国计算语言学学术会议 (CNCCL'07), 2007.

参与的科研项目

[1] 国家863计划项目：大规模网络图文数据的语义分类及适度理解。项目编号：2007AA01Z148。立项部门：中国人民共和国科技部。时间：2007-2009。

[2] 国家自然科学基金项目：汉语复杂网络的性质、结构、演化及其典型应用研究。项目编号：60873174。立项部门：国家自然科学基金委。时间：2009-2011。

研究成果

[1] 郑亚斌，**刘知远**，孙茂松，茹立云，张杨。获取新词的方法和装置。申请号：200910083143.2。公开号: CN101539940。