密级：_____

博士学位论文

短文本话题建模

作者姓名：晏小辉

指导教师：程学旗　研究员

中国科学院计算技术研究所

学位类别：工学博士

学科专业：信息安全

培养单位：中国科学院计算技术研究所

2014年5月

Topic Modeling over Short Texts

By

Yan Xiaohui

A Dissertation Submitted to

Graduate University of Chinese Academy of Sciences

In partial fulfillment of the requirement

For the degree of

Doctor of Philosophy

Institute of Computing Technology

Chinese Academy of Sciences

May,2014

声　　明

我声明本论文是我本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，本论文中不包含其他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

作者签名：　　日期：

论文版权使用授权书

本人授权中国科学院计算技术研究所可以保留并向国家有关部门或机构送交本论文的复印件和电子文档，允许本论文被查阅和借阅，可以将本论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编本论文。

（保密论文在解密后适用本授权书。）

作者签名：　　导师签名：　　日期：

摘　　要

随着近年来Web2.0技术和社交媒体的兴起，短文本信息如微博、评论、状态信息等在互联网上越来越多。这些短文本信息通常由大量用户所产生，其长度虽短，但规模大、更新快、内容更丰富多样。海量的短文本数据中蕴含着丰富的有价值信息，但如何挖掘这些信息，考验着目前计算机对短文本自动理解与处理的能力。

提高计算机对短文本智能化处理水平的一个难点在于如何挖掘文本背后的语义知识。目前常用的一种方法是话题模型，其通过建模文档集合中潜在的话题结构来自动分析文本的语义。过去十年间，关于话题建模技术的研究取得了很多进展，但大多是基于长文本数据，并没有考虑到短文本的特殊性。在实际应用中，这些短文本数据不仅内容稀疏，而且高速增长，话题内容在不断更新演化，新话题也在不断涌现。这给现有的话题建模技术带来了很多新的挑战。

受日益增长的短文本语义分析需求驱动，本文以微博等实际应用为背景，对短文本话题建模技术展开了研究。本文重点关注了目前实际应用中短文本话题建模的三个重要问题：内容稀疏问题，动态演化问题和突发涌现问题。具体研究内容包括：

首先，针对短文本的内容稀疏问题，我们分析了传统话题建模方法的不足，即其过于依赖文档内部的词共现信息来学习话题，而短文本由于文档过短，其内部词共现信息严重不足。为了克服这一问题，本文提出了一种新的话题学习思路，即利用全局中的词共现信息来学习话题，从而弥补文档内部词共现信息稀疏的缺陷。基于这个思想，我们提出了一种新的概率话题模型，即双词话题模型（Biterm Topic Model或BTM）。该模型通过直接建模文档集合中双词（即共现词对）的产生来学习话题，从而避免了受短文本文档过短导致的内容稀疏性问题。和现有的方法不同，该模型对短文本话题的学习无需借助任何外部数据或信息。这也是目前首个通用的短文本话题模型。

其次，在线应用中的短文本话题数据动态增长，内容也在不断演化。为了适应这种在线短文本数据上话题建模的需要，本文对在线话题建模方法展开了研究。我们在双词话题模型的基础上提出了两种在线学习算法：oBTM 算法和iBTM 算法。它们通过使用一小部分最近的历史数据来增量更新模型，大大降低了模型更新所需的时间和空间复杂度，同时还能及时地追踪数据中话题的动态变化。

最后，在微博等流式短文本数据中每天都会有大量的突发话题涌现。为了发现这些突发话题，本文研究了适应短文本数据流中的突发话题建模方法。我们通过分析突发话题与突发双词之间的联系，认识到：突发性强的双词，更可能是由突发话题所产生；而突发性弱的双词，更可能是由普通话题所产生。基于此，本文提出了一种突发话题模型，即组合双词话题模型(Composite Biterm Topic Model或CBTM)。CBTM对

数据中的突发话题和普通话题分别建模，并利用双词的突发性来指导不同类型话题的学习，从而自动地学习到突发话题。

通过以上研究，本文提出了一套针对短文本话题建模的新方法，为短文本话题建模提供了新的思路。同时，本文的研究紧紧围绕实际应用需求，具有广泛的应用价值。然而，和长文本话题建模研究相比，目前关于短文本建模的研究仍处于初步阶段，在实际应用过程当中仍有很多问题需要解决，希望本文的研究能推动这一领域的发展。

关键词：话题模型，短文本，文本聚类，突发话题检测，在线学习，文本挖掘，语义分析

Topic Modeling over Short Texts

Yan Xiaohui (Information Security)
Directed by Professor Cheng Xueqi

With the rise of Web2.0 technology and social media in recent years, short texts such as microblogs, comments and status messages, are prevalent on the Web. These short texts are usually generated by millions or even billions of users. Though their length is short, their scale is very large with fast-changing and diverse content. The Massive short texts contains a wealth of valuable information. But to mining these information, it tests the ability of our computes in understanding and processing these short text data.

One major challenge to improve the ability of computer in processing short texts is how to grasp the semantics behind them. Topic models are widely used tools for this task, which can automatically analyze the semantics of text by modeling the latent semantic structure of a document collection. Over the past decade, research on topic models has made a lot of progress, but most of them are based on normal texts without considering the speciality of short texts. In practical applications, the content of short texts are very sparse and large-scale. Moreover, topics are continually evolving, and new topics are emerging. It brings many new challenges to existing topic modeling techniques.

Driven by the growing demand of semantic analysis for short text, this article takes real-world applications such as microblog as a background to study topic modeling over short texts. Mainly speaking, we focuses on three key issues of topic modeling in practice: the content sparsity issue, the dynamic evolution issue, and the topic emergence issue. Specifically, the major studies include:

First, for the content sparsity problem of short text, we find traditional topic models overly rely on the document-level word co-occurrences to learn topics, which is sparse for short texts. To overcome this problem, this article proposes a novel idea to learn topics by exploiting rich global word co-occurrences, rather than the sparse document-level ones. Based on this idea, we propose a probabilistic topic model, called Biterm Topic Model (BTM). BTM learns topics by modeling the generation of biterms (i.e., unordered co-occurring word pairs) in the document collection, thus avoid the content sparsity issue over short texts. Different with existing methods, BTM does not rely on any external data or information. To our best of knowledge, BTM is the first topic model for general short texts.

Second, for the large-scale and dynamic short text data, we study the issue of online

topic modeling for short texts. We propose two online algorithms for BTM, i.e., oBTM and iBTM. Both of them can incrementally update the model with only a small portion of recent data, that greatly reduces the time and space complexity. Meanwhile, they are capable to track the dynamic changes of topics timely.

Finally, considering there are lots of topics emerging in short text streams like microblogs, we further study the problem of bursty topic modeling over short text streams. Through analyzing the connection between bursty topics and bursty biterms, we recognize that: A bursty biterm is more likely to be generated by a bursty topic; Conversely, a non-bursty biterm is more likely to be generated by a common topic. Based on this idea, we propose a novel way to model bursty topics using a model called composite biterm topic model (CBTM). CBTM models two types of topics, i.e., bursty topics and common topics, separately, and then leverages the burstiness of biterms to guide the learning of bursty topics. Consequently, the model can discover bursty topics automatically.

Through these studies, we propose a series of novel methods for short text topic modeling. These methods can be applied to a wide range of real-world shortt text related applications. However, compared to normal text topic modeling, the research of short text topic modeling is still at a preliminary stage. There still remains many problems to be studied. We hope this article can promote research in this field.

Keywords: Topic models, short text, clustering, topic detection, online learning, text mining, semantic analysis

目　　录

图 目 录

表 目 录

第一章　引言

互联网的快速发展加速了信息更新与传播的速度，同时也带来了信息内容上变化。从Web1.0到Web2.0，再到社交媒体的兴起，互联网上信息发布的门槛越来越低，普通用户参与信息创作的兴趣也越来越高。在很多应用当中，普通用户并不注重消息的质量，其发布的消息通常都很随意且简短，如微博、状态信息、评论等。虽然这些消息非常短，但由于用户数量多且消息发布更为频繁，导致互联网上短文本信息的规模非常巨大。海量的短文本数据中给文本挖掘领域带来了新的机遇，也带来了新的挑战。

1.1　研究背景

1989年，在欧洲粒子物理研究所工作的Tim Berners-Lee及其同事发明了万维网（World Wide Web）。人们可以通过编写HMTL网页然后在万维网上发布自己的信息，并通过超链接来和实现网页之间的相互连接。万维网的出现带来了信息爆发式增长，此后越来越多的公司通过万维网来宣传自己的产品，越来越多的机构和个人通过万维网发布自己的信息，越来越多的商业网站通过万维网提供自己的服务等。在早期的万维网时代，即Web1.0时代，网页都是以信息公布与分享为目的，普通用户通常只能通过浏览网页来获取信息，对网页内容没有发言权。通常，这些网页的内容需要专门的人员去撰写和制作，内容都比较规范完整，文档长度较长，如新闻、公司介绍、资讯等。

随着Web2.0时代的来临，互联网不仅仅再是一个信息发布与共享平台，而转变成了一个更开放的内容协作平台。用户既能从网站上获取信息，同时还能参与到网站内容的创作过程中来。典型的Web2.0网站如博客，维基百科，问答社区等，用户在这些网站中积极发表自己的观点，分享信息等。大量的用户产生内容使得互联网上的文本信息规模也大量增长。其中既包含长文本信息，如博客、维基百科等，也包含很多短文本信息，如在问答社区中的问题与回帖，评论信息，即时通讯信息等。

近年来，社交网站的兴起更是极大地激发人们发布与分享信息的兴趣，越来越多的网民加入到这些社交网站中成为其内容贡献者。图1.1展示了最近几年两个国外主要的社交网站Facebook[1]和Twitter[2]的月活跃用户的增加情况（来源于其官方数据）。其中，Fackbook在2013年的时候月活跃用户数已经超过了12 亿，接近全球人口的1/5；而Twitter在2014年初的时候月活跃用户数目也达到了2.4亿。此外，用户在社交网站上花费的时间也较长。根据Globalwebindex最新调查，超过44%的网民每天在社交网络花费的时间超过1个小时，而只有略高于10%的网民基本不光顾社交网络网站。在这些社

[1]http://facebook.com

[2]http://twitter.com

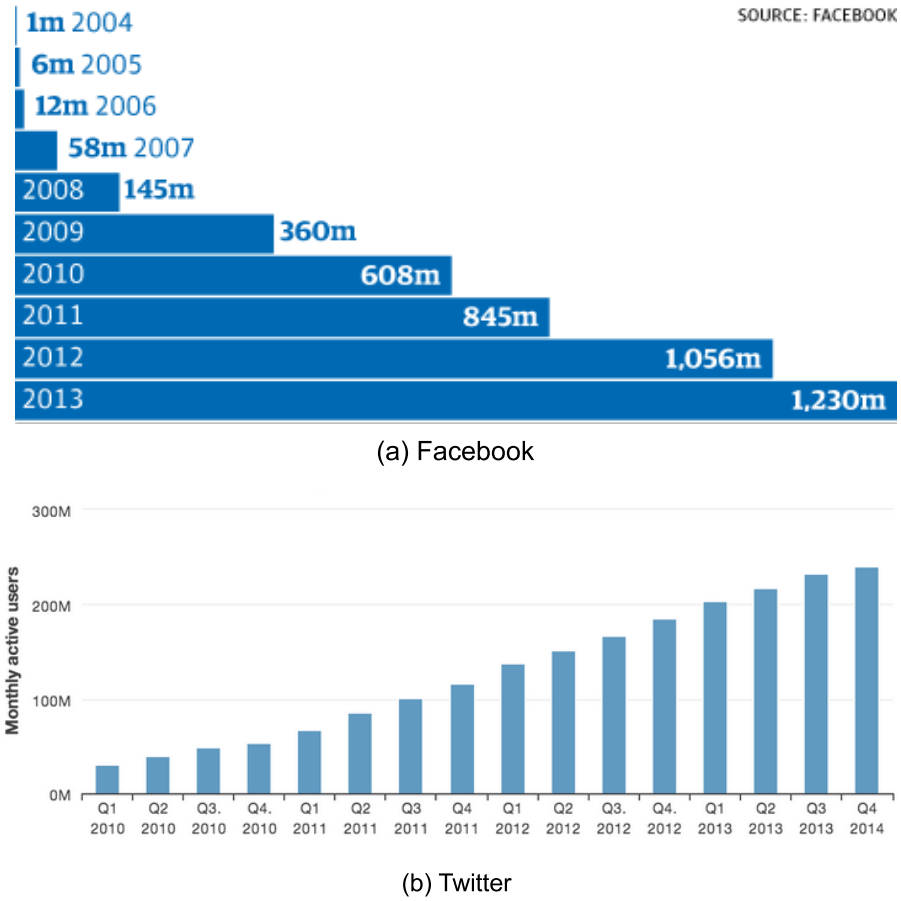


图 1.1: 最近几年Facebook和Twitter的月活跃用户增长情况

交网站中，人们除分享信息之外，更注重人与人之间的交流，因此其内容都比较随意，而且简短。如Fackbook的状态信息通常就只有一两句话，而Twitter则干脆限制其消息最大长度不能超过140个字符。

在当今的互联网上，短文本信息的流行有其必然性，即适应信息产生与传播速度不断增长的要求。从信息发布者角度看，发布一篇长文本文档需要耗费较长时间和精力。而短文本信息则风格比较随意，编写简单，发布起来没有任何门槛限制。特别是近年来移动互联网的高速发展，使得短文本消息的发布更为便捷。其次，从信息接收者角度看，短文本对信息的表达比长文本简约紧凑、来源更丰富，使得用户可以更快更多的获取和消化信息。最后，从消息传播的角度上看，短文本信息依托于社交网络，短文本具有更快的传播速度和更广的传播途径。因此，在生活节奏不断加快的今天，短文本信息逐渐成为人们在互联网上信息交流与传播的一种主流形式。

互联网上海量的文本数据是一座有待开采的金矿，其中蕴含着丰富的有价值信息。比如微博被称为分布式的“人类的传感器”[125]，实时地记录着个人和社会息息相关的信息，自动问答系统中的问题与回答蕴含了大量经验知识，电子商务网站的评论中包含大量的用户反馈信息等。这些信息对很多应用都有重要意义，下面我们简单举几

个例子。

- **舆情监控**　Web2.0技术的发展，尤其是社交媒体的出现为人们提供了便利的言论场所。人们也乐于在网上，如论坛、博客、微博等网站，发表和分享自己的意见与看法。但总有一些不法分子恶意散播谣言、诈骗、暴动等有害社会的信息，通过舆情监控软件检测出这些敏感信息，防范危机于未然，对社会的安定具有重要的意义。另外，舆情监控还能帮助决策者掌握社会动态和民众意见，以便更好地做出决策。

- **用户行为分析**　互联网上的短文本信息大部分都是用户产生内容，因此这些信息和用户非常相关。在用户为王的互联网时代，如何掌握用户需求和行为特征，对很多互联网公司至关重要。我们可以通过分析用户的微博，来判断用户的行为特征、兴趣爱好，甚至是潜在意图。基于这些用户信息，我们可以对用户提供个性化的服务，如精准营销等。

- **商业情报收集**　我们可以从微博、评论等短文本信息中收集市场对产品的反馈信息、行业资讯、竞争对手的动态信息等，以帮助客户掌握市场动态，完善经营。

- **信息推荐与过滤**　海量的短文本信息也带来严重的信息过载问题。从中过滤无用信息，把有价值的信息推荐给用户，可以帮助用户更全面地、更快捷地获取有用信息。

然而，从这些短文本中挖掘有价值的信息并非一件容易的事情。首先，短文本由于长度短。根据我们对实验数据的统计，经过预处理之后，百度问答中的问题的平均长度（即词个数）为3.94，tweet的平均长度为5.21，微博的平均长度为5.87。我们可以看到这些短文本中内容非常稀疏，上下文信息严重不足，再加上其编写缺乏规范和约束，通常包含很多错别字、新生词、垃圾信息等，给文本语义分析带来了很大的困难。

其次，互联网上短文本数据动态性极强。大量活跃用户每天源源不断的产生新的信息，如Fackbook上每20分钟就有3百万条消息产生[3]，而在Twitter上平均每天发帖量超过5800万[4]。这些大规模短文本信息涉及的话题五花八门，如个人的喃喃自语，朋友之间闲聊，新闻事件报导，热门话题讨论，明星八卦，生活资讯等等。随时间的变化，其中有些话题会动态演化，比如关于时尚、经济以及一些长期事件的新闻话题（如美国总统选举）等话题；而有些话题会突然涌现，如一场NBA篮球比赛、某部电影上映、某个会议开幕以及一些突发事件（如恐怖袭击）等；但也有些话题随时间变化不大，如天气、星座、情绪表达等和日常生活相关的一些话题。如此复杂多变且规模急速增长的内容也给文本挖掘和处理带来很大的挑战。

[3]http://www.statisticbrain.com/facebook-statistics/

[4]http://www.statisticbrain.com/twitter-statistics/

1.2 研究现状

本节中我们回顾过去文本挖掘相关领域中与短文本语义分析与处理相关的研究工作。

由于Web2.0时代之前，短文本信息在互联网上并非主流，与短文本语义分析与处理相关的研究也并不多见。一个相关的研究方向就是信息检索领域中对查询（query）理解和处理。由于通常查询的长度在2-4个词之间[6]，查询也是一种典型的短文本。信息检索的一个核心问题就是计算查询和文档之间的匹配程度。早期的信息检索中，主要基于向量空间模型[127]或者统计语言模型[127]来计算查询和文档之间词的匹配度。这种简单的处理方式只能搜索到文档中包含至少一个查询词的相关文档，无法检索到那些语义上相关但和查询的词汇并不匹配的文档。于是，这对用户构造查询的能力就是一个很大的考验。为了解决此问题，研究者们提出了相关反馈技术[98]。其主要思想是先用基本检索方法（如向量空间模型或统计语言模型）获取一些文档集合，然后让用户标记出其中的相关文档，再利有相关的文档信息扩充原查询，进行二次检索。该方法能部分克服查询与文档词汇不匹配问题，但人工标注及二次检索会严重影响用户的体验。于是一种折中的办法就是直接取基本检索方法返回的前几个文档来扩充，该方法称为伪相关反馈技术[98]。Metzler等人[104]尝试了用词、短语、外部语料扩充等多种方式来丰富查询的表达，然后用来计算查询与查询或文档之间的相似度。Sahami和Heilman[124]提出了一种核方法来计算查询之间的相似度，思想是利用查询的返回结果集合中的文档片段来对查询进行扩充。Xu等人[158]利用Wikipedia对查询进行扩充。从以上研究中我们可以看到，在信息检索领域中最常用的处理短文本数据稀疏性的方法就是数据扩充，包括利用相关文档、外部语料以及查询内部的各种特征等信息来丰富短文本的表达。

进入Web2.0时代，随着短文本数据在互联网应用中的增多，短文本挖掘相关工作也逐渐受到重视，研究者们尝试了很多种方法来改进短文本语义分析与处理。例如，Banerjee等人[12]用Wikipedia中最相关的前10个文档来扩充每个短文本文档的表示，再对短文本文档聚类。Gabrilovich等人[57]提出一种显式语义分析（Explicit Semantic Analysis）技术，将短文本表示成Wikipedia中的概念向量，以此来改进短文本相似度计算。其基本思想仍然是利用Wikipedia语料扩充了短文本的表示。Hu等人[75]同时利用内部特征（如短语）和外部特征（如Wikipedia和WordNet）来对短文本进行扩充，以此来提高短文本聚类的效果。Song等人[132]利用开放网页构建一个大型概率化知识库，以此来推断短文本中的概念表示，然后再聚类。Yan等人[160]提出了一种基于词共现关系的词权重计算方式，改进了NMF在短文本聚类上的效果。该方法的好处是只需利用短文本数据内部信息，无需利用外部知识库。在短文本分类方面，Phan等人[41, 115]通过在Wikipedia上学到话题模型推断短文本文档的话题表示，并以此来扩充短文本文档的表示，再去分类。Yu等人[164]开发了一个短文本SVM分类器LibShortText，除了词

之外，还利用了短语特征来表示文档。另外，Ferragina等人[55]研究了用Wikipedia中的实体对短文本进行自动语义标注的问题。

近年来，社交媒体的兴起也带动了对短文本语义分析与处理的研究。在这些工作中，研究者为了克服短文本内容稀疏性问题作出了很多尝试。例如，很多研究者将多条微博聚合在一起，形成一个虚拟长文档，再用话题模型对其进行语义分析[155, 169]。Hong等人[73]对比普通LDA，以及按用户和按词来聚合微博再学习LDA等不同种话题学习方式，发现按用户聚合微博后再训练LDA的效果最好。Mehrotra等人[101]尝试了更多种聚合方式，如按时间，按突发词，按hashtag等，其结论是按Hashtag聚合的效果更好。这种微博聚合的方式，实际上可以看成是利用内部数据来扩充了原短文本文档的表示。另外，也有研究者尝试利用微博中各种用户标注信息如Hahstag，表情符号，功能标志符号（如“@”，“Reply”）等来指导话题学习[119]。这种方法本质是一种多标签分类方法，只能学习到指定类型的话题，不能发现潜在话题。Jin等人[79]提出一种话题知识迁移学习方法，利用微博中URL所指向网页中的长文本信息来辅助Tweets中的话题学习。但该方法只能改进那些有URL的微博内的话题学习，而大部分微博中并不包含URL。

总结以上工作，我们发现目前对短文本语义分析和处理的主要手段可分为三种：

- 利用外部数据扩充文档的表示，如借用Wikipedia、搜索结果或者其他辅助数据。这种方式的效果取决于原短文本文档与扩充的外部数据的相关程度。有时候要找到合适外部数据源是很困难的。比如微博，其内容实时性很强，其语言表达方式也很其他长文本文档（如百科）有很大差异。此时，盲目的扩充反而会带来影响原短文本文档的语义。

- 利用内部数据扩充文档的表示，如伪相关反馈、加入短语特征，微博聚合等方式。这种方式在目前微博相关地研究中的比较多，其优点是不会引入一些异质数据源中的噪音。但是作为一种启发式做法，如何扩充效果好，并没有一个通用的准则。如果扩充的不好，同样可能给原文档的语义带来大的偏差。

- 利用用户标注信息来表达短文本的语义。和上两种手段不同，这种方法并没有解决短文本的内容稀疏性问题。而且用户标注信息通常非常少，所以该方法的作用范围比较有限。

我们可以看出，目前对短文本的语义分析与处理的方法仍有很多局限性。无论是利用外部数据，还是内部数据扩充的方法都依赖于一些启发式的策略，并没有作出原则性的改进。

1.3 本文工作

1.3.1 研究目标与内容

为了提高短文本语义分析与处理的水平，本文研究短文本话题建模方法。话题建模方法通过建模文档集合中潜在的话题结构来分析文本中的语义信息。虽然在过去十年，话题建模一直是机器学习领域中的一个研究热点，对其的研究也取得了很多进展。但这些工作都是基于长文本的。但当今互联网上的短文本数据和以往的长文本数据的特性有很多差异。首先，短文本中每个文档非常短，其内容非常稀疏；其次，互联网上的短文本规模大而且持续增长，其内容也在不断演化；另外，在微博等流式短文本数据中，每天都会有很多突发话题涌现。以往的话题建模研究并没有充分考虑到短文本数据的这些特性，因此并不适合直接用于短文本数据。

为了处理日益增长的短文本数据，研究合适的短文本话题建模方法势在必行。为此，本文中以微博等互联网应用为背景，研究如何更好去对这些短文本数据进行话题建模，以提高对这些短文本语义分析的水平。具体地说，我们主要研究的挑战性科学问题如下：

1. **内容稀疏问题** 传统的话题建模方法的研究基本上都以长文本为研究对象，如新闻、网页、科技文献等。这些方法通常建模的是文档的产生过程，即假设一个文档中包含多个话题，且文中每个词分别来自一个话题。这种假设容易受短文本的内容稀疏性影响。考虑到单个短文本文档通常只包含几个或十几个词，无论是词频信息还是词共现信息非常缺乏，要基于如此有限的信息来推断出文档内部的话题结构非常困难。另一方面，在自然语言当中存在大量的多义词和一词多用的情况，再加上短文本中很多用词并不规范，加剧了我们对短文本话题建模的难度。

2. **动态演化问题** 在线互联网应用中的短文本数据，如微博中，众多用户每时每刻都会产生大量的短文本消息。这些短文本数据不仅规模大，实时性强，内容也在不断演化。这使得原有静态模型加批处理计算方式难以跟得上数据的变化。为了适应这种大规模动态短文本数据中话题建模的需求，我们研究了针对短文本数据流的在线话题建模方法，以提高算法的可伸缩性和其对新数据的的响应能力。

3. **突发涌现问题** 在微博等短文本数据流中，每天都会有大量的突发话题涌现。发现突发话题对舆情监控、商业情报收集、消息推荐等有重要的意义。突发话题是一类特殊的话题，传统的话题建模方式并没有考虑到话题的突发性，无法自动学习突发话题。于是，我们研究针对短文本数据流中的突发话题建模问题，以便自动学习这些突发话题。

1.3.2 研究成果

我们对以上研究问题展开了详细地研究，并取得了一系列的研究成果：

1. 提出了一种通用的短文本话题建模方法

   我们发现传统话题模型在短文本上效果不佳的一个主要原因是其过于依赖文档内部的词共现信息来学习话题，在文档过短时，容易受数据稀疏性影响。为了克服这一问题，我们提出了一种新的话题学习思路——利用全局丰富的词共现关系去学习话题。为此，我发设计了一种新的概率话题模型，即双词话题模型（Biterm Topic Model或BTM）。该模型通过直接建模文档集合中双词（即共现词对）的产生来学习话题，从而避免了受短文本文档过短导致的内容稀疏性问题。和现有的方法不同，该模型对短文本话题的学习无需借助任何外部数据或信息。这也是目前首个通用的短文本话题模型。在真实短文本数据集上的实验表明，BTM对短文本话题学习的效果要显著好于现有方法。

2. 提出了两种基于双词话题模型的在线话题学习算法

   针对短文本中话题动态演化问题，我们基于BTM提出了两种在线话题学习算法：oBTM和iBTM。其中，oBTM将短文本数据流切分成多个时间片序列，每次只需在最新时间片内的数据上来学习BTM，每个时间片上的话题学习结果会通过先验的方式传给后续时间片。iBTM则通过维护一个固定长度时间窗口内的最近历史数据，每接收一个文档对这部分数据进行随机采样更新。这两种在线算法的共同特点就是：1）模型可以增量更新，而且只需用到部分最近的历史数据，能使模型更新所需要的时间和内存开销降低到常数级别；2）由于是采用最近历史数据更新模型，因此更新后的模型能自适应数据中话题的演化。通过在大规模微博数据上的实验表明，这两个在线话题学习算法的效果和批处理算法相差不大，但效率要高很多。同时，我们也验证这两个算法追踪流式数据中话题演化的能力。这表明它们能很好的适应在线话题建模需求。

3. 提出了一种短文本突发话题学习模型

   针对短文本中话题突发涌现性问题，本文提出了一种突发话题模型，即组合双词话题模型（Composite Biterm Topic Model或CBTM）。通过分析突发话题与突发双词之间的联系，我们认识到：突发性强的双词，更可能是由突发话题所产生；而突发性弱的双词，更可能是由普通话题所产生。基于此，CBTM对数据中的突发话题和普通话题分别建模，并利用双词的突发性来指导不同类型话题的学习，从而自动地学习到突发话题。与目前其他突发话题发现方法相比，该模型无需任何后处理手段和启发式技巧，而且学习到的突发话题更准确、全面，话题可解释性更好。

1.4 论文组织

本文内容的组织如下：

第一章介绍了本文的研究背景以及现状、本文的研究目标与内容、以及取得的研究成果。

第二章对话题学习方向的研究进行了综述，为后面章节提供必要的背景知识。我们首先回顾话题学习方法的发展历史，然后介绍了概率话题模型的统计推断方法，以及目前常用一些话题学习评价方法和相关应用。

第三章详述了双词话题模型，包括模型定义、参数估计方法、文档中话题比例推断，以及其在真实短文本数据集上的实验结果。

第四章研究了针对大规模动态短文本数据中的在线话题建模，介绍两种基于双词话题模型的两种在线话题学习算法，以及它们在大规模数据集上的实验效果。

第五章研究了短文本数据流的突发话题建模方法，即组合双词话题模型。首先，我们介绍了该模型的原理与定义；随后，给出了模型参数估计算法。最后，我们在Twitter数据集上验证了该模型的效果。

第六章对整个研究工作进行了总结，阐明了本论文的主要贡献和创新，并在最后对未来的研究工作进行了展望。

第二章　话题建模综述

本章首先回顾话题建模方法发展的历史，然后介绍概率话题模型中的统计推断方法，还有目前常见的一些话题模型结果评价方法。最后，介绍了话题模型在文本挖掘相关领域中的一些应用。

2.1　发展历史

文本的表示是文本信息处理中的一个首要问题。在早期的信息检索和自然语言处理的研究中，文本通常用向量空间模型[127]和统计语言模型[117]来表示。向量空间模型基于线性代数理论，它把一个文档表示成词空间中的一个向量，其中每个维度对应于一个词；而统计语言模型则是基于概率论，它把文档看成是从词汇集合中采样出来词序列。虽然二者的支撑理论不同，但其共同点就是直接用词来表示文本，每个词看成是一个语义单元。这种基于词的文本表示方式数学形式简单、容易计算，而且在实际应用当中效果也不错。因此，在信息检索、自然语言处理和文本挖掘等领域中被广泛使用。

然而随着文本信息处理水平的提高，人们逐渐认识到基于词的文本表示并不能准确、完整的表达文档的语义，这种缺乏语义的表示严重制约着计算机对文本信息理解和处理的智能化水平。具体而言，基于词的文本表示方式存在两个根本性问题，即同义词问题和多义词问题[50]。同义词问题指的对于同一个事物，由于用户的背景以及上下文环境不同，往往会有很多种不同的表达方式。比如，“高兴”和“快乐”两个词的含义原本是一致的，但在基于词的表示方式中，这两个词分属不同的维度。而多义词问题则相反，指同一个词在不同的场景下表达的含义不一致。比如，“qq”可能指某通信软件，也可能是某汽车品牌。同义词和多义词问题在自然语言中很常见，给文本信息处理带来很大的困难。如在信息检索中，同义词问题会导致某些相关的文档被忽略，降低检索结果的召回率；而多义词问题会导致某些不相关的文档被检索出来，从而降低检索结果的准确率。

于是，人们开始探索更高级的文本表示方式，即能刻画出文本内部的语义信息的表达方式，从而提升计算机对文本信息的理解和处理的智能化水平。

2.1.1　潜在语义分析

一个突破性的工作是Deerwester等人在1990年提出的潜在语义分析（Latent Semantic Analysis或LSA）[50]。LSA首次提出了一种完全以数据驱动的方式学习文档内部的语义结构的方法。它假设文本数据中存在某种潜在语义结构，但由于噪音的影响，

表现在词上的语义信息并不准确。为了去除这些噪音，LSA 采用对词-文档矩阵（即每行表示一个词，每列表示一个文档，每个元素代表一个词在相应文档中的权重）进行奇异值分解（Singular Value Decomposition或SVD），然后做低秩估计。

LSA的具体形式如下。令$\mathbf{X}$表示词-文档矩阵，对$\mathbf{X}$进行奇异值分解,

$$\mathbf{X} = \mathbf{T}_0\mathbf{S}_0\mathbf{D}_0^T,$$

其中$\mathbf{T}_0$和$\mathbf{D}_0$分别为$\mathbf{X}_0$的左、右奇异向量构成的正交矩阵，$\mathbf{S}_0$ 为$\mathbf{X}_0$的奇异值按从大到小排列所构成的对角矩阵。假设潜在语义空间的维度为K（K小于$\mathbf{X}$ 的秩）, LSA对$\mathbf{X}$做秩为K的低秩估计，得到：

$$\hat{\mathbf{X}} = \mathbf{TSD}^T,$$

其中$\mathbf{S}$为$\mathbf{S}_0$的前K个非零元素构成的对象矩阵，$\mathbf{T}$和$\mathbf{D}$分布为$\mathbf{T}_0$和$\mathbf{D}_0$ 只保留前K 列所构成的矩阵。$\mathbf{T}$和$\mathbf{D}$中的每行分别可看成是词和文档在潜在语义空间上的表示。用变换后的词-文档矩阵$\hat{\mathbf{X}}$去计算文档与文档、文档与词、或词与词之间的内积相似度，可转换成词和文档在潜在语义空间上的向量表示的（经过$\mathbf{S}^2$缩放之后的）内积。

LSA通过SVD可以学习到文档中潜在语义表示，暗含的假设是经常共现的词，应该总是在同一个文档中出现。因此，去噪后文档的表示，除了包含原有的一些词之外，还包含和它们经常共现的词，即使这些词之前并未在该文档中出现。由于经常在一起共现的词，通常语义上都比较相关，所以LSA实际上相当于对文档做了相关词扩充，从而可以解决信息检索中的同义词问题。

LSA方法在文本自动语义分析方向做出开拓性的贡献，其利用词共现关系去学习文本中潜在语义空间的思想直接启发了众多后续的研究。然而，LSA方法存在以下问题，导致其在实际应用并不多见：1）LSA虽然能解决同义词问题，但对多义词问题并没有很好的解决；2）SVD分解过程中可能产生负值，难以解释其物理意义；3）SVD计算复杂度太高（$O(\min\{mn^2, m^2n\})$, m和n为词-文档矩阵的维度），限制了LSA 适用的数据规模。以上三个问题，都是由于SVD分解的局限性造成的。后续对LSA的改进工作都采用了不同的分解方法，我们根据其方法的不同将这些工作分为概率化方法和非概率化方法，分别在下文中加以介绍。图2.1总结了一些常见的话题建模方法的发展脉络图，这些方法都可以看成是LSA的衍生与改进。

2.1.2 概率化方法

Hofmann在1999年从概率论的角度重新定义了LSA，发明了概率化潜在语义分析模型,即PLSA（Probabilistic Latent Semantic Analysis）[71, 72]。PLSA用产生式的方式建模了文档中词-文档共现关系。PLSA把词和文档都看成是随机变量，同时还引入了一个潜在的随机变量z来表示文档中潜在的语义元素，称为一个话题。它假设文档集合中有K（通常远远小于文档和词汇数目）个话题，给定一个文档d，其中每个词w的产生：

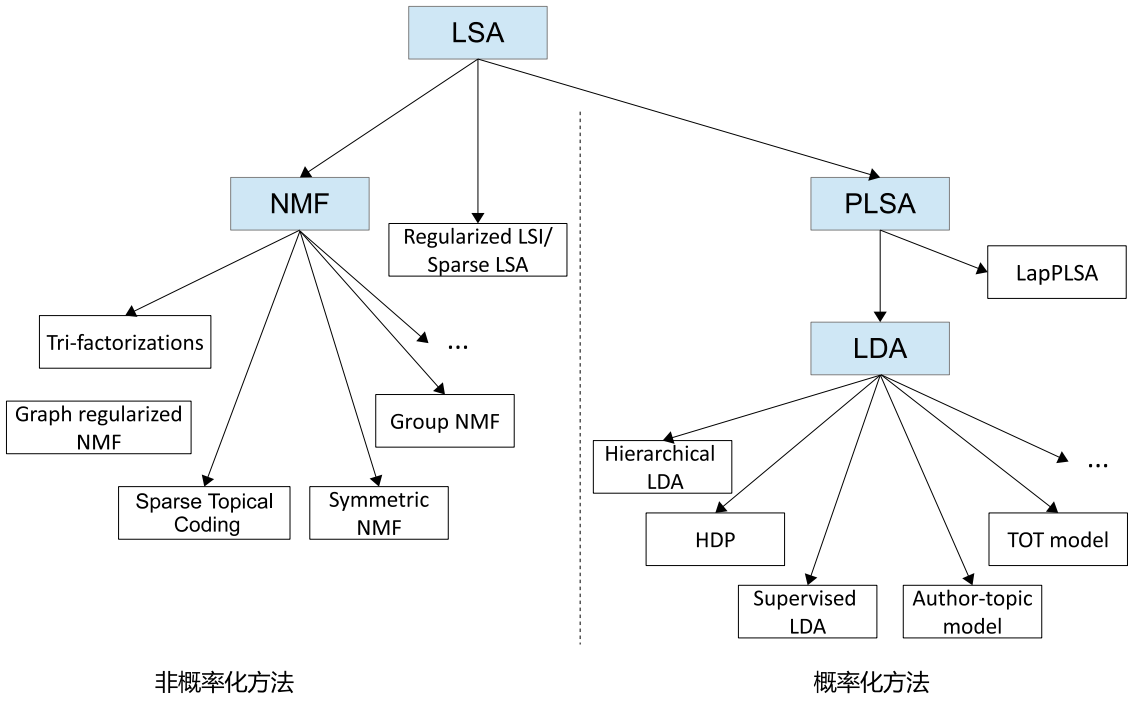


图 2.1: 一些常见话题建模方法发展脉络示意图

1. 从概率分布$P(z|d)$中抽取一个话题z

2. 然后从概率分布$P(w|z)$中抽取一个词w

这里$P(z|d)$和$P(w|z)$都定义成多项式分布。然后我们就可以通过最大似然的方式用期望最大化(Expectation - Maximization或EM) 算法[52]求解出模型参数$\{P(z|d), P(w|z)\}$。

PLSA用统计中常用的一种方法——混合分解（mixture decomposition）替代了LSA中的SVD分解。相比LSA，PLSA具备更坚实的统计学基础，其建模过程更有明确的概率解释，更容易让人理解。更重要的是，PLSA的效果通常也更好。另外，EM求解算法的时间复杂度是O(KWD)，通常要比SVD小的多。因此，PLSA 在工业界应用较多。

然而，PLSA也存在一些问题：

- PLSA并不是一个定义良好的产生式模型，其只定义文档中词的产生，而没有定义文档的产生。因此，PLSA缺乏对新文档生成过程的描述；
- PLSA中的参数过多,容易过拟合。

针对这些问题，Blei等人在2003年提出了LDA(Latent Dirichlet Allocation)[22]。LDA在PLSA的基础上引入了两个Dirichlet先验分布来建模$P(z|d)$和$P(w|z)$的产生。先验的引入不仅使得LDA具备建模新文档产生的能力，同时也减轻了过拟合现象。

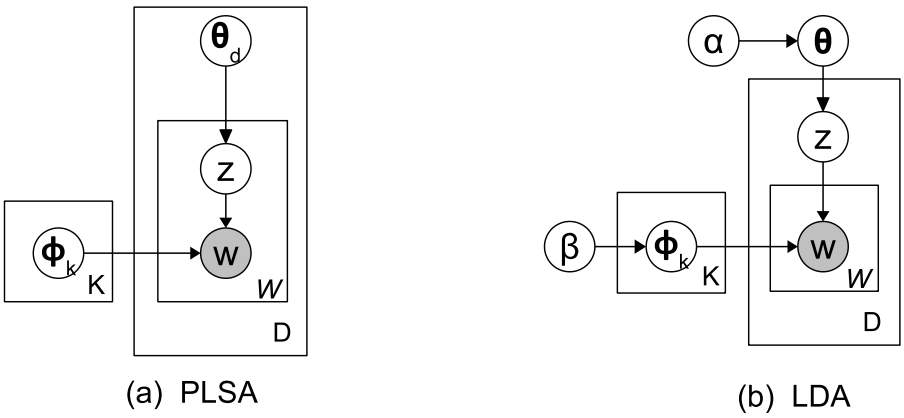


图 2.2: PLSA和LDA的概率图模型表示

在LDA中，文档被认为是话题的混合分布，用$\boldsymbol{\theta}_d$来表示；而话题是词的一个混合分布，用$\boldsymbol{\phi}_k$来表示。同时，$\boldsymbol{\theta}_d$和ϕ_k又是分别从一个Dirichlet分布中产生。详细的文档的产生过程如下：

1. 对于每一个话题$k \in [1, K]$，采样一个词分布: $\boldsymbol{\phi}_k \sim \text{Dir}(\beta)$
2. 对于文档集合中每个文档$d \in [1, D]$
   (a) 采样一个话题分布：$\boldsymbol{\theta}_d \sim \text{Dir}(\alpha)$
   (b) 对于文档中的每个词w
      i. 从$\boldsymbol{\theta}_d$中采样一个话题z
      ii. 从话题z中采样$w \sim \text{Mult}(\boldsymbol{\phi}_z)$

图2.3给出了一个小示例，详细说明参见图下方的文字说明。

从概率图模型上看，LDA（图2.2(b)）和PLSA（图2.2(b)）的差别就是在于引入了两个Dirichlet先验，参数分别为α和β。这里之所以选择Dirichlet先验，因为它是和多项式分布共轭，可以极大地方便参数的估计（参见下一小节）。引入这两个先验有两个重要作用：1）可以缓解PLSA的过拟合问题；2）另外，对于一个新文档，我们也可以估计其产生的概率。这是PLSA不具备的。特殊地，如果α和β都等于0，此时LDA等价于PLSA，因此PLSA可以看成是LDA的一种特殊情况。

PLSA和LDA是两个最基本的概率化话题模型，后续有很多基于这两个工作的扩展。如LapPLSA[29]通过引入基于文档之间的相似度构造正则化因子来约束话题的学习，以保证相似度高的文档其学习到的话题分布一致。考虑到LDA的扩展则非常多，这里我们只简单介绍几种有代表性的工作。Blei 等人随后提出了层次化LDA模型[19],该模型能自动学习话题之间的层次结构。PLSA和LDA都需要事先指定话题的数目，Teh等人提出的HDP模型能用非参方法去自动学习话题的数目[138]。Blei等人结合LDA和

有监督学习，提出了有监督的LDA模型[21]。另外，还有很多研究工作将其他元信息引入到LDA中来学习话题和这些元信息的关系。如author-topic模型[122]同时建模了作者信息，TOT模型建模了时间信息[149]等等。

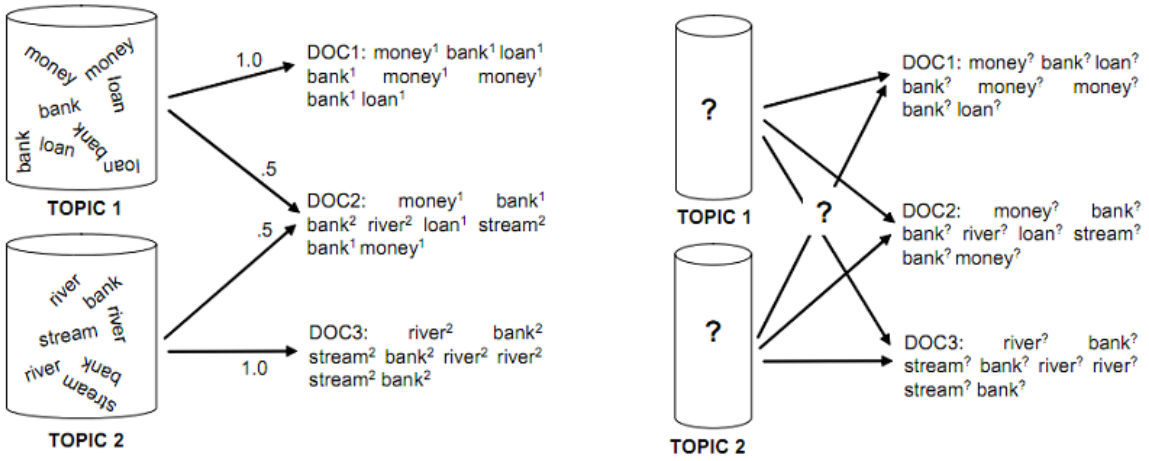
2.1.3　非概率化方法

非概率化方法主要是借助线性代数工具来对话题建模。其中大部分工作用非负矩阵分解（NMF）[87]替代原来的SVD分解。NMF将原词-文档矩阵分解成两个非负低秩子矩阵，分别表示词-话题与话题-文档之间的关系。在使用KL-divergence作为损失函数时，NMF等价于PLSA[58]，所以NMF可以取得和PLSA类似的效果。Xu等人最早将NMF引入到文本处理[157]，发现NMF用来对文档聚类效果不错。基于基本的NMF，后续有很多扩展。如Li等人[90]总结了多种NMF的变形，并提出的Tri-factorization将词-文档矩阵分解成三个子矩阵, 来对文档聚类。GraphNMF[28]引入文档之间的相似度来约束分解，思想和LapPlSA一样。Zhu等人[173]引入稀疏性约束来控制参数的稀疏程度。Yan等人[161]将对称NMF应用到短文本话题建模。Wang等人[147]提出的Group NMF对不同类别中话题协同学习。另外，也有些工作在原LSA分解的基础上引入约束来改善话题建模的效果，如Regularized LSI[148]和sparse LSA[43]。

2.1.4　总结

概率化方法和非概率化方法各有优势。概率化方法的统计学基础更坚实，模型更模块化，容易扩展。而非概率化方法由于没有概率约束，更为灵活，比如可以随意定义词的权重，加入稀疏性约束等。但相比概率化方法，非概率化方法的原则性要差些，效果调优更麻烦些。因此，本文中主要采用概率化方法，在后续篇章中不再涉及非概率方法。但正如NMF与PLSA之间存在等价关系一样，很多概率化的方法可以通过放松概率约束条件，转化为非概率化方法；反之亦然。所以，我们认为概率化方法和非概率化方法之间其实并没有实质性的差别。

除了以上两大类话题建模方法之外，还有一类值得关注的方法，即基于神经网络的话题模型[85, 97, 126]。这类方法用无向图的方式去建模文档、词与话题三者之间的关系，表达方式更灵活，但通常训练代价也更高。随着近年来深度学习方法成为机器学习领域中一个炙手可热的方向，并相继各个应用领域取得突破。目前已经有很多研究者尝试将深度学习的方法引入文本语义分析与处理领域，并取得了一些进展[128, 129]。因此，基于深度神经网络的话题模型有可能成为下一个研究热点，给文本语义分析带来新的突破。



(a) LDA 产生式建模　　(b) LDA 统计推断

图 2.3: (a)LDA的文档产生建模示意图。DOC1中的词都从话题1中抽取得到，DOC1中的词有0.5的概率从话题1，另外0.5 的概率从话题2中抽取得到，DOC3中的词都从话题2中抽取得到。（b）LDA的统计推断过程基于观察到的文档集合，来推断模型结构，包括每个文档的话题分布，话题中词的分布，以及每个词的话题来源。图片来自于[135]

2.2 统计推断方法

概率化话题模型都是利用统计推断方法去学习模型的参数。统计推断可以看成是产生式建模的逆过程，即基于现有观察到的文档集合去推断模型结构，如图2.3中示例所示。话题模型中常用的统计推断算法有Gibbs采样[61]、变分推断[22]、EP(Expectation Propagation) 算法[106]和最大化后验估计[46]等。其中比较最为流行的是Gibbs采样和变分推断方法。其实，这两种算法以及最大化后验估计方法之间的主要差别在于先验的平滑程度[8]。根据[8]的结论，Gibbs采样通常比其他两种算法的求解精度更高，消耗内存也更少。相对于变分推断而言，Gibbs采样的求解过程也更简单。所以在本文当中，我们主要介绍Gibbs采样推断方法，其他方法可以参考相关文献。

Gibbs采样算法[59]是蒙特卡洛模拟（Markov chain Monte Carlo或MCMC）方法的一种特例，比较适合于高维隐变量模型的参数估计。其基本思想是按Markov链的方式交替地去对待估计的随机变量进行后验采样，其中每次采样基于其他随机变量的赋值。下面我们以LDA为例，介绍用Gibbs采样算法来做统计推断的过程。在LDA中，我们需要估计的随机变量包括$\boldsymbol{\phi}_k$、$\boldsymbol{\theta}_d$和每个词的话题赋值z_i，但利用collapsed gibbs采样[61]，$\boldsymbol{\phi}_k$、$\boldsymbol{\theta}_d$在具体求解过程中可以被积掉，并不需要显式的求出。因此，我们只需要对z_i进行迭代采样。

我们假设整个语料包含D个文档，将其表示一个N个词的序列$\mathbf{w} = \{w_1, w_2, ..., w_N\}$，其中$w_i$所属的文档记为$d_i$。Gibbs 采样需要对每个词的话题赋值$z_i$进行采样。其中，采

样的关键是计算隐变量的条件概率分布$P(z_i|\mathbf{z}_{\neg i}, \mathbf{w})$, 这里$\neg i$ 表示不包含第i个词。下面我们来推导$P(z_i|\mathbf{z}_{\neg d,i}, \mathbf{w})$的计算。根据贝叶斯公式：

$$P(z_i|\mathbf{z}_{\neg i}, \mathbf{w}) = \frac{P(\mathbf{z}, \mathbf{w})}{P(\mathbf{z}_{\neg i}, \mathbf{w})} \propto \frac{P(\mathbf{w}|\mathbf{z})P(\mathbf{z})}{P(\mathbf{w}_{\neg i}|\mathbf{z}_{\neg i})P(\mathbf{z}_{\neg i})}. \tag{2.1}$$

其中:

$$\begin{aligned} P(\mathbf{w}|\mathbf{z}) &= \int P(\mathbf{w}|\mathbf{z}, \mathbf{\Phi})P(\mathbf{\Phi})\mathrm{d}\mathbf{\Phi} \\ &= \int \left(\prod_{n=1}^{N} P(w_n|z_n, \boldsymbol{\phi}_{z_n})\right) P(\mathbf{\Phi})\mathrm{d}\mathbf{\Phi} \\ &= \int \prod_{k=1}^{K} \left(\frac{\Gamma(\sum_{w=1}^{W}\beta_w)}{\prod_{w=1}^{W}\Gamma(\beta_w)} \prod_{w=1}^{W} \boldsymbol{\phi}_{k,w}^{n_{w|k}+\beta-1} \mathrm{d}\boldsymbol{\phi}_{\boldsymbol{k}}\right) \\ &= \left(\frac{\Gamma(\sum_{w=1}^{W}\beta_w)}{\prod_{w=1}^{W}\Gamma(\beta_w)}\right)^{K} \prod_{k=1}^{K} \frac{\prod_{w=1}^{W}\Gamma(n_{w|k}+\beta_w)}{\Gamma(n_{\cdot|k}+\sum_{w=1}^{W}\beta_w)}, \end{aligned} \tag{2.2}$$

这里$\Gamma(\cdot)$是标准的Gamma函数，$n_{w|k}$表示词w从话题k中抽取出来的次数, $n_{\cdot|k} = \sum_{i=1}^{W} n_{w_i|k}$。$P(\mathbf{z})$的计算如下:

$$\begin{aligned} P(\mathbf{z}) &= \int P(\mathbf{z}|\mathbf{\Theta})P(\mathbf{\Theta})\mathrm{d}\mathbf{\Theta} \\ &= \int \left(\prod_{n=1}^{N} P(z_n|\mathbf{\Theta})\right) P(\mathbf{\Theta})\mathrm{d}\mathbf{\Theta} \\ &= \int \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_{k|d}^{n_{k|d}+\alpha-1} \mathrm{d}\mathbf{\Theta} \\ &= \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)} \frac{\prod_{k=1}^{K}\Gamma(n_k+\alpha_k)}{\Gamma(D+\sum_{k=1}^{K}\alpha_k)}. \end{aligned} \tag{2.3}$$

同样的，我们可以计算出$P(\mathbf{w}_{\neg i}|\mathbf{z}_{\neg i})$ 和$P(\mathbf{z}_{\neg i})$:

$$P(\mathbf{w}_{\neg i}|\mathbf{z}_{\neg i}) = \left(\frac{\Gamma(\sum_{w=1}^{W}\beta_w)}{\Gamma(\beta)^W}\right)^{K} \prod_{k=1}^{K} \frac{\prod_{w=1}^{W}\Gamma(n_{\neg i,w|k}+\beta_w)}{\Gamma(n_{\neg i,\cdot|k}+\sum_{w=1}^{W}\beta_w)}, \tag{2.4}$$

$$P(\mathbf{z}_{\neg i}) = \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)} \frac{\prod_{k=1}^{K}\Gamma(n_{\neg i,k}+\alpha_k)}{\Gamma(D+\sum_{k=1}^{K}\alpha)}, \tag{2.5}$$

将式（2.2-2.5）代入式（2.1）, 并利用Gamma函数的性质$\Gamma(x+1) = x\Gamma(x)$，消去公共项后可以得到Gibbs采样所需的条件分布：

$$P(z_i = k|\mathbf{z}_{\neg i}, \mathbf{w}) \propto (n_{\neg i,k}+\alpha_k)\frac{n_{\neg i,w_i|k}+\beta_{w_i}}{\sum_{w=1}^{W} n_{\neg i,\cdot|k}+\beta_w}. \tag{2.6}$$

Algorithm 1: LDA的Gibbs采样算法

Input: topic number K, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, word vector $\mathbf{w}$
Output: $\{\boldsymbol{\phi}_k\}_k$, $\{\boldsymbol{\theta}_d\}_d$
Randomly initialize the topic assignments for all the words
for *iter* $= 1$ *to* N_{iter} **do**
  foreach *word* $w_i \in \mathbf{w}$ **do**
    Draw topic k from $P(z_i|\mathbf{z}_{-i}, \mathbf{w})$
    Update n_k and $n_{w_i|k}$
Compute $\boldsymbol{\phi}_k$ by Eq. (2.7) and $\boldsymbol{\theta}_d$ by Eq. (2.8)

现在我们给出用Gibbs采样算法来估计LDA中参数的步骤。首先，我们对所有词的话题赋值进行随机初始化[1]；然后对每个词，我们计算其话题条件概率分布，即式（2.6），并据此采样一个话题。依次迭代，直到收敛。我们通过收集采样得到的样本，就可以估计出模型的参数：

$$\phi_{k,w} = \frac{n_{w|k} + \beta_w}{n_{\cdot|k} + \sum_{w=1}^{W} \beta_w}, \tag{2.7}$$

$$\theta_{d,k} = \frac{n_k + \alpha}{D + \sum_{k=1}^{K} \alpha_k}. \tag{2.8}$$

完整的算法如算法1所示。

2.3 结果评价

如何评价一个话题模型的好坏一直是一个开放性问题，目前仍没有一个统一的标准。主要因为：1）话题模型是一种无监督方法，数据中本身并没有话题标注信息；2）话题模型通常都是作为一种中间件，而不像文本分类、信息检索等应用有一个明确的目标。因此，话题的好坏评价标准对于不同的应用而言也可能不一样。比如，对于文本分类而言，要求话题之间的区分度较高；而对于信息检索而言，可能要求话题结果更全面一些，以保证检索结果的召回率。

以下，我们介绍目前话题模型相关研究中比较常见的一些评价方式，并讨论各自的优点与不足。

2.3.1 基于测试集拟合度的评价

自从PLSA和LDA提出以来，一种典型的评价方式就是在训练模型之前先保留一部分数据，然后用训练后的模型去拟合保留数据。常用的指标有混乱度（Perplexity）和似然函数等。由于这些指标大同小异，我们只介绍最常见的一种——混乱度。混乱度是

[1]这里也可以采用KNN等聚类方式来初始化，在一定程度上可提升效果和加快收敛速度

统计语言模型中常用的一个指标[10]，常被用来评估模型的泛化能力。令$\mathbb{D}'$表示保留的测试集文档集合，混乱度的定义如下：

$$\text{perplexity}(\mathbb{D}'|\mathbf{\Phi},\mathbf{\Theta}) = \exp\{\frac{\sum_{d\in\mathbb{D}'}\log(P(\mathbf{w}_d|\mathbf{\Phi},\mathbf{\Theta})}{\sum_{d\in\mathbb{D}'}N_d}\}$$

其中

$$\log P(\mathbf{w}_d|\mathbf{\Phi},\mathbf{\Theta}) = \sum_{i=1}^{N_d} n_{d,w_i}\log(\sum_{k=1}^{K}\phi_{k,w_i}\theta_{d,k}),$$

其中，n_{d,w_i}表示词w_i在文档d内出现的次数。混乱越低，说明学习到的参数对测试集拟合程度越高，认为该模型效果更好。

基于测试集拟合度的评价方式的好处是不依赖于外部数据和人工标注信息，计算简单。然而，这种评价方式是有争议的。Chang等人[37]发现对测试集拟合度好的模型学习到的话题反而可能与人对话题质量的评价相左。最近David Blei也承认这种评价方式是和人们对话题模型的预期结果是脱节的[18]。

2.3.2　间接评价

另外一种常见的评价方式就是将话题模型看成是降维方式，即将文档从词空间降低到话题空间。然后把降维后的特征作为特定应用的输入，根据应用的结果来判断话题建模结果的好坏。常见的有以下几种：

- 文档分类，相关工作有[22, 173]等。这种评价方式主要是将$\boldsymbol{\theta}_d$作为文档d的表示,然后用常用分类器，如朴素贝叶斯、SVM等对其分类。分类效果越好，说明降维后文档的区分信息保持的越好，从这个角度上讲话题建模的效果也越好。这种评价方式适合文档有类别标签的数据。

- 文档聚类，相关工作有[29, 30]等。这种评价方式认为同属于一个类的文档，降维后的相似度应该较大；相反，原先不属于同一个类的两个文档，降维后相似度应该较小。具体有两种做法：1）把每个话题看做是一个类，然后把每个文档d分配到它最可能采样的那个类,即$P(z|d)$ 最大的那个话题; 2）用$\boldsymbol{\theta}_d$作为文档d的表示，然后用KNN等常用聚类算法对其聚类。同样的，这种方式适合文档有类别标签的数据。

- 文档检索，相关工作有[72, 147]等。基本思想是将查询和文档映射到语义空间，然后再算相似度。检索结果越好，说明话题学习结果对文档的表示越合理。

间接评价的优势就是与应用挂钩，能直接体现话题模型的实用价值。但由于话题模型本身并非针对某个应用的，这种评价方式相对比较片面。

2.3.3 话题一致性评价

近年来，很多研究者认为话题模型的出发点就是要学习到有意义的话题。因此，我们应该直接评价话题的语义一致性：话题中概率较大的词之间的语义相关性越强，该话题的可读性越好。

一种最简单的方式就是把每个话题中概率最大的几个词列出来，然后由人去对其相关性进行标注[169]。但这种方式主观性太强，很少使用。Chang等人[37]提出了一种词入侵和话题入侵的方式来分别对话题和文档的话题分布质量进行评价。词入侵是在每个话题中的概率较大的几个词中间随机插入一个其他话题中概率较大的；话题入侵是在每个文档对应的话题分布中取前3个概率最大的话题，然后随机插入一个概率较小的话题。然后让标注者去找出侵入词或话题。如果侵入词或话题很容易被区分，说明结果较好。

人工标注成本高，而且易受标注者的主观性影响，很多研究者仍在积极探索自动评价方法。Newman等人[111]在2010年提出用外部数据（如WordNet，Wikipedia, google 搜索结果等）来自动化地评价话题一致性。其主要思想是从外部数据中来计算话题中概率最大的那几个词之间的相关性。其实验结果表明，利用大规模知识库或者网页数据，可以取得和人工评价差不多的效果。2011年，Mimno等人[105]提出了另外一种自动化话题一致性评价方法——coherence score。Coherence score的特点是无需借助外部数据，只利用文档集合内部的词共现关系去评价话题中概率最大的前几个词之间的相关性。一个话题中排在前面的词在文档集合中共现的文档频率越高越好。其实验结果表明coherence score评价结果与人工标注有较强的相关性。

相比前面两种方式，话题一致性评价更接近于话题建模方法的初衷，即学习到有意义的话题，是值得提倡的一种评价方式。

2.4 相关应用

话题模型作为一种基本的自动语义分析工具，被广泛应用于文本挖掘以及相关领域当中。由于篇幅所限，我们在这里列举其中的一部分工作。

2.4.1 文档自动组织与管理

如今海量的文本数据，如网页、书籍、科技文献等，给人们带来了丰富信息的同时，也增加了人们对这些信息组织管理和探索的难度。以往的文档管理方式，如情报科学、雅虎早期的目录搜索等，依赖人去定义一个目录结构，然后对文档归类。这种方式一是成本非常高；二是目录结构的制定比较困难。因为对于不同领域的文档，其目录结构也不一样，如果没有全面的领域知识，很难制定一个合理的目录结构。因此，这种人工分类的方式显然不适合如今的海量文本数据。

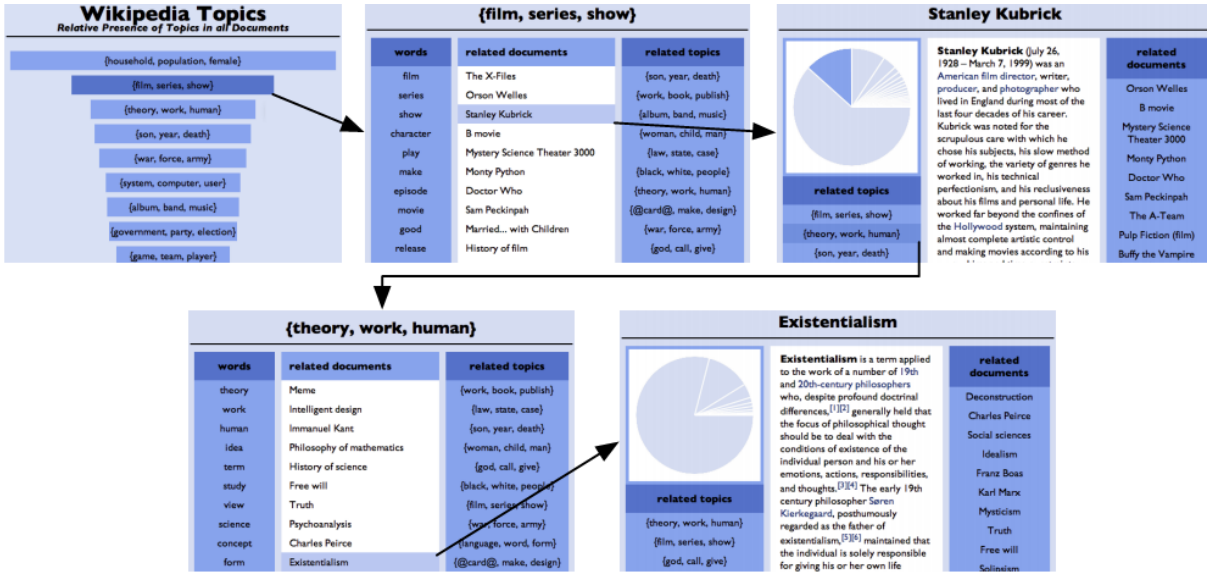


图 2.4: 用话题模型来自动组织Wikipedia内容

话题模型为文档组织和管理提供了一种全自动的解决方案。它不需要人工的去指定目录结构，而是根据数据的分布自动抽取其中的话题，每个话题可看作是一个分类。此外，它还可以学习到每篇文档的话题属性，把文档按话题来组织起来。图2.4展示了用LDA[22]来对Wikipedia语料进行自动组织的一个示意图，图片来源于[36][2]。其中，左上角显示Wikipedia语料中包含的话题，每个话题由其中概率最大的三个词来表示，这些话题概括了整个语料中的主要内容。点击一个其中关于电影和电视的话题，进入到第二个页面，包含三栏内容。第一栏显示的是话题中的概率最大的有些词，可以看出基本上都和电影、电视非常相关，这些词描述了话题的内容；第二栏是和该话题相关的一些文档，可以看出基本上都是一些电影和电视的名称，说明我们可以根据话题来定位文档；第三栏显示的是一些相关的话题，相关性通过计算话题的词分布的相似度得到。点击第二栏中的一个文档，便可以进入到该文档的详细页面。第一栏显示的是文档相关的一些话题及其在文档中的比例，概括了该文档的主要内容；第二栏是该文档的内容；第三栏是根据文档的话题分布计算得到的相关文档集合，方面我们探索相关的内容。点击第一栏中的某个话题，又可以浏览另一个话题。需要注意地是，图2.4中所有的内容和链接结构都是根据话题模型自动计算得到，不需要人工干预。

应用一些更高级的话题模型，还可以挖掘语料中更复杂、更丰富的结构，从而提供更灵活的组织方式。如Blei等人提出的层次话题模型[19]可以自动生成层次目录结构。Steyvers等人[136]提出的author-topic模型把作者建模成一个话题分布，来学习作者对应的话题。Blei和Lafferty提出相关话题模型（Correlated Topic Models）[17]可以自动学习话题之间的相关关系。Li等人提出的Pachinko Allocation模型[91]可以学习

[2]Demo地址: http://www.princeton.edu/ achaney/tmve/wiki100k/browse/topic-presence.html

到话题之间的有向图连接关系。Blei和Lafferty提出的动态话题模型（Dynamic Topic Models）[20]用状态空间模型来建模话题分布随时间的演化与趋势。Wang等人[149]则把时间作为文档的一个特征用一个连续型随机变量建模，从而学习到话题与时间的关系。Wang等人提出了马尔科夫话题模型（Markov Topic Models）[144]来学习多个语料中的话题结构与关系，可用来对多个语料的组织与管理。

2.4.2 信息检索

1) 语义检索

在信息检索中，通过话题模型可以对查询和文档进行语义分析，实现二者之间的语义匹配，从而能帮助人们更准确的找到想要的信息。早期的LSA[50]和PLSI[72]都是从对文档进行语义索引出发，将文档以及查询映射到一个潜在的语义空间，从而实现语义上的匹配。随着话题建模技术的发展，人们也开始采用更高级的话题模型来改进语义检索性能。Wei等人[153] 提出了一种基于LDA[22]的文档建模方式和传统语言模型进行结合，并取得了和伪相关反馈接近的效果。但和伪相关反馈相比，基于话题模型的方法优势是可以离线计算。Wang等人[150]提出了一种topical n-gram模型来识别文档中和话题相关的n-gram来对文档建模，在文档检索中取得了比[153]更好的效果。然而话题是多个文档所体现出来的共同属性，如果仅仅用话题来表示文档可能会损失文档本身内部的一些特殊性信息，从而降低检索的精度[152]。为此，Chemudugunta等人[39]提出了一种特殊词话题模型，它将文档内部的词分为三类：仅和该文档相关的特殊词，和话题相关的词和背景词，分别对应一个概率分布。由于该模型同时考虑了文档内部的特殊信息与话题信息，该模型的检索结果显著优于TFIDF和基于LDA的检索方法[22]。

2) 查询理解

通常搜索引擎中的查询都很短，而且很多都是二义性的，给搜索引擎判断用户的信息需求带来很大的困难。话题模型可以挖掘查询中潜在的语义信息，从而更准确地理解查询。Guo等人[65]提出一种弱监督的LDA方法，借用一些已标注的种子命名实体及其类别信息来识别查询中的命名实体及其类别。Bing等人[15]将点击同一个网站的查询聚合起来用LDA学习话题，然后基于历史查询中大量语义相似的查询去精化当前查询。Guo等人[64]利用正则化话题模型从查询的搜索结果片段和查询之间的共同点击信息学习查询背后的意图。Zhang等人[168]利用查询点击的URL对应的网页内容作为文档，通过学习查询的话题属性来计算查询的话题相似度并用来对查询会话进行切分。

3）个性化搜索

话题模型还可以用来对用户的搜索行为来学习到用户对话题的偏好程度，并以此提供个性化搜索服务。Carman等人[31]通过建模用户及其点击的URL对应的文档之间的话题相关关系，推断出用户的话题属性，并结合语言模型框架来做个性化搜索。

Song等人[130]从用户的历史查询结果中选择用户点击过的或者排名高的网页内容来训练PLSA，然后在查询模型中融合用户的话题属性，以生成个性化的搜索结果。类似地，Jayarathna等人[77]利用LDA来对用户标注的文档进行话题学习，然后根据用户的话题属性生成个性化结果。

4) 结果多样性排序

对于同一个查询，不同的用户以及不同的上下文背景，用户的信息需求可能不一样。多样性的排序结果能提供给用户更多的信息量，从而满足用户不同的信息需求。一种简单的假设每个查询有多个子话题，每个子话题对应于一种不同的信息需求[165]。因此，话题模型为自动分析查询和文档中潜在的子话题的提供了一种工具。2009年，Carterette等人[32]提出了分面话题检索（faceted topic retrieval）的任务，认为一个查询可能包含多个分面，一个检索系统返回的一小部分文档集合应尽可能覆盖所有的分面，以保证结果的多样性。基于这样一个目的，其提出了一种基于LDA的排序方法，把在伪相关反馈的文档集合中学到的话题当作分面，然后用贪心算法每次选择使当前期望分面最大的文档加入结果文档集合。类似地，Welch等人[154]基于查询返回的前200个文档来训练LDA，将学到的话题作为子话题应用到一种不同的排序模型当中。

2.4.3　情感分析

情感分析(sentiment analysis)又称文本倾向性分析、观点挖掘，其主要目的在于识别文本语料中用户对事物或人的看法、态度。随着Web 2.0的发展，人们越来越多地利用互联网工具（如博客、论坛、微博等等）来发表自己的意见与观点。对互联网上海量的用户数据进行情感分析可以帮助我们快速掌握用户对商品、人物、事件等事物的褒贬态度与意见，从而帮助相用户优化自己的决策。

要准确的分析出文本中的情感要素要求计算机深刻地理解文本内容，对语义分析技术要求较高。以往的研究或多或少的依赖于人工语义分析，比如用户挑选一些情感词作为种子词[47]，或者去标注一部分文本的情感类别来训练分类器[80, 114]等。近年来，话题模型的发展促进了自动语义分析水平的提高，目前已被成功地应用到情感分析相关的任务中，以降低情感分析中的人工干预成本。

1）方面发现（Aspect Discovery）

方面是评论中所涉及到的事物的某个属性，比如宾馆的服务、设施、饮食等。早期的方面发现多采用信息抽取手段，如抽取频繁出现的名词短语[74]或者基于专家制定的抽取规则[118]，但这种方式得到结果不全面，鲁棒性差。近年来，话题模型被广泛应用到方面的自动学习当中。这类方法将方面看成是一类特殊地话题，然后通过挖掘评论中的词共现关系自动学习方面。2008年, Titov[141] 最早探索用话题模型来学习评论中的方面。他们发现原始LDA学习到的话题是全局性的，并不能很好的对应于评论中的方面。于是，他们提出了一种多粒度话题模型(MG-LDA), 将评论中的话题分为

全局话题和局部话题两种。每条评论包含一个全局话题分布，而评论中每个滑动窗口内的相邻的句子包含一个局部话题分布。评论中的每个词由一个指示变量来决定是来源于全局话题还是局部话题。实验结果表明，该方法学习到的局部话题与评论中的方面有较好的对应关系。Titov的方法比较繁琐，Brody等人[27]采用了一种更简单地局部LDA方式去学习方面，即把每个句子看成是一个文档去训练LDA。标准LDA可看成是文档聚类，而局部LDA的思想是对句子聚类。因为通常一条评论中可能涉及很多个方面，如食物、服务等，对评论聚类容易导致各种方面混合在一块。但一个句子通常只包含一个方面，因此对句子聚类的方式更容易得到可读性好的方面。Jo等人[81]提出Sentence-LDA来学习方面，即在LDA的基础上约束每个句子中的词来自同一个话题，该方式也是沿用了对句子聚类的思想。从以上几个工作中的论文结果来看，它们都可以比标准LDA方法更有效的学习到方面，但这几种方法之间到底孰优孰劣，缺乏定量的分析比较。

2）话题情感联合分析

2007年，Mei等人[102]首先提出话题情感联合分析问题，即同时发现文档中的话题与情感，以及二者之间的关系。他们并提出TSM （Topic-Sentiment Mixture) 模型来分析博客中的话题与情感关系。TSM在PLSA的基础上将话题分为三类：K个普通话题，1个褒义情感话题和1个贬义情感话题，另外还引入了一个背景词语言模型来去除高频词。其中文档中每个词产生时，先从K个普通话题中选择一个话题j，然后关于j 选择一个情感标签，如果是中性的，则从普通话题中产生；否则该词从对应情感话题中产生。为了能引导普通话题和情感话题的学习，TSM借助第三方情感分析工具Opinmind获得的数据来构造话题词分布的Dirichlet先验。另外，TSM需要经过后处理的方式来判断文档的情感类别。为了克服这些问题，Lin等人[92]提出了一个基于LDA的扩展模型——JST（Joint Sentiment/Topic Mode)。JST文档和话题之间引入了一个情感层，把文档建模成一个情感类别的分布，而一个情感类别对应于一个话题的分布。为了引导不同情感类别的话题学习，Lin等人首先构造一个情感词分类字典用来指导JST中初始化过程。Jo等人[81]提出的ASUM模型在sentence-LDA的中将情感-方面搭配作为一个话题，并利用一些种子情感词给两种不同话题的词分布设置非对称先验来区分褒义情感与贬义情感所对应的方面。

3）话题（或方面）相关的情感词抽取

情感词（sentiment words）又称意见词或观点词（opinion words），即带有情感色彩的词。学习到话题对应的情感词可以方便对话题进行情感分类，在评论情感分析很有必要。起初，Brody等人[27]采取两部走的策略。首先用局部LDA去学习评论中的方面，然后抽取评论中的形容词及其并列关系构建一个图，再利用一些已知极性的种子词进行极性传播来获得方面的情感极性。另一种做法是将情感词也建模成一个特殊的话题，然后用一个统一的话题模型进行学习。如Zhao等人提出的MaxEnt-

LDA[170]和Mukherjee等人提出的SAS 模型[107]。MaxEnt-LDA的核心思想是将LDA中每个话题分为方面词话题与情感词话题两类，并训练一个最大熵分类器来判断一个词应该属于哪种类别。SAS在MaxEnt-LDA的基础上通过约束一些种子方面词在话题中的概率尽可能一致来增强方面的可读性。

2.4.4　自动文摘

自动文摘，即从文本语料中自动抽取有代表性的内容或句子作为摘要。在互联网上信息过载的今天，自动文摘可以方便用户快速浏览信息，节省时间和精力。自动文摘考验的是计算机对文本内容的理解和概括能力。早期的自动文摘方法都依赖于较强的先验知识和人为设定的启发式技术，实现起来比较费时费力，而且通用性较差。话题模型具备的自动语义分析能力为自动文摘提供了一个新的工具来更好的理解和概括文本内容。

1) 代表句子抽取

从文档中抽取代表性句子的一种常用的方法就是选择和文档最相似度的句子作为该文档的摘要[67, 142]。但由于句子非常短，基于词空间的相似度计算容易受同义词和多义词的影响。为了克服这个问题，Arora等人[7]先用LDA学习文档中的话题，然后计算句子由文档通过话题来产生的概率，选择生成概率大的句子作为摘要。类似地，Chang和Chien[38]提出了一种基于句子的LDA扩展模型，把文档中每个句子建模成一个话题的混合分布，然后计算文档和句子之间的话题相似度。Wang等人[145]假设每个句子来源于一个话题，然后选择这些话题下概率最大的句子作为摘要。Haghighi等人[66]基于层次LDA[19]分别对背景话题和不同粒度的话题建模，包括文档集合级别话题、文档级别话题、句子级别话题，然后基于句子和文档级别话题中词的分布计算句子和文档的相似度。该方法在实验中取得了比以往判别式方法更好的效果。Celikyilmaz 等人[34]同样采用层次LDA 模型，但在[66]的基础上引入了句子的一些元特征，如N-gram、词的文档频率等特征用回归模型来预测句子的得分。

2）面向查询的摘要

面向查询的摘要是在给定一个查询和其相关文档集合的情况，从其相关文档中抽取和该查询最相关的句子。针对这一任务，Daume等人[48]在2006年提出了一个贝叶斯查询摘要模型，该模型在LDA的基础上定义了三类话题：背景话题（主要是常见词）、文档相关话题和查询相关话题。通过统计推断，最终产生查询相关话题的概率大的句子被选择作为查询对应的摘要。Celikyilmaz[35]等人基于有向图话题模型Pachinko allocation[91]提出一种两层话题模型，相比之前的工作，其特点在于用上层话题用来描述下层话题的相关性。

3）更新摘要

更新摘要是自动文摘中比较新的任务，其假设在用户之间已经读过一些相关文档

的基础上，对目前的文档集合做摘要。和以往的摘要不同，更新摘要还需要考虑生成的摘要的新颖性。2012年，Delort等人[51]开始采用话题模型来产生更新摘要。为了能学习到当前文档集合中较新的话题，其建模了四种不同的话题类型：和以往相关文档集合相关的话题、和当前文档集合相关的话题、公共话题、和背景话题。然后选择和当前文档集合相关的话题中概率最大的句子作为更新摘要。Li等人[146]用两个三层HDP模型[138]分别学习历史话题和新话题，同时考虑了句子与句子之间的话题相关性。

2.4.5 总结

话题模型在过去十年间一直是机器学习和数据挖掘领域中的一个热门研究话题。本章回顾了话题模型发展的历史，并着重概率话题模型及其统计推断方法，为后续章节提供相应的背景知识。同时，我们总结了目前常用的几种话题模型评价方式，指出了各自的优点与不足。最后，我们例举了话题模型在文本挖掘的各个领域中的应用。通过以上综述，我们可以看到话题模型为文本的语义分析提供了一种自动化的工具，能有效提高计算机对大规模文本语料处理的水平，在实际应用当中具有广泛的应用价值和深远的意义。

然而，我们也发现以往的话题模型研究基本上都是围绕长文本进行的，因为过去的文本语料（如网页、新闻、科技文献等）多为长文本数据。但在如今的互联网上，短文本信息非常普遍。如何有效的对短文本进行话题建模，相关的研究非常不充分。本文后续章节围绕此问题展开了研究。

第三章　双词话题模型

3.1　引言

短文本在互联网上很常见，如网页与新闻标题、文本广告、图片说明等。特别是近年来随着微博等社交媒体的发展，短文本逐渐成为互联网上信息传播的一种主要形式。这些短文本消息通常只包含十几个甚至几个词，内容非常稀疏，给现有的话题建模方法带来了新的挑战。人们发现直接应用以往的话题模型（如PLSA[71] 和LDA[22]）对短文本进行话题建模，效果并不理想[73, 169]。

本章中，我们首先分析了短文本的内容稀疏性对以往话题建模方式的影响。然后针对短文本的内容稀疏问题，提出了一种新的话题模型，从而更有效的学习短文本数据中的话题。

3.2　概述

为了明确传统话题模型在短文本上的问题，我们首先简单回顾下传统的话题模型的建模方式。这些话题模型的一个基本假设就是文档是话题的混合分布$\boldsymbol{\theta}_d$，而话题是一个词的分布$\boldsymbol{\phi}_k$，文档中的每个词来源于从$\boldsymbol{\theta}_d$中采样的话题。然后，基于观察到的文档集合，我们可以通过统计推断工具去估计模型参数$\boldsymbol{\theta}_d$和$\boldsymbol{\phi}_k$，分别对应于一个文档中话题的比例和话题的表示。直观的讲，这些话题模型是通过挖掘文档级别的词共现关系来发现话题，短文本上面临严重的数据稀疏性问题：由于文本很短，单篇文档内部的词共现关系非常有限，从而影响对模型参数的准确估计。一方面，长文本文档中，相关词会反复共现，因此较容易通过词频来判断两个词在文档当中的相关程度。而由于短文本中的词的词频通常都为1，因此我们很难通过词频来判断哪两个词更相关些。另一方面，我们知道自然语言中很多词都是多义性的，要准确判断词的含义需要充分的上下文信息。而短文本中，文档内部的上下文信息非常少，这也给我们判断这些多义词的话题带来了很大的困难。

为了克服短文本上的数据稀疏性问题，目前主要有两种方式。第一种方式是简单地将短文本文档聚合起来，得到一些长度较长的虚拟文档，然后再套用传统的话题模型。比如，Weng等人[155]把每个用户发的微博聚合起来作为一个文档，然后再训练LDA。Hong等人[73]尝试了多种聚合方式，如按用户和按词来聚合微博，发现聚合后要比直接训练LDA的效果要好，其中按用户来聚合又比按词聚合要好。但是这种启发式的数据聚合方式的效果是和数据相关的。如果聚合的方式不合理，反而可能产生噪音从而影响结果。第二种方式通过在模型中引入一些较强的约束来克服对短文本文档建模所面临的数据稀疏性问题。比如，Zhao等人[169]和Lakkaraju等人[84]限定每条微

博中的词来自于同一个话题。类似地，Gruber等人[63]限定每个句子中的词来自于同一个话题。这种强制性约束以牺牲模型的灵活性为代价来降低模型的复杂度，同时也带来了两个新问题：1）无法刻画一个文档中包含多个话题的情况；2）导致文档中话题的后验估计会过于陡峭，容易过拟合[22]。总之，这两种方法并没有很好的解决短文本话题学习问题。

在本章中，我们提出了一种新的话题模型来克服短文本文档过短所带来的数据稀疏性问题。该方法的主要思想来源于对话题模型本质的思索。从根本上讲，话题模型学习到的话题其实是一组语义相关词的集合，其中词之间的相关性是由词在文档中的共现模式所决定。既然这样，为什么不直接通过建模词共现模式来学习话题呢？另外一方面，虽然对于单篇短文本文档来说，其中的词共现关系是稀疏的，但对于整个语料来说，（只要数据充分大）其中的词共现关系是丰富的。因此，为什么不直接利用丰富地全局词共现关系来学习话题呢？

基于以上动机，我们提出了一个新的产生式话题模型：双词话题模型（Biterm Topic Model或BTM）。它的特点是通过建模双词的产生来学习话题。这里，“双词”指的是一个在同一个上下文（即一个固定大小的滑动窗口限定的词序列）中共现的无序词对。我们假设一个双词中的两个词来源于同一个话题，而这个话题则是来源于整个语料上的一个话题分布。这里的话题的定义和传统话题模型一样，即一个词的分布。不同的是：1）BTM通过显式建模词共现模式（双词）来改进话题的学习；2）BTM通过将整个语料上的词共现模式聚合在一起去学习话题，从而克服传统话题模型利用文档级别词共现关系去学习话题所面临的数据稀疏性问题。另外，由于BTM没有建模文档的产生过程，我们不能直接从BTM中估计一个文档的话题比例$P(z|d)$。但幸运地是，我们可以基于BTM学习的话题结构通过后处理地方式推断出文档的话题比例。

本章其余内容组织如下：3.3节介绍了相关工作；3.4和3.5节详细介绍了双词话题模型及其参数估计方法；3.6节介绍了文档中话题比例的推断；3.7节给出了实验结果和讨论；3.8节对本章工作进行了小结。

3.3 相关工作

我们从两方面来介绍相关的工作：长文本话题模型和短文本话题模型。

3.3.1 长文本话题模型

在过去的10年间，长文本上的话题模型研究受到了广泛关注，也取得了众多成果，我们在第二节有较详细地介绍。这里仅仅列举其中和本文非常相关的几个工作。Wallach[143]提出的Bigram话题模型和BTM都没有使用词袋模型，前者利用到了Bigram信息去学习话题，为了建模词之间的顺序关系；而BTM则是利用Biterm信息去学习话题，没有考虑词之间的顺序关系。此外，Bigram话题模型还是基于文档建模

的，即假设每个文档一个话题分布，因此并不适用于短文本。另外两个和我们的工作比较相关的模型是最近提出来的正则化话题模型[110]与GPM模型（generalized Pólya model） [105]。这两个模型同样利用词共现统计量来改进话题的学习，但区别在于它们将词共现信息作为先验的方式来指导词的产生，而我们是直接建模词共现模式的产生。

3.3.2　短文本话题模型

由于短文本内容的稀疏性，一种常见做法是利用外部数据来辅助短文本的话题学习。如Phan等人[115]先在大规模外部数据（Wikipedia,开放网页等）上训练一个话题模型，再去推断短文本文档中的话题分布。Jin等人[79]提出的Dual-LDA模型同时学习短文本及相关辅助长文本语料上的话题，该方法能利用长文本上的话题知识来改进短文本上的话题学习。利用外部语料来辅助短文本话题学习虽然从一定程度上可以缓解短文本的数据稀疏性问题，但可能导致话题偏移现象，即学习到的话题并不能反映原短文本数据的真实分布。尤其是在辅助数据和原短文本数据相差较大的情况下。

近年来，随着社交媒体的迅速发展，话题学习方法也被应用社交媒体分析的更种任务中，如内容分析[119, 169]、事件监测[94] 和内容推荐[40, 116]等。然而，由于目前没有专门针对短文本的话题模型，很多工作还是简单地采用传统话题模型[151, 155]，或者简单地加以修改[119, 169]。Hong等人[73]对目前的Twitter数据上的话题学习方法做了一个经验性的对比，同时指出很有必要去针对短文本数据设计专门的话题模型。

3.4　模型描述

在话题模型当中，一个有意义的话题应该具有较强的语义相关性的词的集合。在没有先验知识的情况下，一个基本的假设就是两个词共现的次数越多越相关。为了说明这个假设的合理性，我们举个例子：“ipad”和“iphone”。对于不熟悉平果产品的人来说，可能并不知道这两个词的含义，但若观察到这两个词经常在一块出现，我们可以推断这两个词具有较强的语义相关性，因此很可能属于同一个话题。

传统的话题模型通过对文档的产生式建模，来隐式的利用文档级别的词共现关系来学习话题。这种方式的效果对文本长度比较敏感，因为短文本文档中的词共现关系不充分。然而，如果把语料中所有的词共现关系聚合在一起，共现词对的频率信息就会丰富起来。我们也就容易根据其频率大小来推断词之间的相关关系。基于这样的一个想法，我们提出了双词话题模型（BTM，Biterm Topic Model），它通过直接建模词共现模式来学习话题。

3.4.1　双词提取

在我们详细介绍我们的模型之前，我们介绍下双词的概念。双词指的是在同一上

下文中共现的无序词对，即词共现模式的实例。在自然语言处理当中，通常用词序列上的一个固定大小的滑动窗口来代表一个上下文。这里，由于短文本文档长度短，我们简单地把每个文档当做一个上下文。因此，对于一个短文本文档，我们其中任意两个不同的词组合构成一个双词。举个例子，从一个短文本“I visit apple store”中抽取出来的双词包括（这里忽略了停用词“I”）：“visit apple”，“visit store”，“apple store”。容易看出，双词的抽取仅需对文档集合进行一遍扫描即可完成。

3.4.2 模型定义

与传统话题模型通过建模文档的产生过程不同，BTM通过建模文档集合当中每个双词的产生过程来学习话题。其关键思想是两个词共现次数越多，它们越可能属于同一个话题。基于此，我们假设每个双词中的两个词都是独立地从同一个话题中产生，而该话题则是从一个全局的话题分布中产生。

给定一个短文本语料$\mathbb{D} = \{d_1, ..., d_{N_D}\}$，其对应的双词集合为$\mathbb{B} = \{b_1, ..., b_{N_B}\}$，其中$b_i = (w_{i,1}, w_{i,2})$。令$z \in [1, K]$为一个话题标示变量，$\boldsymbol{\theta}$表示语料上$K$个话题的分布，即$\theta_k = P(z = k)$，$K \times W$维矩阵$\boldsymbol{\Phi}$表示K个话题中词的分布，其中每个元素$\phi_{k,w} = P(w|z = k)$。为简单起见，本章我们对$\boldsymbol{\theta}$和$\boldsymbol{\phi_k}$均采用对称Dirichlet先验，其超参分别为$\alpha$和$\beta$。BTM的产生式过程描述如下：

1. 对整个语料采样一个话题分布：$\boldsymbol{\theta} \sim \mathrm{Dir}(\alpha)$
2. 对每个话题$k \in [1, K]$
   (a) 采样一个词分布$\boldsymbol{\phi}_k \sim \mathrm{Dir}(\beta)$
3. 对每个双词$b_i \in \mathbb{B}$
   (a) 采样一个话题$z_i \sim \mathrm{Multi}(\boldsymbol{\theta})$
   (b) 独立地采样两个词$w_{i,1}, w_{i,2} \sim \mathrm{Multi}(\boldsymbol{\phi}_{z_i})$

注意这里我们假设每个双词的产生式独立的。和词袋模型假设一样，该假设主要为了方便计算。图3.1（c）给出了BTM的概率图模型表示。

根据以上产生式过程，在$\boldsymbol{\theta}$和$\boldsymbol{\Phi}$给定的情况下，一个双词b_i的产生概率为：

$$\begin{aligned} P(b_i|\boldsymbol{\theta}, \boldsymbol{\Phi}) &= \sum_{k=1}^{K} P(w_{i,1}, w_{i,2}, z_i = k|\boldsymbol{\theta}, \boldsymbol{\Phi}) \\ &= \sum_{k=1}^{K} P(z_i = k|\theta_k) P(w_{i,1}|z_i = k, \phi_{k,w_{i,1}}) P(w_{i,2}|z_i = k, \phi_{k,w_{i,2}}) \\ &= \sum_{k=1}^{K} \theta_k \phi_{k,w_{i,1}} \phi_{k,w_{i,2}} \end{aligned} \tag{3.1}$$

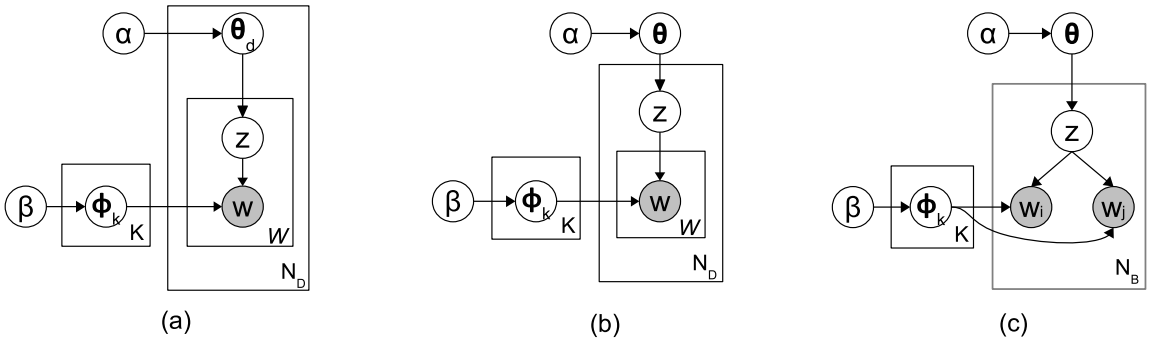


图 3.1: （a）LDA，（b）mixture of unigrams和（c）BTM的概率图模型表示。每个节点代表一个随机变量，其中有阴影的为已观察变量。矩形表示其内部的模型重复n次，n 由矩形内右下角变量给出。

给定超参α and β，我们可以把参数$\boldsymbol{\theta}$和$\boldsymbol{\Phi}$积掉，得到:

$$P(b_i|\alpha,\beta) = \int\int \sum_{k=1}^{K} \theta_k \phi_{k,w_{i,1}} \phi_{k,w_{i,2}} \mathrm{d}\boldsymbol{\theta}\mathrm{d}\boldsymbol{\Phi} \tag{3.2}$$

考虑整个双词集合$\mathbb{B}$，其似然函数为：

$$P(\mathbb{B}|\alpha,\beta) = \prod_{i=1}^{N_B} \int\int \sum_{k=1}^{K} \theta_k \phi_{k,w_{i,1}} \phi_{k,w_{i,2}} \mathrm{d}\boldsymbol{\theta}\mathrm{d}\boldsymbol{\Phi} \tag{3.3}$$

3.4.3　模型比较

这里我们讨论下BTM与其他两种常用的模型LDA [22]和Mix（mixture of unigrams）[113]的不同。后两种模型是目前短文本话题学习的常用方法 [119, 151, 169, 171]，因此可以视为BTM的直接竞争者。图 3.1展示了三种者的概率图模型表示。

如图 3.1（a）所示，LDA建模的是文档的产生过程：对文档d中的每个词w，先从一个文档相关的话题分布$\boldsymbol{\theta}_d$中采样一个话题z，然后从该话题中采样w。从图中可以看出，LDA对文档中词之间的相关性刻画是通过令它们的话题共享同一个话题分布$\boldsymbol{\theta}_d$来实现的。因此LDA过于依赖于文档内部的信息来推断文中每个词的话题z和$\boldsymbol{\theta}_d$。如果文档非常短，每个词的话题z和$\boldsymbol{\theta}_d$就难以估计准确。如此一来，也直接影响话题的词分布$\boldsymbol{\Phi}$的准确估计。

如图 3.1（b）所示，Mix采用了另一种方式来建模文档的产生过程。它把整个文档集合看成是一个话题的混合分布$\boldsymbol{\theta}$，然后假设文档中所有的词都采样自同一个从$\boldsymbol{\theta}$中采样的话题。Mix利用所有数据来估计一个全局的话题混合分布$\boldsymbol{\theta}$，可避免数据稀疏性问题。但是在Mix中，文档中所有的词属于同一个话题的约束过于严格。直观上讲，即使在短文本中，一个文档包含多个话题的情况也很常见。所以，该约束会导致Mix无法到文档中较细的话题，从而影响整体话题学习的效果。

Algorithm 2: 针对BTM的Gibbs采样算法

Input: topic number K, α and β, biterm set $\mathbb{B}$
Output: $\boldsymbol{\Phi}$, $\boldsymbol{\theta}$
Randomly initialize the topic assignments for all the biterms
for *iter* $= 1$ *to* N_{iter} **do**
  foreach *biterm* $b_i = (w_{i,1}, w_{i,2}) \in \mathbb{B}$ **do**
    Draw topic k from $P(z_i|\mathbf{z}_{-i}, \mathbb{B})$
    Update n_k, $n_{w_{i,1}|k}$, and $n_{w_{i,2}|k}$
Compute $\boldsymbol{\Phi}$ by Eq. (3.5) and $\boldsymbol{\theta}$ by Eq. (3.6)

从以上分析可以看出，LDA和Mix对短文本文档的建模要么过于复杂，要么过于简单（约束过强）。由于文档长度过短，直接对一条短文本内部的潜在话题结构的建模与推断是很困难的。既然如此，我们是否换可以换种思路来学习语料中的话题呢? 如图 3.1（c）所示，BTM没有去建模文档的产生，而是转而建模了语料中的双词的产生。和LDA相比，BTM通过学习一个全局的话题混合分布$\boldsymbol{\theta}$来避免了数据稀疏性问题；和Mix相比，BTM没有强制文档中所有的词必须属于同一个话题。

3.5 参数估计

本节中，我们给出BTM模型的参数估计算法来学习$\boldsymbol{\theta}$和$\boldsymbol{\Phi}$，并和LDA的参数估计算法的复杂度做了对比。

3.5.1 Gibbs采样算法

在似然函数式 (3.3)中，由于$\boldsymbol{\theta}$和$\boldsymbol{\Phi}$存在耦合关系，我们无法通过最大似然估计精确求解这两个参数。这里我们借鉴[61]中的collapsed Gibbs采样算法来近似求解$\boldsymbol{\theta}$和$\boldsymbol{\Phi}$。其主要思想是交替地去对待估计地随机变量进行后验采样，其中每次随机变量进行采样基于其他随机变量的赋值。具体地说，在BTM中我们需要对每个双词b_i采样一个话题。采样的后验分布为（详细推导过程见附录A.1）：

$$P(z_i = k|\mathbf{z}_{\neg i}, \mathbb{B}) \propto (n_{\neg i,k} + \alpha)\frac{(n_{\neg i,w_{i,1}|k} + \beta)(n_{\neg i,w_{i,2}|k} + \beta)}{(n_{\neg i,\cdot|k} + W\beta)^2}, \tag{3.4}$$

其中，$\mathbf{z}$表示所有双词的话题赋值，n_k表示属于话题k的双词个数，$n_{w|k}$表示词w赋予话题k的次数，$n_{\cdot|k} = \sum_{w=1}^{W} n_{w|k}$，下标$\neg i$表示不计双词$b_i$。式(3.4)的意义很直观：第一个因子表示语料中话题k的比例，第二部分则表示$w_{i,1}$和$w_{i,2}$属于话题k的概率的乘积。

针对BTM的Gibbs采样算法的详细步骤如算法2所示。首先，我们随机的分配一个话题给每一个双词作为初始状态。然后，在每次的迭代过程中，我们根据式(3.4)进行

表 3.1: LDA和BTM的复杂度对比

方法	时间复杂度	存储变量数目
LDA	$O(N_{iter}KN_D\bar{l})$	$N_DK + WK + N_D\bar{l}$
BTM	$O(N_{iter}KN_D\bar{l}(\bar{l}-1)/2)$	$K + WK + N_D\bar{l}(\bar{l}-1)/2$

采样，逐个更新每个双词的话题。经过充分多的迭代次数之后，我们开始收集充分统计量n_k和$n_{w|k}$。利用这些充分统计量，我们可以估计$\boldsymbol{\Phi}$和$\boldsymbol{\theta}$(详细推导参见附录A.2):

$$\phi_{k,w} = \frac{n_{w|k}+\beta}{n_{\cdot|k}+W\beta}, \tag{3.5}$$

$$\theta_k = \frac{n_k+\alpha}{N_B+K\alpha}. \tag{3.6}$$

3.5.2　复杂度分析

接下来，我们对比了在同样使用Gibbs采样算法的情况下，学习BTM与LDA所需的时间与内存开销。表3.1列出了这两个模型训练的时间复杂度和需存储变量数目。其中，$\bar{l}$表示单个文档中的平均词数。

在时间复杂度方面，Gibbs采样算法中的主要耗时部分是对采样条件概率的计算，需要$O(K)$时间。下面我们假设LDA和BTM的迭代次数都为N_{iter}，来比较它们的具体采样次数。在LDA中，我们需要对文档中的每个词采样它的话题，因此总采样次数为$N_{iter}KN_D\bar{l}$，故总的时间复杂度为$O(N_{iter}KN_D\bar{l})$。在BTM中，我们需要对每个双词来采样它的话题，因此总的时间复杂度为$O(N_{iter}KN_B)$。假设一个文档中含有l不同的词，其包含双词的个数为$l(l-1)/2$。于是，我们有[1]：

$$N_B \approx \frac{N_D\bar{l}(\bar{l}-1)}{2}.$$

我们可以看到BTM的时间复杂度约为LDA的$(\bar{l}-1)/2$倍。但是，考虑到短文本中的词数非常少，比如在Tweets2011数据集上$\bar{l}=5.21$，因此BTM的运行时间不会比LDA相差太多。

在空间复杂度方法，我们来看二者在训练时所需要存储的主要变量，即话题赋值变量和计数器变量。在LDA中，我们主要需要存储的变量包括：每个词w赋予给话题k的次数$n_{w|k}$，每个话题k在文档d中采样出来的次数$n_{k|d}$，以及每个词的话题赋值情况[69]，总计$N_DK+WK+N_D\bar{l}$个变量。在BTM中，我们主要需要存储的变量包括：词w赋予给话题k的次数$n_{w|k}$，每个话题中双词的个数n_k，以及每个双词的话题赋值情况，总计$K+WK+N_B$个变量。对比二者需要存储的总变量数目，我们发现LDA的内存需求会随着文档数目N_D和话题数目K的增长而急剧增长。因此在大数据集上，BTM要比LDA更省内存。

[1]因为短文本比较短，这里我们简单地认为每个文档中的不同词的个数一致。

3.6 文档中话题比例推断

在话题学习方法当中，除了学习话题之外(即$\{\boldsymbol{\phi}_k\}_{k=1}^K$)，我们还经常需要推断一个文档中的话题比例，即计算$P(z|d)$。然而，由于BTM没有建模文档的产生过程，我们不能直接从模型中学习出$P(z|d)$。幸运地是，我们可以通过双词的话题来推导出文档中的话题比例。

假设文档d包含N_d个双词$\{b_{i_j}|j \in [1, N_d]\}$，根据链式法则，我们有：

$$P(z|d) = \sum_{j=1}^{N_d} P(z, b_{i_j}|d) = \sum_{j=1}^{N_d} P(z|b_{i_j}, d)P(b_{i_j}|d). \tag{3.7}$$

给定一个双词$b_{i_j} = (w_{i_j,1}, w_{i_j,2})$，我们假设其话题$z$关于文档$d$条件独立，即满足$P(z|b_{i_j}, d) = P(z|b_{i_j})$。于是，上一个等式可以写成:

$$P(z|d) = \sum_{j=1}^{N_d} P(z|b_{i_j})P(b_{i_j}|d). \tag{3.8}$$

在式 (3.8)中，$P(z|b_{i_j})$可以根据贝叶斯定理计算得出：

$$P(z = k|b_{i_j}) = \frac{\theta_k \phi_{k,w_{i_j,1}} \phi_{k,w_{i_j,2}}}{\sum_{k'} \theta_{k'} \phi_{k',w_{i_j,1}} \phi_{k',w_{i_j,2}}} \tag{3.9}$$

而$P(b_{i_j}|d)$则可以通过经验估计得到:

$$P(b_{i_j}|d) = \frac{n_d(b_{i_j})}{N_d},$$

这里$n_d(b_{i_j})$双词b_{i_j}在文档d中出现的次数。

3.7 实验结果与分析

本节中，我们通过实验来验证BTM在短文本话题学习上的效果与性能。首先我们介绍实验中使用的数据集、基准方法以及评价方法。然后，我们对实验结果进行了展示和分析。

3.7.1 实验数据

因为我们关注的是短文本话题建模的实用性，我们采用了两个真实的短文本数据集。

- **百度问答数据** 我们从百度知道[2]网站上收集了648,514个问题，其中每个问题都由提问者指定了一个类别标签，总共是35个类别。

[2]http://zhidao.baidu.com

表 3.2: 实验数据集预处理后的统计信息

统计项	百度问答	Tweets2011
文档数目	189,080	4,230,578
词汇表大小	26,565	98,857
平均文档长度	3.94	5.21

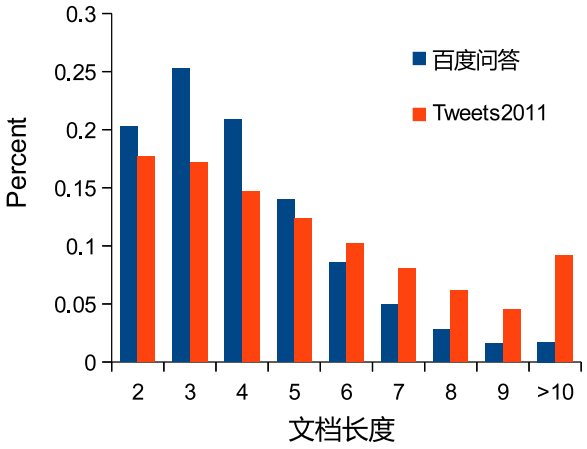


图 3.2: 实验数据集上的文档长度分布

- **Tweets2011数据** 该数据来源于TREC 2011微博任务[3]，是目前比较常用的短文本数据集。该数据集包含了在2011年1月23日-2011年2月8日期间采集的tweets，总共约包含1600万条。每条tweet除了其内容之外，还包含其作者、发布时间等信息。

为了减少数据中的噪音，我们对原始数据做了如下的预处理: (a)过滤非中文且非英文的字符; (b)所有的英文字符都转换成了小写; (c)去除了出现次数小于10的词; (d)去除了词数少于2的文档；(e)考虑到Tweets中转发功能导致很多文档是重复的，我们去除了重复的文档。表3.2中展示了预处理后的各数据集中的文档数、词汇表大小，以及文档的平均长度。图3.2展示了这两个数据集上的文档长度分布。我们可以看到，这些文档的长度非常短，其中百度问答数据的文档长度总体上要比Tweets2011更短。

3.7.2　基准方法

我们选择目前三种典型的短文本话题学习方法作为基准方法：

- **Mix** 即mixture of unigrams模型，它假设每条短文本中的词都属于同一个话题。
- **LDA** 标准LDA模型，将每条短文本作为一个文档。

[3]http://trec.nist.gov/data/tweets/

- **LDA-U** 将每个用户所发的短文本消息聚合成一个较长的虚拟文档，然后训练LDA。

其中LDA使用的是开源的GibbsLDA++[4]，其他程序也都是采用Gibbs采样算法用C++实现。所有的实验都在一台Intel Xeon 2.33 GHz CPU、16G内存的linux服务器上进行.

我们用网格搜索的方式来选择各个方法的参数。在Mix和BTM中，$\alpha = 50/K$，β=0.01; 在LDA中，α=0.05，β=0.01; Gibbs采样过程中的迭代次数设为1000，实验结果取10次的平均值。

3.7.3 评价方式

我们的实验目标是评价短文本话题建模方法的有效性，即学习到的话题结构是否满足人们的期望以及实际应用需求。在话题模型相关工作中，一种常用的评价指标是perplexity[22, 62, 63]。Perplexity通过用学到的模型参数来计算一部分保留数据的似然函数来判断一个话题建模方法的好坏，在这里并不适合我们。原因如下：1）由于BTM和其他几种方法建模的对象不一致，它们的似然函数不具备可比性；2）Perplexity与人们对话题模型的目标是脱节的[18]。话题模型应该更注重学习到的话题的可解释性、对实际应用的有效性等指标，而不是对数据的拟合能力。相关工作表明，perplexity较优的结果有时反而与人工评价相左[25]。因此，本文中没有采用perplexity作为评价指标，而是分别针对话题模型的两部分输出（即话题和文档中话题比例）的质量进行评价。

3.7.4 话题质量评价

针对话题的质量，我们这里采用一种目前比较常用的自动评价指标：PMI-score[111]。直观上讲，一个话题中的词相关程度越大，其可解释性越好。PMI-score通过借用大规模外部数据来计算一个话题中概率最大的前几个词相互之间的平均PMI（Pointwise Mutual Information）。PMI越高，说明这两个词越相关。因此如果一个话题的PMI-score越高，说明该话题的可解释性越好。

假设话题k中概率最大的T个词为$(w_1, ..., w_T)$，PMI-Score的定义如下:

$$\text{PMI-Score}(k) = \frac{1}{T(T-1)} \sum_{1 \le i < j \le T} \text{PMI}(w_i, w_j)$$

其中$\text{PMI}(w_i, w_j) = \log \frac{P(w_i, w_j)}{P(w_i)P(w_j)}$，$P(w_i, w_j)$和$P(w_i)$分别是双词$(w_i, w_j)$和词$w_i$在外部数据中出现的概率。这里，在百度问答语料上的评价，我们采用的外部数据是500万篇百度百科数据；在Tweets2011上的评价，我们采用的是4百万篇Wikipedia数据。

[4] http://gibbslda.sourceforge.net/

表 3.3: LDA、LDA-U、Mix和BTM的PMI-Score（越大越好）

	K	50			100		
数据	方法	Top5	Top10	Top20	Top5	Top10	Top20
百度知道	LDA	2.15 ± 0.05	1.70 ± 0.03	1.40 ± 0.04	2.16 ± 0.04	1.71 ± 0.03	1.39 ± 0.02
	Mix	2.28 ± 0.06	1.82 ± 0.03	1.43 ± 0.03	2.34 ± 0.05	1.80 ± 0.04	1.40 ± 0.03
	BTM	$\mathbf{2.34 \pm 0.05}$	$\mathbf{1.88 \pm 0.03}$	$\mathbf{1.48 \pm 0.03}$	$\mathbf{2.42 \pm 0.06}$	$\mathbf{1.89 \pm 0.05}$	$\mathbf{1.49 \pm 0.03}$
Tweets2011	LDA	2.61 ± 0.06	1.93 ± 0.04	1.77 ± 0.02	2.64 ± 0.06	2.02 ± 0.04	1.78 ± 0.02
	LDA-U	2.63 ± 0.02	2.14 ± 0.06	1.77 ± 0.02	2.72 ± 0.02	2.20 ± 0.02	1.79 ± 0.01
	Mix	2.72 ± 0.07	2.19 ± 0.03	1.83 ± 0.02	2.85 ± 0.04	2.28 ± 0.02	1.83 ± 0.02
	BTM	$\mathbf{2.74 \pm 0.04}$	$\mathbf{2.26 \pm 0.04}$	$\mathbf{1.86 \pm 0.02}$	$\mathbf{2.88 \pm 0.02}$	$\mathbf{2.33 \pm 0.04}$	$\mathbf{1.87 \pm 0.03}$

表 3.4: Tweets2011数据集中“job”话题中的前20个词（第二行）与排名1000-1021的词（第三行），其中斜体词是经过人工判断发现和“job”不太相关的词

LDA	LDA-U	Mixture of unigrams	BTM
job jobs business web	job jobs *design*	jobs job business	jobs job manager business
website google	manager project *web*	marketing *social media*	sales hiring service services
design online marketing	*website site* business	online *web design*	project company senior
site blog project	service company	*website* manager	engineer management
manager search *www*	hiring *www* support	*blog* project seo	marketing nurse
company service	sales services london	internet sales tips	office assistant center
sales services *post*	*blog* senior engineer	company *site* hiring	customer development
nonprofit gallery announced	expertise unemployed	understand rep industrial	springfield mlm recruit
presence published	med iii *host* educational	*sustainability rankings*	*oil* req unemployment
converting select reps	*fort tags apps*	scholarships stay *single*	processing *overview*
requirement mgr territory	*assignments* labor	campus extra *cheap* 101	awards recruiters *ict finish*
recruiters *power involved*	*introduction* leads github	vp *relationships* beginners	entrepreneur comp assist
announce poster larry	assurance *avon manchester*	colorado compliance	1000 *alliance* locations
dynamics feeds bristol	*starting automotive table*	*face winning* mechanical	patent auditor

为了综合评估一个话题学习方法学习到的话题质量，我们计算这些话题的平均PMI-score，即$\frac{1}{K}\sum_k \text{PMI-Score}(k)$。表3.3列出来各方法在百度问答数据和Tweets2011上的平均PMI-score，其中T分别取5，10，20。我们可以看到BTM的PMI-score一致高于其他方法(P-value$<$ 0.01)。Mix学习的话题质量也优于LDA，但并不显著。LDA-U的效果同样优于LDA，说明文档聚合能改进LDA在短文本上所面临的数据稀疏性问题。但我们也发现LDA-U相对于BTM和Mix来说改进并不大。

我们进一步对各个方法学习到的话题进行了定性分析。由于空间限制，我们采用[29]中的方式随机的从这些方法学到的一些共同话题中抽取了两个话题（分别关于”job”和“snow”）进行案例分析。对于每个话题，我们除了列出来其中概率最大的20个词，还列出来20个概率不大的词（按概率排序，排在1001 到1020 之间）来更全面的判断一个话题的质量。一个质量更好的话题，不仅仅其前20个词语义更相关，其20个概率不大的词应该也更相关。结果如表3.4和表3.5所示。

表 3.5: Tweets2011数据集中“snow”话题中的前20个词（第二行）与排名1000-1021的词（第三行），其中斜体词是经过人工判断发现和“job”不太相关的词

LDA	LDA-U	Mixture of unigrams	BTM
snow *car* weather	snow weather cold	snow weather cold	snow cold weather
cold *drive* storm	winter ice storm rain	storm winter ice rain	early stay ready ice
winter ice *road bus*	stay warm *due car*	warm degrees stay	winter storm hour
driving rain *ride* traffic	*closed* coming spring	sun spring safe blizzard	hours *weekend* warm
cars safe closed	*drive traffic safe*	coming wind cyclone	late coming spring
due warm *train*	sun blizzard city	*chicago* freezing *inches*	rain *tired* sun hot
western *dmv* covering	locations sunset drizzle	*australian thankful* station	temperature cyclone
a4 push pulling	*mississippi interstate*	*stops groundhogday*	warmth issued colder
milwaukee remains *pace*	residents *portland*	*possibly cleveland*	*mood couch* snows
idiots 95 *commuter*	*students* fireplace *letting*	*traveling sidewalk* covering	pre *traveling polar*
buick owner cta	*yuck ton counties* signal	predicting ten *grass*	*outages* umbrella filled
transmission cyclist	*counting* blankets pushed	*meant double* affect *zoo*	*yawn outage* flurries
flurries *camping tyre*	*3pm springfield venture*	*schedule* blew *causing*	online gloves speed

在表3.4中，我们很容易看出每个方法的前20个词大部分都和“job”比较相关。但在LDA中，出现的“web”，“website”，和“google”等词更应该属于一个关于“website”的话题。LDA-U和Mix中的结果稍微比LDA好点，但还是包含了一些不相关词，如“website”和“www”。对比第三行排名在1000-1020词中，我们发现LDA中和“job”相关的词是最少的。相反，BTM中无论是前20个还是排名在1000-1020的20个词中，和“job”相关的词的数目是最多的。从表3.5中，我们可以得到同样的结论。以上结果表明，LDA由于受短文本数据稀疏性影响，其学习到的话题质量不佳，而BTM发现的话题的语义内聚度更高、可解释性更好。

3.7.5 文档中话题比例的评价

对于文档中话题比例的评价，我们通过其在文本分类当中的有效性来评价其质量。这里，我们把话题学习看成是一种降维方法，即把文档从词空间降维到概率话题空间。降维后损失的类别区分信息越少，我们认为文档中的话题比例学习的越好。具体步骤如下：首先，我们将每个文档d表示成一个向量$[P(z=1|d),...,P(z=K|d)]$。然后我们随机地将文档集合按4 : 1的比例进行划分训练集和测试集。我们用线性SVM分类器LIBLINEAR[5]来进行分类。

注意在Tweets2011数据集上文档没有预定义的类别标签。由于tweet信息非常短而且通常书写不正规，手工标注极为困难。幸运地是，在部分tweet中，用户以hashtag（形如“#keyword”的词）的形式对消息进行了标注。通过观察，我们发现这些hashtag大致可分为三类：1）标注事件或话题；2）说明消息的类型，如

[5]http://www.csie.ntu.edu.tw/~cjlin/liblinear/

表 3.6: Tweets2011数据中选择用来分类评价的50个Hashtags

jan25	superbowl	sotu	wheniwaslittle	mobsterworld	jobs	
agoodboyfriend	bieberfact	glee	lfc	rhoa	itunes	thegame
celebrity	tcyasi	americanidol	cancer	socialmedia	jerseyshore	
photography	jp6foot7remix	factsaboutboys	meatschool			
libra	android	sagittarius	thissummer	tnfisherman	sagawards	
ausopen	bears	weather	jaejoongday	skins	bfgw	fashion
pandora	realestate	teamautism	travel	nba	football	marketing
design	oscars	food	dating	kindle	snow	obama

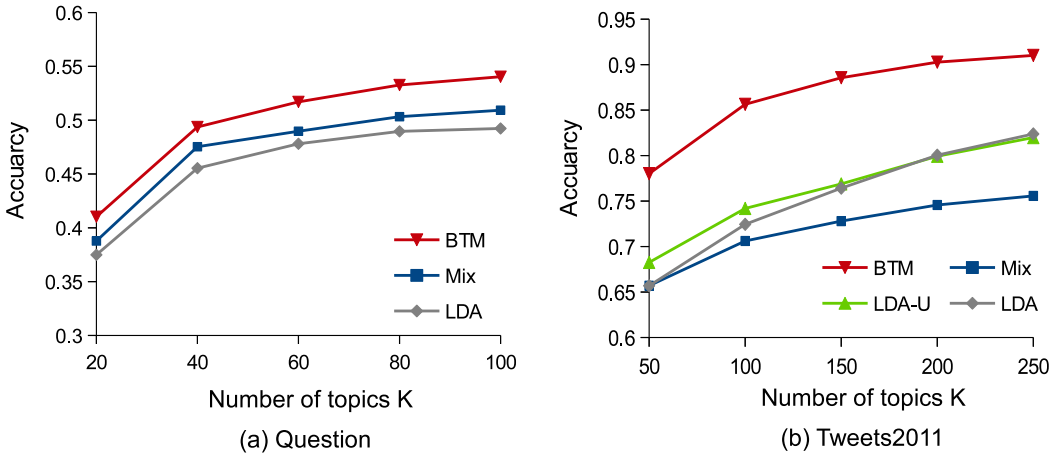


图 3.3: BTM，Mix和LDA在（a）百度问答数据和（b）Tweets2011 数据上的文本分类实验对比

“#ijustsayin”和“#quote”；3）实现某些功能，如“#mark”用来对消息加书签，“#fb”用来将消息同步到facebook。这三类hashtag中只有第一类中的hashtag是和消息语义相关的。因此，我们仅从第一类hashtag中手工挑选了50个高频的hashtag作为50个分类。这50个hashtag如表3.6所示。然后把包含这些hashtag 的文档用来做分类实验。注意，如果一个文档包含其中多个hashtag，则忽略该文档。

在百度问答数据和Tweets2011数据上的文档分类效果如图3.3所示。我们发现1) BTM显著优于其他方法(P-value < 0.01)。2) Mix在百度问答数据上的表现要比LDA好，但在Tweets2011数据上反而不如LDA。主要是由于百度问答数据的文本长度总体上要比Tweets2011上的要短，因此前者的文档更可能只包含一个话题，更符合Mix的模型假设。3) LDA-U相对于LDA的改进并没有 [73]中提到的大。经过对比二者的数据，主要的原因是因为[73]中仅仅抽取了发消息较多的用户进行试验。而在Tweets2011数据中，有63.3%的用户只发了一条消息，另外只有2.1%的用户发的消息数目超过9条，如图3.4所示。所以按用户来聚合短文本文档，对LDA的效果改进并不大。

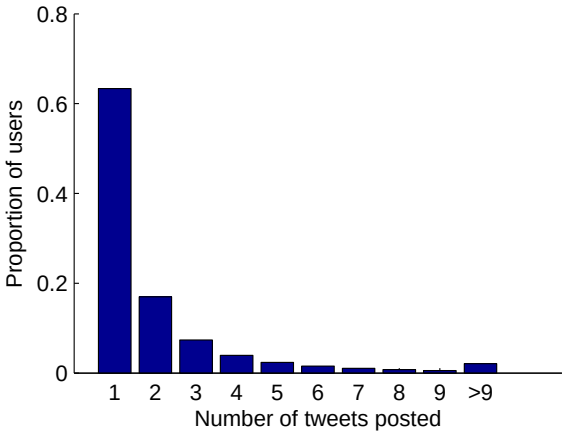


图 3.4: Tweets2011数据中用户发的消息数目分布

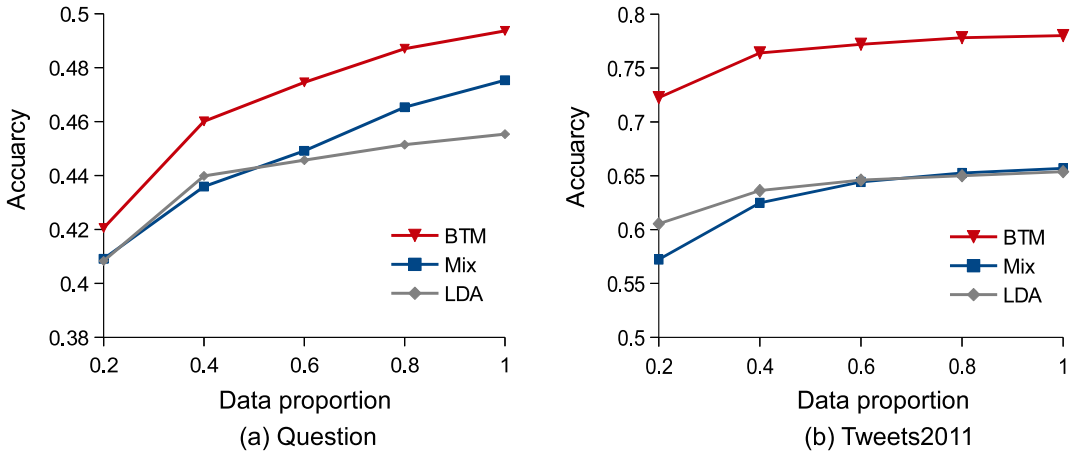


图 3.5: 不同大小的数据集对BTM结果的影响：（a）百度问答数据，（b）Tweets2011数据

3.7.6 数据集大小的影响

我们接下来看不同大小的数据集对这几种话题学习方法的影响。我们随机地从原数据集中采样一部分数据（采样比例分布为0.2，0.4，0.6，0.8）作为分类实验数据。我们在这些采样的数据集上先学习话题以及文档的话题比例，然后再用LIBLINEAR进行分类实验，实验步骤如前所述。最后的分类结果随数据集大小的变化如图3.5所示。其中，在百度问答数据中，话题个数设置为$K = 40$；在Tweets2011上，话题个数设置为$K = 50$。可以看出，所有的方法随着数据集的大小的增加，分类准确度也在增加。这说明，数据集越大，越能提高话题学习的效果。但同时，我们也发现，BTM在不同大小的数据集上总是明显优于其他方法。Mix虽然在百度问答数据上优于LDA，但在Tweets2011上相对于LDA没有优势，尤其是在数据非常少的情况下。这说明Mix对数据集大小很敏感。相对于其他两种方法，LDA随着数据集大小的增加，效果改进较小，

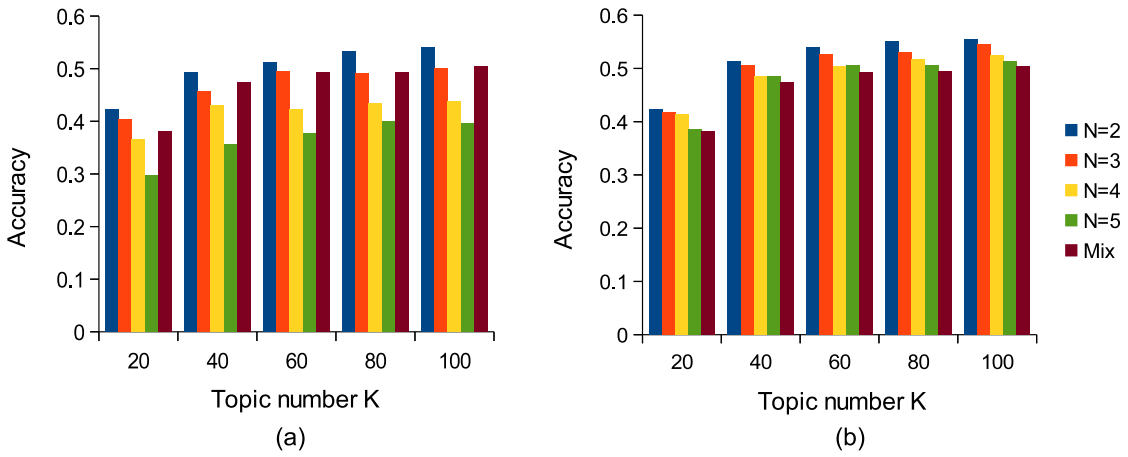


图 3.6: 双词话题模型与多词话题模型的分类效果对比：（a）N-term不带权重，（b）N-term带权重

而且最后趋于平稳。这是因为，数据集增大改变的是文档的数目，但文档的长度依然很短，所以并不能克服LDA所面临的数据稀疏性问题。

3.7.7 Biterm VS. N-term

如果把一个话题在文中采样得到的实例称为一个语义单元的话，我们会发现LDA中一个词是一个语义单元，Mix中一条短文本是一个语义单元，而BTM中一个biterm是一个语义单元。很自然的，我们可能会问如果用多个词作为一个语义单元，效果会怎样？为了回答这个问题，我们从双词话题模型扩展到多词话题模型（N-term Topic Model），即用长度为N的无序共现词组（称为N-term）来替代BTM中长度为2的双词。同样地，我们假设一个N-term 中的N 个词都是从同一个话题中独立地产生出来的。注意，这里$N > 1$。因为$N = 1$的时候，所有的词都是独立的。特别地，$N = 2$的时候，NTM就退化成了BTM；当N大于或等于文档集合中最大的文档长度的时候，NTM就退化成了Mix。因为此时一个N-term就是一个文档。

我们仍然用分类实验来评估NTM的效果。图3.6（a）展示了在百度问答数据上不同的N对效果的影响。我们发现随着N增加，NTM的性能会逐渐降低。这个现象是可以理解的，因为随着N-term中词数的增加，这些词属于一个话题的假设也就越来越可能不成立。但奇怪地是，但$N > 3$的时候，NTM的性能反而比Mix还要差。进一步调查发现，但$N > 3$时，不同长度文档产生N-term的数目会差别很大。比如，一个长度为L的文档，会产生$\binom{L}{N}$个N-term。我们考虑$N = 3$的情况，一个长度为3的文档会产生一个N-term，但一个长度为10的文档会产生120个N-term。这样一来，导致数据中一些较长的文档产生的N-term会过多，扭曲原数据的分布，从而影响效果。

一种比较简单的修正方法是对不同长度文档产生的N-term加上不同的权重。这里，我们将一个长度为L的文档中产生的N-term的权重设为$1/\binom{L-1}{N-1}$。其目的是保持语料

表 3.7: BTM和LDA在Tweets2011数据上每次迭代所需时间（单位：秒）

K	50	100	150	200	250
LDA	38.07	74.38	108.13	143.47	178.66
BTM	128.64	250.07	362.27	476.19	591.24

表 3.8: BTM和LDA在Tweets2011数据上的内存消耗（单位：MB）

K	50	100	150	200	250
LDA	3177	5524	7890	10218	12561
BTM	927	946	964	984	1002

中的词的分布不变。容易看出，文档的长度越长，其产生的N-term的权重就越低。我们在加了权重的N-term集合上重新训练NTM模型[6]。调整后的结果如图3.6（b）所示。我们发现NTM的效果仍然会随着N的增加渐渐降低，但是此时是逐渐趋向于Mix的。和图3.6（a）相比，N-term加上权重后对NTM的效果有明显改进，尤其N比较大的时候。但是$N = 2$的时候，改进相对比较少。因为$N = 2$的时候，不同长度的文档产生的biterm的数目相对来说差别不大。

3.7.8 效率对比

为了对比BTM与LDA的效率，我们在表3.7列举了它们在Tweets2011数据上一次迭代所需的平均时间。我们发现，BTM的所需的时间大约为LDA的3倍，这是3.5.2节的时间复杂度分析相符的。在表3.8中，我们展示了BTM和LDA在Tweets2011数据上的内存消耗。我们可以看到，LDA消耗的内存随着话题数目K的增加而快速增长。但$K > 200$的时候，LDA消耗的内存超过BTM的10倍。相反，BTM消耗的内存增长非常小。这是因为其主要的内存消耗是存储所有的biterm上，这一部分内存开销还可以通过在Gibbs采样时即时生成双词来进一步降低。

3.8 小结

随着互联网上短文本数据的增多，研究有效的短文本话题建模方法来自动分析短文本当中的潜在语义信息，成为很多应用中迫切需要解决的一个问题。短文本由于缺乏词频和上下文信息，给传统基于文档建模的话题模型带来严重的数据稀疏性问题。针对这一问题，本章中我们提出了一种新的概率话题模型，即双词话题模型（BTM）。据我们所知，这也是目前第一个通用的短文本话题模型。BTM的特点是通过直接建模语料中的词共现模式来学习话题，从而很好的利用全局丰富的词共现模式来避免单个

[6] 注意，在具体实现过程中，我们只需要简单的修改Gibbs采样算法中的计数器变量的类型，即将整形调整为浮点型。然后每次更新话题赋值时，用N-term的权重去提到原来的数目即可。

文档中词共现模式较少所导致的数据稀疏性问题。

我们通过在两个真实短文本数据集上的大量实验验证了BTM的效果与效率，发现BTM能比现有方法学到解释性更好的话题，同时学习到的文档中话题比例也能更好地保持文档中原有的区分性。除此之外，BTM比较简单，容易实现，而且消耗内存相对较少，因此BTM具有很高的实用性。

本章的研究成果已发表于2013年国际万维网大会（WWW13），论文题目为“A Biterm Topic Model for Short Texts”。

第四章　在线话题建模

4.1　引言

互联网上短文本信息的另外一个重要特点就是动态性。在微博等在线应用中，每时每刻都有大量的消息源源不断被产生，导致短文本的数据规模随时间呈线性增长，而且其内容也在不断地更新与演化。本章我们研究如何针对这种大规模动态数据进行话题建模。

大规模动态数据对话题建模方法提出了新的需求：我们不仅要高效地发现这些大规模短文本数据中的话题，还能时刻追踪话题的演化。为此，我们在双词话题模型的基础上提了两个在线话题学习算法：oBTM和iBTM。它们只需要利用当前最近的一小部分历史数据去增量更新模型，从而将模型更新所需的时间和空间降低到常数级别，同时还能及时地根据最新数据来动态更新话题的内容。我们通过在大规模微博（包括Twitter和新浪微博）数据上验证了这两种在线话题学习算法的效果与性能。

4.2　概述

在社交媒体时代，在微博、微信等在线应用当中，海量的用户每时每刻都会产生大量的短文本信息。根据新浪微博官方数据，当前新浪微博每月有1.438亿活跃用户，平均每天约一亿条的微博产生。而据国外媒体报道，目前全球社交媒体平台每小时的留言大约有15亿条，社交媒体用户每月分享的内容约300亿条，这些内容包括评论、观点、视频、播客和图片等。在这些在线应用当中，短文本数据不仅规模大，而且其数据量是随着时间的推移不断增长的。另外，这类短文本具有很强的时序特征，它们的内容也是随时间动态变化的，其中的话题也在不断地演化。这些在线应用产生的海量短文本数据中蕴含着丰富地价值信息，如新闻线索、商业情报、用户兴趣爱好等。面对如此大规模流式短文本数据，人工分析显然不现实。利用话题学习方法可以自动从这些数据中找出潜在的话题，对于舆情分析、商业情报收集等应用有重要意义。

在线话题建模研究的是如何从这些大规模流式的短文本信息中有效发现话题，并能追踪话题的变化。它对现有的话题建模方法带来了更多的挑战。首先，学习算法的时间和空间复杂度必须较低，否则难以处理如此大规模的数据。其次，我们还希望学习算法必须自适应数据的变化，即能根据最新的数据动态的更新话题内容。比如在微博数据当中，越是最新的话题越有价值。这也要求学习算法训练的时间不能过长，否则话题学习的结果难以赶上数据的变化。

然而，一般地话题模型的求解算法复杂度都比较高，而且多是以批处理的方式学习话题，因此并不能直接应用到如何大规模流式数据。其一，批处理方式需要多次遍

历整个数据集来不断更新模型的参数。在大规模数据下每次遍历所有样本需要花费的时间非常长。而且每次遍历过程中，它需要维护所有的模型参数与变量，这将需要巨大的空间开销。其二，批处理算法到的是静态的模型，不能自适应话题的动态变化。如果要更新模型，批处理方法需要对所有数据进行重新计算，代价随着数据的增加会越来越高。

本章中，我们针对短文本在线话题建模问题，在双词话题模型（BTM）的基础上提出了两种在线话题学习算法：oBTM（online BTM）和iBTM （incremental BTM）。这两种在线话题学习算法的基本思想是每次仅存储最近所接收到的数据用来持续更新模型参数。由于每次参数更新只用到了部分数据，在大规模数据下，在线话题学习算法节省了大量的时间和空间开销。除此之外，在线话题学习算法也能随新数据的到来，即时更新模型以适应数据的动态变化。

我们通过在Tweets2011和微博数据集上对着这两种在线话题学习算法的有效性进行了实验验证。我们发现，oBTM和iBTM的话题学习效果和BTM的批处理学习算法相差不大，但其更新模型所需的内存和时间比批处理方式要少很多，特别是在数据不断增加的情况下。另外，我们也和在线LDA算法[30]进行了对比，发现oBTM和iBTM的话题学习效果要明显优于在线LDA算法。我们也发现oBTM和iBTM的学习到的话题能动态反映数据的变化，说明这两种算法具备追踪话题的演化的能力。

本章其余内容组织如下：4.3节介绍了相关工作；4.4小节详细介绍了基于BTM的两种在线话题学习算法；4.5节给出了实验结果和讨论；4.6节对本章工作进行了小结。

4.3 相关工作

根据我们的调研，目前还没有专门针对短文本的在线话题学习的研究，但基于长文本的在线话题学习已有不少工作。另外，随着对社交媒体研究的兴起，近年来也有一些应用涉及在线话题学习。接下来，我们分别从在线话题学习方法和社交媒体中相关应用两方面对相关工作进行概括。

4.3.1 在线话题学习方法

近年来，随着大数据的热潮兴起，在线话题学习算法的研究也越来越受到人们的重视。目前主要的在线话题学习方法都是围绕LDA进行的，大致可以分为两类：基于Gibbs采样的方法和基于变分推断的方法。较早期的工作[11, 131]中使用了一种基于Gibbs采样的简单增量LDA学习算法。他们首先用批处理算法训练一个LDA 模型，然后当每个新文档到来时，固定当前话题参数$\mathbf{\Phi}$，对新文档中每个词的话题赋值进行采样。采样完后，再用采样结果取更新话题参数$\mathbf{\Phi}$。该方法的效果严重依赖于预先用批处理方式训练好的模型的好坏。由于对每个文档只进行一次采样，后续采样结果随机性太大，而且随着后续文档的增加，会导致误差累积，从而导致话题学习结果会越来

越差。AlSumait等人[4]于2008年提出了一种在线LDA算法。其思想是将数据流按时间分片，然后每一个时间片内用批处理的方式训练一个单独的LDA模型，但之前时间片的模型参数会用来构造后续时间段的LDA模型的先验，从而影响后续时间片内的话题学习。2009年，Canini等人[30]提出了另外一种基于Gibbs采样的增量LDA学习算法，该算法不需要预先用批处理方式作为初始化。该方法是decayed MCMC[99]方法的一个实例。它每次对新文档进行Gibbs采样的时候，同时还会随机地从以往的文档中挑选一些进行重采样。重采样过程可以利用后续数据来修正之前采样的结果，从而减少由于数据不充分导致的采样误差。在同一篇文章中，Canini等人[30]还提出了另外一种基于粒子滤波（Particle filter）的在线话题学习方法，方法同时进行多个增量Gibbs 采样。同时会根据采样结果对样本的匹配程度计算一个权重，最后对所有的采样结果加权。虽然该方法比增量LDA更复杂，但我们在实验过程中发现，该方法的一个很大的问题是权重随着样本等增加会变得非常陡峭，最后甚至会退化成增量LDA。基于变分推断的在线话题学习方法方面，Hoffman等人[70]在原变分推断算法的基础上首次提出了LDA的在线变分推断方法。其思想是和基于Gibbs采样的方法类似，即每来一个新文档，首先固定话题相关参数来估计新文档对应的参数，然后用该参数反过来去估计一个话题相关参数，并与以往的话题相关参数加权得到最新的话题话题参数。从理论上讲，通过动态设置合适的权重大小可以使得该算法收敛。

除了以上两类方法之外，还有一些非LDA的话题学习方法，主要是基于随机梯度下降的思想，如Zhang等人[166]针对稀疏话题编码（sparse topical coding）提出的稀疏在线话题学习方法。

4.3.2　社交媒体中的相关应用

在线话题学习方法在社交媒体中应用比较多的是话题检测与追踪。但由于缺乏专门针对短文本的在线话题学习方法，这些工作中基本还是采用了一些以往的在线话题学习方法。如Lau等人[86]应用论文[4]中提出用在线LDA方法来发现和检测Twitter中的趋势。其思想是按不同时段内来动态更新模型，同时比较两个时间内的话题结果差异。若差别不大，则认为是旧话题；否则认为是新话题。Saha等人[123]采用了在线NMF（Non-negative Matrix Factorization）的方式来学习社交媒体中话题的演化。其思想是在每处理一批新数据的时候，在原结果矩阵上进行维度扩充，以学习到新话题。该方法的一个缺陷在于不能有效地控制话题的数目，也就是说，其话题的数目会随着数据增长越学越多。

4.4　在线BTM学习算法

本章我们依次介绍两种基于双词话题模型（BTM）的在线话题学习算法：oBTM (online BTM)和iBTM（incremental BTM）。这两种在线话题学习算法在更新BTM模型

Algorithm 3: oBTM（Online BTM）算法

Input: $K, \alpha, \beta, \lambda$, Biterm sets $\mathbf{B}^{(1)}, ..., \mathbf{B}^{(T)}$
Output: $\{\mathbf{\Phi}^{(t)}, \boldsymbol{\theta}^{(t)}\}_{t=1}^{T}$
Set $\boldsymbol{\alpha}^{(1)} = (\alpha, ..., \alpha)$ and $\{\boldsymbol{\beta}_k^{(1)} = (\beta, ..., \beta)\}_{k=1}^{K}$
for $t = 1$ *to* T **do**
  Randomly assign topics to biterms in $\mathbf{B}^{(t)}$
  for $iter = 1$ *to* N_{iter} **do**
    foreach *biterm* $b_i = (w_{i,1}, w_{i,2}) \in \mathbf{B}^{(t)}$ **do**
      Draw topic k from Eq. (4.1)
      Update $n_k^{(t)}$, $n_{w_{i,1}|k}^{(t)}$, and $n_{w_{i,2}|k}^{(t)}$
    Set $\boldsymbol{\alpha}^{(t+1)}$ and $\{\boldsymbol{\beta}_k^{(t+1)}\}_{k=1}^{K}$ by Eq.(4.2) and Eq.(4.3)
  Compute $\mathbf{\Phi}^{(t)}$ by Eq.(3.5) and $\boldsymbol{\theta}^{(t)}$ by Eq.(3.6)

的时候，只需要存储和计算一小部分最近的数据，极大了提高了BTM模型处理大规模数据的能力。这两种算法都是基于Gibbs采样来实现的。其中，oBTM适合于对数据分批次（如按天）进行模型更新的场景，而iBTM则适合于对模型进行即时更新的场景。

4.4.1 oBTM（Online BTM）算法

oBTM算法的设计是受在线LDA算法[4]的启发。我们首先对流式短文本数据按时间（如天）切分成多个时间片，每个时间片对应一个文档子集。oBTM算法的主要思想是对每个时间片内的文档子集用一个单独的BTM模型按批处理的方式学习其中的话题。同时，为了延续话题学习结果，我们记录当前模型学习到的充分统计量，即话题中双词计数$n_k^{(t)}$与词的话题赋值计数$n_{w|k}^{(t)}$，并用它们来构造下一个时间片内的BTM模型的Dirichlet先验。

整个oBTM算法的流程如3算法所示，下面我们开始详细介绍其过程。在预处理阶段，我们需要将每个时间片t内的文档集合转换成双词集合$\mathbf{B}^{(t)}$。对于初始时间片，我们仍然采用对称Dirichlet先验$\boldsymbol{\alpha}^{(1)} = (\alpha, ..., \alpha)$和$\boldsymbol{\beta}_k^{(1)} = (\beta, ..., \beta)$来训练BTM模型。而其他时间片的Dirichlet先验则通过之前时间片内模型训练结果来构造。这里，我们用K维向量$\boldsymbol{\alpha}^{(t)}$来表示$\boldsymbol{\theta}^{(t)}$的先验，用$W$维向量$\boldsymbol{\beta}_k^{(t)}$来表示$\boldsymbol{\phi}_k^{(t)}$的Dirichlet分布，其中$^{(t)}$表示第$t$个时间片。

我们考虑第t个时间片内oBTM算法的具体过程。给定$\boldsymbol{\alpha}^{(t)}$和$\{\boldsymbol{\beta}_k^{(t)}\}_{k=1}^{K}$之后，我们反复为该时间片内的每个双词$b_i \in \mathbf{B}^{(t)}$采样其话题。此时，具体地采样的条件概率分布

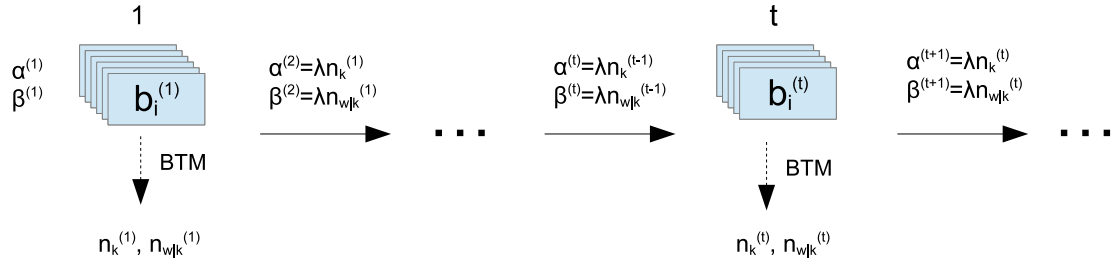


图 4.1: oBTM算法对短文本数据流的处理流程

如下：

$$P(z_i = k|\mathbf{z}_{-i}^{(t)}, \mathbf{B}^{(t)}, \boldsymbol{\alpha}^{(t)}, \{\boldsymbol{\beta}_k^{(t)}\}_{k=1}^K) \propto (n_{-i,k}^{(t)} + \alpha_k^{(t)}) \frac{(n_{-i,w_i|k}^{(t)} + \beta_{k,w_i}^{(t)})(n_{-i,w_j|k}^{(t)} + \beta_{k,w_j}^{(t)})}{[\sum_{w=1}^{W}(n_{-i,w|k}^{(t)} + \beta_{k,w}^{(t)})]^2}, \tag{4.1}$$

经过足够多的迭代次数，采样过程完成之后，我们统计每个词的话题赋值情况，可以得到以下充分统计量：每个话题k中双词的个数$n_k^{(t)}$和词w赋予给话题k的次数$n_{w|k}^{(t)}$。这两个计数器涵盖了当前时间片内的话题统计信息。其中，前者表达了话题k的流行度，后者表达了词w 与话题k 的相关度。考虑到数据的时序相关性，下一个时间片中的话题分布应该和当前的时间片内的话题分布比较接近。因此，我们用$n_k^{(t)}$和$n_{w|k}^{(t)}$来构造下一个时间片内BTM模型参数$\boldsymbol{\alpha}^{(t+1)}$和$\{\boldsymbol{\beta}_k^{(t+1)}\}_{k=1}^K$的Dirichlet先验：

$$\alpha_k^{(t+1)} = \alpha_k^{(t)} + \lambda n_k^{(t)}, \tag{4.2}$$

$$\beta_{k,w}^{(t+1)} = \beta_{k,w}^{(t)} + \lambda n_{w|k}^{(t)}, \tag{4.3}$$

其中参数$\lambda \in [0,1]$是一个衰减因子。根据Dirichlet-multinomial共轭性质[4, 16]，超参$\alpha_k^{(t)}$和$\beta_{k,w}^{(t)}$可以分别看成是$n_k^{(t)}$和$n_{w|k}^{(t)}$的先验观察数目。因此，式(4.2-4.3)可以理解成将当前时间片的充分统计量作为下一个时间片内对应充分统计量的先验观察数目。在此构造过程中，衰减因子λ控制着先验的强度。特别地，当$\lambda=0$的时候，不同时间片内的模型完全独立，即只利用自己时间片内的数据来学习话题；当$0<\lambda<1$，历史结果的影响会随着时间片的推移而指数级的衰减; 当$\lambda=1$，历史结果的影响永远都不会衰减，而是不断地累积。图4.1给出了oBTM算法对短文本数据流处理流程的一个示意图。

oBTM算法原理比较简单，且易于实现——仅仅需要对每个时间片单独运行一个BTM的Gibbs采样器。同时，由于历史数据是作为先验存在的，我们实际上并不需要额外地对历史数据进行计算和存储，所以oBTM算法的时间和空间开销仅仅与当前时间片内的数据规模相关。另外，oBTM算法的缺点是只时间片内所有的数据都收集完之后，才能去更新模型，难免对最新话题的学习有延时。在有些应用中，如微博实时话题检测，我们更希望当新文档到来的时候，立即更新模型。此时，oBTM算法便无法胜任。为此，我们接下来介绍另外一种更适合即时更新的在线话题学习算法——iBTM。.

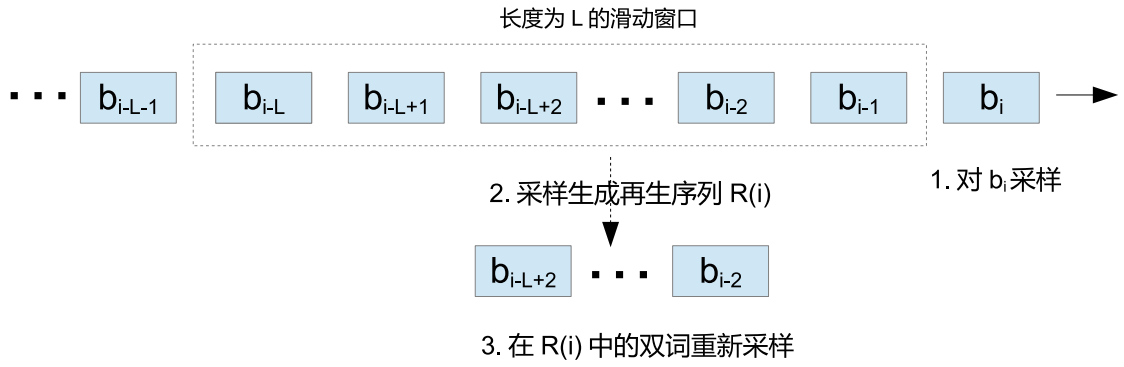


图 4.2: iBTM算法对短文本数据流的处理流程

Algorithm 4: iBTM（Incremental BTM）算法

Input: K, α, β, Biterm sequence $\mathbf{B} = \{b_1, ..., b_N\}$
Output: $\boldsymbol{\Phi}, \boldsymbol{\theta}$
for $i = 1$ *to* N **do**
  Draw topic k from $P(z_i|\mathbf{z}_{i-1}, \mathbf{B}_i)$
  Update n_k and $n_{w|k}$
  Generate rejuvenation sequence $R(i)$
  for $j \in R(i)$ **do**
    Draw topic assignment k' from $P(z_j|\mathbf{z}_{-j,i}, \mathbf{B}_i)$
    Update $n_{k'}$ and $n_{w|k'}$
Compute $\boldsymbol{\Phi}$ by Eq.(3.5) and $\boldsymbol{\theta}$ by Eq.(3.6)

4.4.2 iBTM（Incremental BTM）算法

和oBTM算法不同，iBTM算法在新数据到来的时候，可立刻对模型参数$\boldsymbol{\Phi}$和$\boldsymbol{\theta}$做出更新，从而即时的捕捉数据的变化。为了保证Gibbs采样的充分性，iBTM采用了一种称为再生Gibbs采样的技术[30]。具体地说，iBTM算法每接收一个新的双词b_i会做两步操作来更新模型。首先，和传统Gibbs采样算法一样，我们会为b_i采样一个话题。其采样条件分布为$P(z_i|\mathbf{z}_{i-1}, \mathbf{B}_i)$，其中$\mathbf{z}_{i-1} = \{z_j\}_{j=1}^{i-1}$表示之前所有的双词的话题赋值，$\mathbf{B}_i = \{b_j\}_{j=1}^{i}$表示到目前为止接收到的双词集合。然后，我们随机地从之前的双词中抽取一小部分，构成一个再生双词序列$R(i)$。对于该序列中的每个双词b_j，我们从条件概率分布$P(z_j|\mathbf{z}_{-j,i}, \mathbf{B}_i)$中重新采样它的话题$z_j$，以修正之前由于数据不充分导致的采样偏差。图4.2展示了iBTM算法对短文本数据流处理流程的一个示意图，详细过程请参见算法4。

在iBTM算法中，如何产生再生双词序列$R(i)$是一个重要问题。首先，到底选择多少个双词来重采样对iBTM算法的效果和性能有直接的影响。$R(i)$的双词个数越多，则采样更充分，从而对后验概率$P(\mathbf{z}_i|\mathbf{B}_i)$估计也就越准确。特别地，如果$R(i) = \mathbf{B}_i$，随着

双词数目趋于无穷，那么每个双词的话题都会被采样无限多次，因此iBTM算法等价于批处理BTM算法。但另一方面，$R(i)$的双词个数越多，每次处理一个新双词的计算量也会相应的增加。在双词数目非常多的情况下，计算时间的增加非常明显。另外，按何种策略来选择$R(i)$中的元素也影响着iBTM算法的效果，因为这决定了不同双词对模型更新的贡献程度，会导致话题学习结果偏向于那些重采样次数多的双词中的数据分布。例如，如果我们从一个概率按时间衰减的分布（如指数分布或逆多项式分布[99]）中来从原双词序列中采样$R(i)$，学到的模型便会偏向于反映近期的数据分布。然而，考虑到随时间的增加，这些分布的参数也会发生相应的变化，会增加额外的计算量。为尽量简化计算，这里我们采用一个固定大小的滑动窗口内的均匀分布来采样$R(i)$中的元素。假设窗口大小为L，更明确地说，我们会存储最近的L个双词作为候选序列，在一个新的双词带来的时候，我们先从这L个双词中按同等概率随机抽取$|R(i)|$个双词并进行重采样。重采样完之后，我们会把新双词加入到候选序列，同时移除最旧的那个双词。注意，虽然我们是按均匀分布采样，但是随着滑动窗口的移动，距离现在时间比较久的双词会被选择进行重采样的概率会越来越低。因此，该方法同样能使模型倾向于反映最近的数据中的话题分布。同时，由于滑动窗口大小是固定的，我们只需要存储L个以前的双词，通常L远远小于总的双词数目。我们可以通过调整L的大小来有效控制内存消耗。

4.4.3　复杂度分析

经过前面的分析，我们可以看到，oBTM算法和iBTM算法相比批处理BTM算法的主要优势在于具有较低的时空开销，因此能适应大规模数据的短文本话题学习。为了更明确这一点，我们现在来分析和对比这三种算法的时间和空间复杂度。为了方便比较，假设原数据集中总共包含T个时间片，我们来估算在第t个时间片，这三种算法对模型更新所需的时间和内存。

在批处理BTM算法中，对模型的更新需要对所有已观测到的数据重新计算，即$\mathbf{B}^{(1..t)} = \mathbf{B}^{(1)} \cup ... \cup \mathbf{B}^{(t)}$。由于对其中每个双词进行Gibbs采样的时间复杂度是$O(K)$，该过程的总时间复杂度为$O(N_{iter}K|\mathbf{B}^{(1..t)}|)$，这里$|\cdot|$表示一个集合中的元素个数。为了比较内存消耗，我们估算算法执行时需要存储的主要变量个数。对于批处理算法，主要包含两部分变量：一是所有的双词，二是Gibbs采样用到的计数器n_k和$n_{w|k}$，所以需在内存维护的总变量个数为$K + WK + |\mathbf{B}^{(1..t)}|$。在oBTM算法当中，我们仅需要对当前时间片内的数据进行运算，因此时间复杂度为$O(N_{iter}K|\mathbf{B}^{(t)}|)$。相应地，需维护的变量个数为$K + WK + |\mathbf{B}^{(t)}|$。iBTM算法需对$\mathbf{B}^{(t)}$中的双词逐个遍历并进行采样。虽然它不需要像iBTM与批处理BTM算法一样反复迭代，但每次对新双词采样的时候需要同时对$R(i)$中的双词进行重采样。容易估算出iBTM的时间复杂度为$O(K|\mathbf{B}^{(t)}| \cdot |R(i)|)$。另外，iBTM中我们不需要存储$\mathbf{B}^{(t)}$，而是存储当前最近的$L$的双词，因此需维护的变量个数为$K + WK + L$。注意通常$L \ll |\mathbf{B}^{(1..t)}|$。

表 4.1: 批处理BTM、oBTM和iBTM算法在第t个时间片时更新模型所需的时间复杂度以及需在内存中维护的变量个数

算法	时间复杂度	变量个数
batch BTM	$O(N_{iter}K\|\mathbf{B}^{(1..t)}\|)$	$K+WK+\|\mathbf{B}^{(1..t)}\|$
oBTM	$O(N_{iter}K\|\mathbf{B}^{(t)}\|)$	$K+WK+\|\mathbf{B}^{(t)}\|$
iBTM	$O(K\|\mathbf{B}^{(t)}\|\cdot\|R(i)\|)$	$K+WK+L$

表4.1对批处理BTM，oBTM和iBTM三种算法的时间复杂度和内存消耗进行了总结。我们可以看到，批处理BTM算法的时间和内存消耗会随着时间片t的增长而线性增加。当t很大的时候，很容易超出目前计算机计算能力的限制。相反，oBTM算法和iBTM算法的时间和空间开销基本上不会随t的增长而增长。实际应用当中，如果每个时间片内的数据通常变化不大，oBTM算法的时间和空间消耗基本上变化不大。iBTM算法由于采用了固定长度的候选序列，其更新模型所需的时间和空间开销也是基本维持在常数级别。但需要指出的是，我们可以通过实际应用需求以及计算能力的强弱来设置oBTM算法中时间片的大小，以及iBTM算法中候选序列的长度L。

4.5 实验结果与分析

本节中，我们通过实验来验证oBTM和iBTM在大规模流式短文本话题学习上的效果与性能，包括话题质量评价，文档中话题比例的评价，以及效率评价。首先我们介绍实验中使用的数据集和基准方法。然后，我们对实验结果进行了展示和分析。

4.5.1 实验数据

因为我们关注的是针对大规模流式下在线话题学习的效果，我们采用了两个较大的微博数据集。

- **Tweets2011数据** 该数据来源于TREC 2011微博任务[1]，是目前比较常用的短文本数据集。该数据集包含了在2011年1月23日-2011年2月8日期间采集的tweets，总共约包含1600万条。每条tweet除了其内容之外，还包含其作者、发布时间等信息。

- **Weibo数据** 我们从新浪微博爬取了2011年8月1日-2012年7月31日期间的部分微博。原始数据大小约600G，要比Tweets2011数据集大很多。

为了减少数据中的噪音，我们对原始数据做了如下的预处理: (a)过滤非中文且非英文的字符; (b)所有的英文字符都转换成了小写; (c)去除了出现次数小于10的词;

[1]http://trec.nist.gov/data/tweets/

表 4.2: 实验数据集预处理后的统计信息

统计项	Tweets2011	Weibo
文档数目	4,230,578	155,617,473
词汇表大小	98,857	187,994
平均文档长度	5.21	5.87

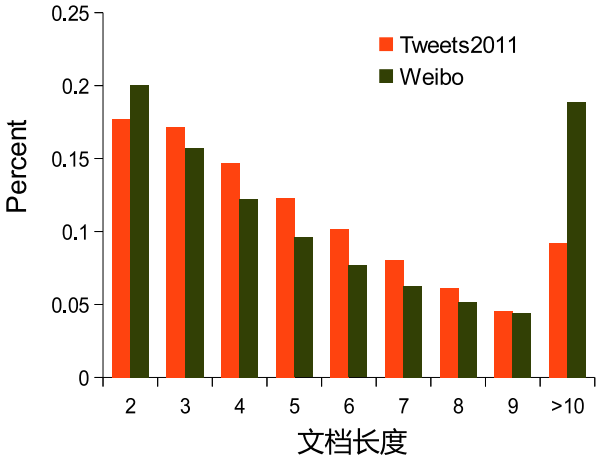


图 4.3: 实验数据集上的文档长度分布

(d)去除了词数少于2的文档；(e)考虑到Tweets和Weibo中转发功能导致很多文档是重复的，我们去除了重复的文档。表4.2中展示了预处理后的各数据集中的文档数、词汇表大小，以及文档的平均长度。图4.3展示了这两个数据集上的文档长度分布。我们可以看到，这些文档的长度非常短，但二者的平均文档长度相差不大。

4.5.2　基准方法

首先，我们本章中提出的两种在线BTM学习算法与批处理BTM算法进行了对比，目的在于判断在学习方式与批处理方式在效果和性能上的差异。此外，由于目前针对大规模流式数据上话题学习方法比较普遍的是用在线LDA算法，我们也和它进行了对比。考虑到在线LDA算法有很多种不同的实现方式，效果有所差异。我们先在初步实验中对比了各种在线LDA算法，如在线LDA[4]，增量LDA（iLDA）[30]，基于粒子滤波的在线LDA算法[30]和在线变分推断算法[70]等。我们发现其中iLDA的效果相对来说比较好。因此，我们选择了用iLDA算法进行对比。为了公平起见，所有的方法都采用C++程序实现[2]。所有的实验都在一台Intel Xeon 2.33 GHz CPU、16G内存的linux服务器上进行。

BTM的参数设置还是和上一章的一样，即$\alpha = 50/K$和$\beta = 0.01$。oBTM中每个

[2]Code of BTM : http://code.google.com/p/btm/

表 4.3: 批处理BTM、oBTM、iBTM和iLDA算法的PMI-Scores对比（值越大越好）

	K	50			100		
数据集	算法	Top5	Top10	Top20	Top5	Top10	Top20
Tweets2011	BTM	**2.74 ± 0.04**	**2.26 ± 0.04**	**1.86 ± 0.02**	**2.88 ± 0.02**	**2.33 ± 0.04**	**1.87 ± 0.03**
	iLDA	2.53 ± 0.04	2.00 ± 0.05	1.59 ± 0.03	2.50 ± 0.04	1.97 ± 0.05	1.55 ± 0.02
	oBTM	2.63 ± 0.03	2.13 ± 0.03	1.80 ± 0.02	**2.72 ± 0.03**	2.16 ± 0.03	1.80 ± 0.02
	iBTM	**2.72 ± 0.03**	**2.17 ± 0.02**	**1.83 ± 0.02**	2.71 ± 0.03	**2.18 ± 0.02**	**1.83 ± 0.04**
Weibo	iLDA	2.37 ± 0.05	1.95 ± 0.02	1.69 ± 0.02	2.43 ± 0.05	1.90 ± 0.02	1.70 ± 0.03
	oBTM	**2.49 ± 0.04**	**2.02 ± 0.02**	**1.75 ± 0.04**	2.50 ± 0.03	**1.95 ± 0.02**	1.74 ± 0.04
	iBTM	2.48 ± 0.05	2.01 ± 0.02	1.74 ± 0.03	**2.54 ± 0.04**	**1.95 ± 0.02**	**1.75 ± 0.05**

时间片设成一天。在和批处理BTM对比的时候，为了排除时间效应的影响，我们将λ设成1，即历史信息不会衰减。同时iBTM中每次重采样的双词数据（即$|R(i)|$）设置成N_{iter}，即oBTM和批处理BTM算法中Gibb采样的迭代次数。同时滑动窗口大小L设置成$|\mathbf{B}^{(1)}|$，目的是为了使iBTM算法和oBTM算法的运行时间和占用内存尽可能一致，以方便二者之间的对比。考虑到两个数据集都比较大，我们将N_{iter}设置成100。

4.5.3 话题质量评价

首先，我们来检验这两种在线话题学习算法学习到的话题质量。这里，我们还是用PMI-score[111]来评价。PMI-score通过借用大规模外部数据来计算一个话题中概率最大的前几个词相互之间的平均PMI（Pointwise Mutual Information）。PMI越高，说明这两个词越相关。因此如果一个话题的PMI-score越高，说明该话题的可解释性越好。

假设话题k中概率最大的T个词为$(w_1, ..., w_T)$，PMI-Score的定义如下:

$$\text{PMI-Score}(k) = \frac{1}{T(T-1)} \sum_{1 \le i < j \le T} \text{PMI}(w_i, w_j)$$

其中$\text{PMI}(w_i, w_j) = \log \frac{P(w_i, w_j)}{P(w_i)P(w_j)}$，$P(w_i, w_j)$和$P(w_i)$分别是双词$(w_i, w_j)$和词$w_i$在外部数据中出现的概率。这里，在Weibo数据上的评价，我们采用的外部数据是500万篇百度百科数据；在Tweets2011上的评价，我们采用的是4百万篇Wikipedia数据。

各对比算法在两个实验数据集上的平均PMI-score如表4.3所示。T分别取5，10和20。注意，由于Weibo数据非常大，我们没有在其上面运行批处理BTM算法。从Tweets2011数据上的结果上看，批处理算法的PMI-score是最优的，因为其对样本的采样最为充分。但对比三种在线话题学习算法，我们发现oBTM和iBTM的效果非常接近，而且和批处理BTM相差不大，但要明显好于iLDA。这说明oBTM和iBTM算法都能有效地学习短文本上的话题。

下面我们通过两个示例来展示在线话题学习的效果。这里我们用iBTM为例。图4.4展示了iBTM算法在Twitter2011数据上学到的一个话题随时间的变化情况（K =

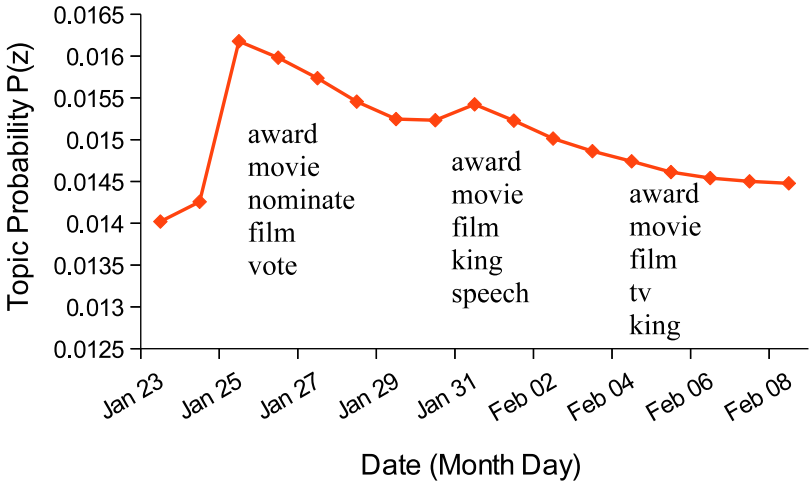


图 4.4: iBTM在Tweets2011数据上学到的一个话题演化示例

50)。图中曲线表示了话题的概率$P(z)$，代表了话题在数据中的流行程度。此外，曲线下方的文字是该话题在不同时间节点上的概率最大5个词。由于时间轴较短，我们仅针对三个具有代表性的时间节点展示了该话题的内容。通过曲线下方的文字，我们容易得知该话题主要关于2011年奥斯卡奖（Oscars Awards，又称Academy Awards)。通过分析该话题的概率变化，我们发现在2011年1月25日的时候，该话题的流行度突然增加。事实上，奥斯卡组委会在当天宣布了提名名单。这也正好和从曲线下方的话题内容（如“nominate”和”vote”）相拟合。在随后的几天，该话题的流行度逐渐从峰值开始下降。但在2011年1月31日的时候，又有一个小波峰出现，并且“king”和“speech”在该话题中的概率增大。这是因为当天一部提名电影“The King's Speech”获得了美国演员工会奖（Screen Actors Guild Awards)，该奖历年来与奥斯卡表演类奖项吻合度很高，被视为是奥斯卡最重要的前哨站之一。因此电影“The King's Speech”被视为夺奖大热门，在随后的几天中依然在该话题中比重较大。图4.5展示了另一个在Weibo数据上的一个例子（$K=100$)。由于Weibo数据包含了一整年的微博，这里我们按月来展示话题的变化。从内容上看，该话题谈论的是关于大学教育。我们可以看到，在2011年9月的时候该话题的流行度达到一个峰值。这是因为9月份正值大学新生入学时期，和入学相关的微博比较多。这一点也可以从该话题的内容上反映出来。在随后的月份中，“入学”等关键词在该话题中不再明显，替代的是“教育”和“同学”等词。到了2012年6-7月份的时候，该话题的流行度又出现一个波峰。这是因为此时接近期末，各种考试相关的微博比较多。我们也可以看到该话题此时也出现“考试”等词。这两个例子都说明，在线话题学习算法能有效的学习到有意义的话题，并能学习到话题随时间的演化，这对微博舆情和情报分析等应用有重要的意义。

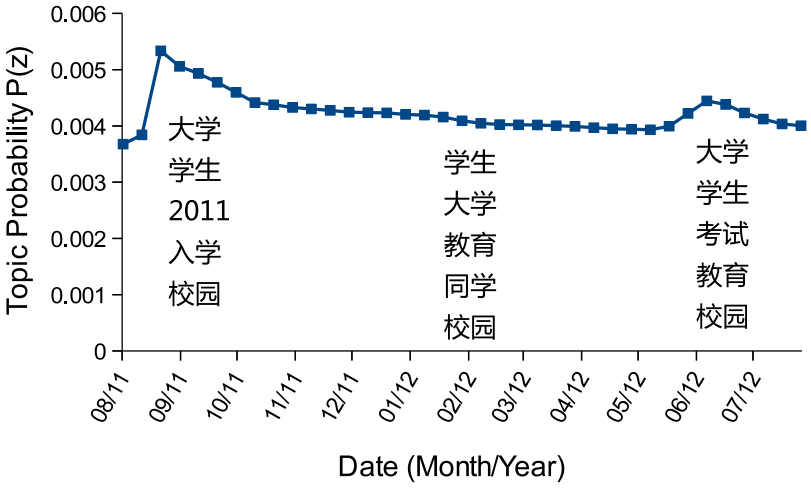


图 4.5: iBTM在Weibo数据上学到的一个话题演化示例

4.5.4 文档中话题比例的评价

我们进一步来检验在线话题学习方法对文档中话题比例推断的质量。我们仍然采用文本分类来评价。我们把话题学习看成是一种降维方法，即把文档从词空间降维到概率话题空间。降维后损失的类别区分信息越少，我们认为文档中的话题比例学习的越好。具体步骤如下：首先，我们将每个文档d表示成一个向量$[P(z=1|d),...,P(z=K|d)]$。然后我们随机地将文档集合按4 : 1的比例进行划分训练集和测试集。我们用线性SVM分类器LIBLINEAR[3]来进行分类。

注意在这两个数据集上文档没有预定义的类别标签。由于两种微博信息都非常短而且通常书写不正规，手工标注极为困难。我们依然利用微博数据中用户标注信息，即hashtag作为类别标签。为了保证文档的类别定义良好，我们分别从两个数据中的手工挑选了50个和事件或话题非常相关的高频hashtag作为类别。然后把包含这些hashtag的文档用来做分类实验。注意，如果一个文档包含其中多个hashtag，则忽略该文档。其中，Twitter2011数据中挑选的50个hashtag和上一章的实验一致，参见表3.6。微博数据中的50个hashtag如表4.4所示。

图4.6和图4.7分别展示了批处理BTM、oBTM、iBTM和iLDA在Tweets2011和Weibo数据上的分类实验结果。其中，在Tweets2011数据上话题个数设置为50；考虑到Weibo数据规模比Tweets2011大很多，其话题个数设置为100。我们也实验过其他不同的话题个数设置，结果类似。由于批处理BTM需要对当前时刻t之前所有数据都拿来训练，在大规模数据上，当t很大时运算代价非常高。所以在Weibo 数据上，批处理BTM算法只记录其到80天为止的结果。

从这两幅图中，我们可以观察到以下几点。总的来说，oBTM和iBTM的分类实验效果和批处理BTM非常接近，而且总是明显优于iLDA。注意，在流式数据场景下，

[3]http://www.csie.ntu.edu.tw/~cjlin/liblinear/

表 4.4: Weibo数据中选择用来分类评价的50个Hashtags

Android	BIGBANG	IT新闻	NBA	YOKA时尚	中国好声音
健康	军事新闻	北京爱情故事	北京路况	古剑奇谭	吸血鬼日记
大武生	奥运加油	娱乐新闻	婚纱	寻龙记	小智慧
屋塔房王世子	微博派福	微招聘	德克萨斯扑克	心灵旅程	情人节
摄影	星座	春晚	暴走漫画	最接近天堂的地方	欧洲杯
步步惊心	汽车新闻	汽车知道	海贼王	温州动车追尾	游戏前沿
游戏美女	火影忍者	爱搭配精选	父亲节	社会新闻	科技新闻
美女车模	美食	薄荷	财经新闻	酷车鉴赏	银魂
高考	乔布斯去世				

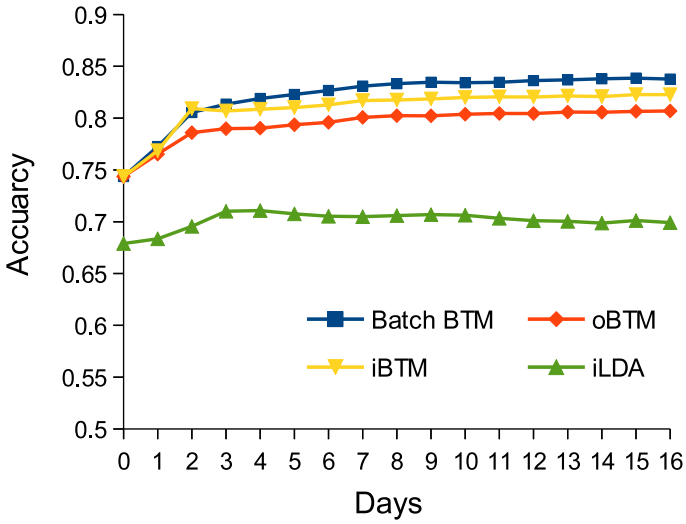


图 4.6: 批处理BTM、oBTM、iBTM和iLDA在Tweets2011数据上的分类实验结果

随着时间的增加，模型训练的数据也在增加。在开始阶段，我们发现批处理BTM、oBTM和iBTM算法的实验效果也会逐渐增加。增加到一定程度后，它们效果会逐渐趋于平稳。但是，iLDA的结果却不一样。虽然在Tweets2011数据上，iLDA的分类精度初始阶段有一定的上升，但其余时候基本上随着接收数据的增加，反而下降。类似地现象也在论文[30]中有提到。这说明oBTM和iBTM的稳定性也要比iLDA好的多。另外，对比oBTM和iBTM，我们发现iBTM一直都超过oBTM，即使优势不是特别大。可能地原因是iBTM是完全按数据接收顺序来学习话题的，能更好的保持数据之间的时序相关性。反之，oBTM按时间片硬性划分会导致某些时间上本来临近的数据被分配到不同的时间片中，从而损坏了数据内部的时序相关性。

4.5.5　效率比较

相比批处理算法，在线话题学习算法的另一个关键优势就是在大规模数据上的可伸缩性。为了验证这一点，我们以Tweets2011数据为例，对比了这几种算法的时间和

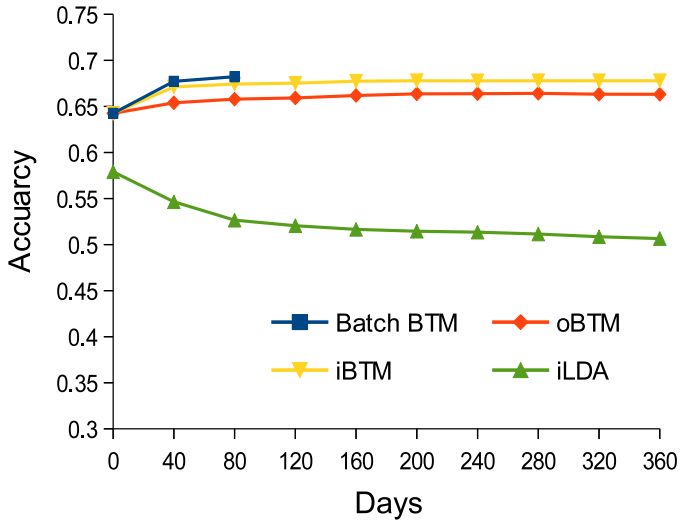


图 4.7: 批处理BTM、oBTM、iBTM和iLDA算法在Weibo数据上的分类实验结果

内存消耗，见图4.8。我们可以看到随时间的增加，三种在线算法的时间和内存消耗基本保持不变，而批处理BTM算法的时间和内存消耗都是呈线性增长。这也和前面的复杂度分析（表4.1）的一致。显然，在线算法要比批处理算法更适合大规模数据的处理，特别是在效果差不多的情况下。对比iLDA，oBTM和iBTM的时间消耗要稍微多些，但内存消耗相对要低些。虽然从效率上讲，oBTM、iBTM与iLDA相当，但如前面的实验结果所述，oBTM和iBTM的效果要比iLDA更好，而且更稳定。

4.6 小结

如何将短文本话题建模方法应用到互联网上的一些在线应用（如微博）是一个很实际的问题。这些在线应用产生的数据具有很强的动态性，即数据规模不断增长，而且内容也在不断变化。这就要求我们不仅仅要提高学习算法的效率，还能及时地学习话题的变化。传统的批处理算法无法满足这种应用需求。为此，我们在双词话题模型（BTM）提出了两种在线话题学习方法：oBTM（online BTM）和iBTM （incremental BTM）。这两种算法通过存储和计算一小部分最近的数据来增量的更新模型，从而大大降低了模型更新所需的时间和内存。同时，它们还能追踪话题的演化。

我们通过在Tweets2011和Weibo等大规模短文本数据上，验证了这两个在线话题学习算法的效果与性能。oBTM和iBTM能在非常少的效果损失的下，极大地提高原BTM算法的更新效率，从而适应不断增长的数据规模。通过和在线LDA算法的对比，我们也证明了oBTM和oBTM在话题质量和文档中话题比例的学习上都要明显优于在线LDA算法。因此，本文的工作对在线话题建模提供了一种有力工具，并进一步提高双词话题模型的实用价值。

本章的研究成果已被国际期刊IEEE Transactions on Knowledge and Data Engi-

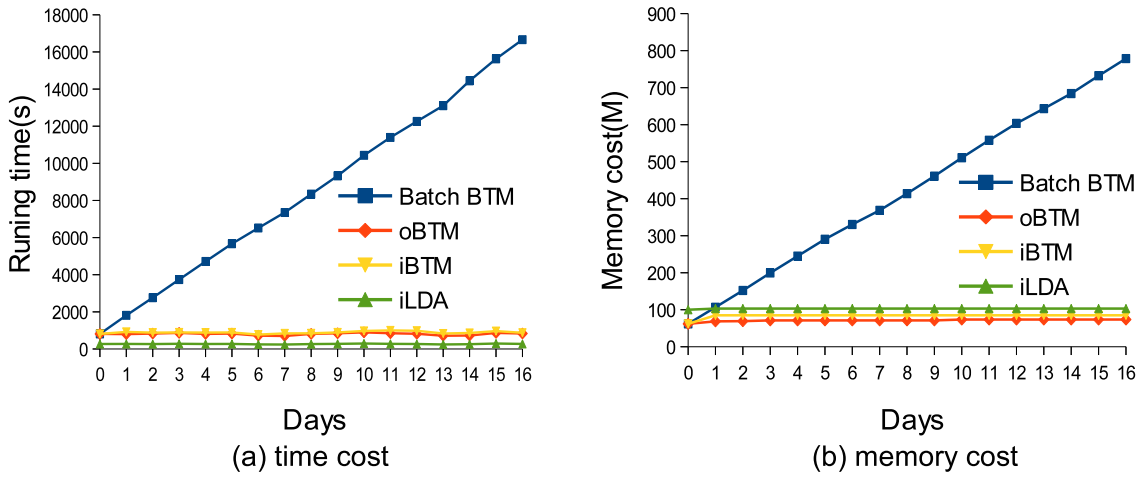


图 4.8: 批处理BTM、oBTM、iBTM和iLDA算法在Tweets2011数据上的时间和内存消耗

neering（TKDE）接收，论文题目为“BTM: Topic Modeling over Short Texts”。

第五章　突发话题建模

5.1　引言

在微博等流式短文本中，除了话题会动态演化之外，每天还会有很多突发话题涌现。这些突发话题通常和一些受一些事件或活动所激发，或者是一些热门的讨论，在短时间内吸引很多人的关注。发现这些突发话题对舆情监控等应用有重要意义。然而，普通的话题模型并没有考虑话题的时序特性，因此不能自动学习突发话题和普通话题。

本章中，我们在双词话题模型的基础提出了一个突发话题模型，即组合双词话题模型（Composite Biterm Topic Model或CBTM）。CBTM对突发话题和普通话题进行区分建模，然后利用双词的突发性来指导不同类型的话题学习。其主要思想是：一个突发性强的双词更可能由一个突发话题所产生；相反，一个突发性弱的双词更可能由一个普通话题产生。借助双词的突发性特征，组合双词话题模型可以自动学习出短文本数据中潜在的突发话题。通过Twitter数据上的实验表明，该方法能很好的区分普通话题和突发话题。

5.2　概述

随着社交媒体的迅速发展，微博已成为非常流行的大众交流平台。根据官方最新统计数据，Twitter每月的活跃用户已超过2.4亿，平均日发帖量达5千8百万，而新浪微博每月的活跃用户超过1.4亿，平均日发帖量约1亿条消息。这些消息包含各种各样的话题，如日常生活，聊天对话，娱乐信息，新闻事件，商业活动等。其中，有些话题会在突然之间吸引很多人讨论，比如一些新闻事件，线上活动，热点话题等，我们称这些话题为突发话题。发现微博中的突发话题，对于很多应用具有重要意义，如舆情监控、新闻线索收集、商业情报分析、消息推荐等。但另外，微博中也有很多话题内容随时间变化不大，如日常生活，星座，幽默笑话等，我们称之为普通话题。突发话题和普通话题混杂在一块，给突发话题发现带来了很大的困难。

突发话题是一类特殊的话题，即其内容具备突发性。一般的话题学习方法以完全无监督的方式学习话题，在没有任何先验知识或约束的情况下，无法有效的区分突发话题和普通话题。例如，概率话题模型通常以最大化观察数据的似然函数为优化目标。于是，文档集合中越是流行的话题，越容易被发现。然而，在微博数据中，普通话题往往比突发话题要更为流行。根据Pear Analytics 2009 年的调查报告[5]（表5.1），有接近80%的消息属于无意义地喃喃自语或者对话，这部分内容通常随时间变化不大。而新闻等可能包含突发话题的内容只占3.6%。可以预见，一般的话题模型学习到的话题中

表 5.1: Twitter中各类消息比例，数据来源于Pear Analytics

消息类型	比例
无意义地喃喃自语	40.55%
对话（含“@”）	37.55%
转发消息	8.7%
自动推广	5.75%
垃圾消息	3.75%
新闻	3.6%

大部分都是普通话题，难以发现那些关注度较小的突发话题。即使增加话题数目，通常也只会导致更多的普通话题出现，并不能从根本上解决问题。

为了区分突发话题和普通话题，一种简单地做法是对话题学习结果进行后处理。Becker等人[14]通过人工标注部分话题来训练一个分类器来判断突发话题和普通话题。Lau等人[86]通过比较当前话题与之前对应话题的内容差异度，若大于指定阈值，则认为是突发话题。虽然后处理的方式可以区分出突发话题和普通话题，但不能主动发现那些不那么流行的突发话题。Diao等人[53]认为和时间相关话题更可能是突发话题，而和用户相关的话题更可能是普通话题。基于此假设，他们将用户和时间分别建模成一个话题分布，试图利用话题在时间和用户层面上的局部性来区分突发话题和普通话题。然而这种区分仅仅是在话题分布上的区分，并没有在话题作区分。他们的方法最终仍然依赖于后处理的方式来检测突发话题。进一步地，Yin等人[163]将话题分为稳态话题和时态话题两种不同类型。并在话题学习过程中，采用了一种启发式的做法增强突发词在时态话题中的概率，从而使时态话题更具突发性。

为了更好地建模短文本数据流中突发话题，我们在双词话题模型的基础上，我们提出了一个突发话题模型，即双词组合话题模型（CBTM，Composite Biterm Model）。CBTM利用双词的突发性来指导突发话题的学习，无需任何后处理手段与启发式的技巧。具体地说，对于每个双词，我们可以通过统计其出现次数的变化来估计其突发性强弱。一个突发性强的双词，更可能是由突发话题产生；而一个突发性较弱的词，更可能是有普通话题产生。注意，我们这里强调“可能”，因为并不是突发性强的词一定就会是由突发话题产生，也可能是因为垃圾信息等。同样的，一个突发性弱的双词，也还是有可能由突发话题所产生，只不过它在突发话题中作用并不显著而已。

根据以上的直观假设，在CBTM中我们定义了两类话题：突发话题和普通话题。一个双词可能是由其中的某类话题中的一个所产生。由于我们无法直接观察到双词到来自哪类话题，这里用一个二值的隐变量来表示一个双词所属话题类别。该隐变量的值来自于一个Bernoulli分布，它决定了双词话题类别偏好。我们基于双词的时序次数信息估计出其对应的伯努利分布参数，以此来指导突发话题和普通话题的学习。

我们在Twitter数据集上验证了CBTM的效果。实验结果表明，CBTM模型发现的突发话题无论从数量上，还是可读性上都显著优于目前的方法。此外，还能准确的发现和突发话题相关的短文本消息。

本章其余内容组织如下：5.3节介绍了相关工作；5.4节详细介绍了我们的突发话题建模方法；5.5节介绍了其参数估计方法；5.6节给出了实验结果和讨论；5.7节对本章工作进行了小结。

5.3　相关工作

我们分两方面来介绍相关工作：微博中的话题学习方法和突发话题检测。

5.3.1　微博中的话题学习方法

微博中一条消息的长度限制为140个字符。如此短的文档长度给传统的话题学习方法带来严重的数据稀疏性问题[73, 159, 169]。由于之前缺乏针对短文本的话题学习方法，一些工作中只是简单地采用的传统话题模型，如LDA[86, 151]。这种方式通常得到的话题质量并不高。为了克服数据信息性问题，一些研究者把一些相关的消息，比如同一个用户发的消息或包含同一个hashtag的消息，聚合成一个虚拟的长文档[73, 101, 155]，然后再学习话题。经验结果表明该方式比直接应用LDA来学微博的话题效果要好。还有一种常见的做法是假设每条微博只包含一个话题[53, 169]。但该方式不能刻画一条微博中含有多个话题的情况。最近，Yan等人[159]提出了双词话题模型来解决短文本话题学习，即通过建模双词的产生来学习话题。该方式利用全局丰富的词共现信息来解决短文本内部词共现不足对话题学习的影响。本章中，我们采用双词话题模型来更好的学习微博上的话题。

5.3.2　话题检测

话题检测是从文本流中识别话题的一项技术。对于其的研究可以追溯到1997年DARPA等机构组织的话题检测与追踪项目（TDT，Topic Detection and Tracking）[1]。该项目的初衷是从新闻流中检测和追踪事件，所以通常话题检测也成为事件检测[2]。目前主要的话题检测方法可以分为三种：基于文档的方法[3, 162]、基于特征的方法[56]和基于话题学习的方法[4]。下面我们分别介绍这几种方法在微博突发话题中的应用。

5.3.2.1　基于文档的微博突发话题检测方法

其中，基于文档的方法的做法是对文档聚类，然后把每个类视为一个话题。该方法在面向新闻流的话题发现中应用较多。但和新闻数据不同，微博数据中和新闻事件的相关消息只占很小一部分，大部分微博消息都是一些无意义地闲聊等内容。因此，

[1]http://www.itl.nist.gov/iad/mig//tests/tdt/

对微博消息聚类得到的大多上都是和突发话题无关的内容。一种改进的方式就是对聚类结果再进行分类[14, 44]，但该方式需要付出高昂地人工标注代价。而且，作为一种后处理方式，分类虽然能区分出突发内容与普通内容，但并不能改进对突发内容的识别。因此，基于文档的微博突发话题检测方法在实际中用的较少。

5.3.2.2 基于特征的微博突发话题检测方法

基于特征的方法一般首先从文本流中抽取一些突发性强的特征，如词或短语，然后对这些特征聚类得到话题。最后得到的话题就是一些突发词的集合。Michael和Nick[100]设计了TwitterMonitor系统来检测Twitter数据流中的趋势（trend）。该系统首先用一种排队算法来抽取突发词，然后根据其共现关系聚类得到趋势，并通过抽取一些下文信息，如命名实体，来更准确地表达一个趋势。Cataldi等人[33]提出了一种基于衰老理论（aging theory）的词突发性计算方式，然后根据词的共现关系对这些突发性强的词连边构建一个图。最后，找出图中的强连通子图作为突发话题。Weng等人应用小波变化与自相关分析来检测突发词，然后基于模块度（modularity）的图划分的方法对突发词聚类，每个类最后视为一个事件[156]。Li等人[88]提出了一个基于消息片段的微博事件检测系统Twevent。首先，Twevent借助Wikipedia将每条消息切分成多个短语。然后选择突发性强的短语基于共现关系来聚类，得到候选事件。最后，再次借助Wikipedia中的事件相关内容来对候选事件过滤，得到最终的事件结果。

在基于特征的方法中，突发词的抽取一般仅仅需要用到词的统计信息，计算比较简单。而且突发词的比例一般也比较小，其后对突发词聚类的效率也较高，比较适合在线突发话题检测。但其也存在很多不足：1）仅仅抽取突发词，会损失了很多上下文信息，可能导致聚类效果不理想，对话题的表达也不全面。2）为了提高效果，通常还需要很多的额外步骤来进一步精化结果，如对话题排序[33]，或者利用Wikipedia来过滤[88]等。这些额外的步骤大大增加了计算负担。3）通常每一步都包含较多的启发式技巧以及自由参数，而且不同地工作中都采用不同的做法。这给我们在方法和参数的选择上，带来极大地困难。

5.3.2.3 基于话题学习的微博突发话题检测方法

用话题学习的方法来发现突发话题是一种新的尝试。在话题学习方法中，话题通常用一个词的分布[22, 72]或者词空间的一个向量[161]来表示。这种表达方式比用突发词集合来表示更全面。Kasiviswanathan等人[83]采用了字典学习的方式来发现突发话题。字典学习通过矩阵分解来学习文档在一个潜在语义空间上的表达，和话题学习方法非常相关。Kasiviswanathan等人的做法分为两步：首先，识别文档流中与当前字典匹配程度较差的一些文档。然后，对于这些文档再通过字典学习的方式聚类得到突发话题。在此基础上，Saha等人[123]提出了在线NMF来简化两步走的做法。他们将话题分为演化话题和突发话题两类。在每个时间段内，通过增加话题的数目来学习突发话

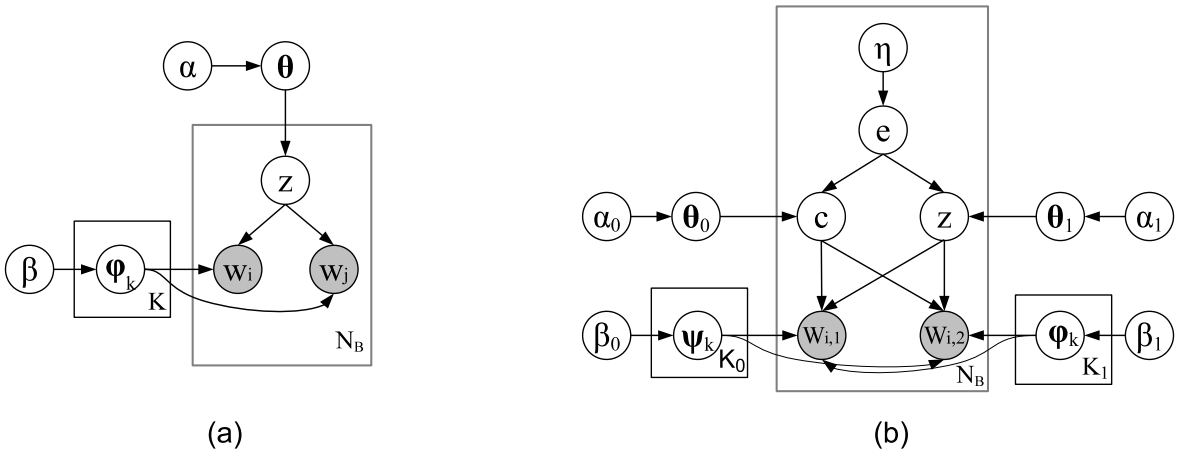


图 5.1: （a）BTM和（b）CBTM的概率图模型表示

题。新产生的突发话题会随着时间的推进，转变成演化话题。同时，演化话题用时序正则化方法来限制其内容随时间而剧烈变化。该方法最大的问题是话题数目会一直随时间而无限增长。受[4]的启发，Lau等人[86]应用在线LDA算法来检测微博中的趋势（trend）。首先，将微博数据流按时间划分成多个时间片，然后学习每个时间片内的话题分布。如果一个话题的词分布于之前时间片中对应的话题的分布差别大于一个阈值，则认为是一个突发话题。由于普通话题学习方法在微博上学到的大部分都为普通话题，这种方法检测效率不高。Diao等人[53]提出了一个TimeUserLDA模型将每个时间片和用户分别建模成一个话题分布，试图利用话题在时间上和用户上的局部性来区分突发话题与普通话题，即和时间相关话题更可能是突发话题，而和用户相关的话题更可能是普通话题。然而这种区分仅仅是在话题分布上的区分，并没有在话题层面上作区分。TimeUserLDA仍然依赖于后处理的方式来检测突发话题。进一步地，Yin等人[163]基于PLSA提出了一个User-Temporal模型将话题分为稳态话题和时态话题两种不同类型。其中，稳态话题由用户所产生，其热度假定比较稳定；而时态话题与时间很相关，其热度在当前时间要平常高很多。为了让时态话题中突发词的概率较大，作者在每几轮EM迭代之后，将时态话题中的突发词的概率乘以该突发词的突发度（大于1）。通过强调突发词在话题中的作用，User-Termporal模型可以自动区分出突发话题，但该方式只是一种启发式的手段。另外，值得注意的是，[53] 和[163]都是回顾式的话题检测（retrospective topic detection），需要对整个语料进行批处理检测，这种方法不适合大规模流式数据中的在线突发话题发现。

和以上方法不同，本文提出了一种直接对突发话题建模的方法。该方法能利用双词的突发性来指导突发话题的学习，而能无需任何启发式技术和后处理手段，比之前的方法更简单，更具原则性。

5.4 组合双词话题模型

5.4.1 模型定义

我们知道，突发性是一种时序特征，而BTM中并没有包含任何时序信息，因此无法主动发现突发话题。为此，我们还考虑将双词的突发性质加入到模型当中，以指导突发话题的学习。假定当前时间段为t，此时的双词集合为$\mathbb{B} = \{b_1, ..., b_{N_B}\}$，其中不同的双词个数为$B$。

首先，我们假定语料中包含两种话题类型：突发话题和普通话题（或者非突发话题），分别对应两个不同的话题集合$\{\boldsymbol{\phi}_k | k \in [1, K_1]\}$和$\{\boldsymbol{\psi}_k | k \in [1, K_0]\}$。其中，突发话题的内容在当前时段内急剧增加，而普通话题的内容则是随时间的变化基本保持不变。由于话题是潜在的，我们无法是直接观察到其内容，但我们可以观察到双词的时序特征。在我们的假设中，双词是由话题所产生的，因此双词的突发性与话题的突发性存在着以下直观地联系：

假设 1 一个突发性强的双词（即该双词在该时间段内出现次数急剧增加），更可能是由一个突发话题所产生；相反，一个突发性弱（即该双词在该时间段内出现次数没有显著增加）的双词更可能是由一个普通话题所产生。

注意，我们强调“可能”，因为有些突发性强的双词仍然有可能是有普通话题所产生。我们举Tweet2011数据中的一个例子，在某个周日晚上，正好有橄榄球决赛，因此“Sunday night”受橄榄球决赛这个突发话题影响，其出现次数急剧增多。但是，“Sunday night”仍然同时也在很多普通话题中出现，比如在消息“I cannot sleep on Sunday night”中，它显然是属于普通话题。另一方面，一个突发性弱的双词也还是有可能是由一个突发话题所产生，只不过因为它在突发话题中出现的次数并不多而已。

于是，我们为每个双词b_i定义一个二值隐变量e_i来对其话题类型进行概率建模，其值服从一个参数为η_i的Bernoulli分布（我们将在下一小节中给出其定义）。其中，$e_i = 0$，表示来自普通话题；$e_i = 1$表示来自突发话题。于是，我们现在可以将双词集合为$\mathbb{B} = \{b_1, ..., b_{N_B}\}$的产生过程描述如下：

1. 对整个语料，
   (a) 采样一个普通话题分布: $\boldsymbol{\theta}_0 \sim \mathrm{Dir}(\alpha_0)$
   (b) 采样一个突发话题分布: $\boldsymbol{\theta}_1 \sim \mathrm{Dir}(\alpha_1)$
2. 对每个普通话题$k \in [0, K_0]$ 采样一个词分布: $\boldsymbol{\psi}_k \sim \mathrm{Dir}(\beta_0)$
3. 对每个突发话题$k \in [0, K_1]$ 采样一个词分布: $\boldsymbol{\phi}_k \sim \mathrm{Dir}(\beta_1)$

4. 对每个双词$b_i \in \mathbb{B}$

   (a) 采样一个类别标识$e_i \sim \text{Bern}(\eta_i)$

   (b) 若$e_i = 0$:

      i. 采样一个普通话题: $k \sim \text{Mult}(\boldsymbol{\theta}_0)$

      ii. 独立地采样两个词: $w_{i,1} \sim \text{Mult}(\boldsymbol{\psi}_k)$,$w_{i,2} \sim \text{Mult}(\boldsymbol{\psi}_k)$

   (c) 若$e_i = 1$:

      i. 采样一个突发话题: $k \sim \text{Mult}(\boldsymbol{\theta}_1)$

      ii. 独立地采样两个词: $w_{i,1} \sim \text{Mult}(\boldsymbol{\phi}_k)$,$w_{i,2} \sim \text{Mult}(\boldsymbol{\phi}_k)$

以上过程相当于把所有双词$\mathbb{B}$分成了两部分：$\mathbb{B}_0 = \{b_i | e_i = 0, i \in [1, N_B]\}$和$\mathbb{B}_1 = \{b_i | e_i = 1, i \in [1, N_B]\}$，然后分别用一个BTM来建模，因此我们称该模型为组合双词话题模型（Composite Biterm Topic Model或CBTM）。注意，这里e_i是一个隐变量，其值是不确定的。图5.1给出了BTM与CBTM的概率图模型。

根据以上产生式过程，在给定$\boldsymbol{\Theta} = \{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\Psi}, \boldsymbol{\Phi}\}$和$\boldsymbol{\eta} = \{\eta_i | i \in [1, N_B]\}$的情况下，一个双词$b_i$的产生概率为：

$$\begin{aligned}
P(b_i|\boldsymbol{\Theta}) &= \sum_{k=1}^{K_0} P(w_{i,1}, w_{i,2}, z_i = k, e_i = 0 | \boldsymbol{\theta_1}, \boldsymbol{\Psi}, \eta_i) + \sum_{k=1}^{K_1} P(w_{i,1}, w_{i,2}, z_i = k, e_i = 1 | \boldsymbol{\theta_0}, \boldsymbol{\Phi}, \eta_i) \\
&= \sum_{k=1}^{K_0} P(z_i = k|\theta_{0,k}) P(w_{i,1}|z_i = k, \psi_{k,w_{i,1}}) P(w_{i,2}|z_i = k, \phi_{k,w_{i,2}}) P(e_i = 0|\eta_i) + \\
&\quad \sum_{k=1}^{K_1} P(z_i = k|\theta_{1,k}) P(w_{i,1}|z_i = k, \phi_{k,w_{i,1}}) P(w_{i,2}|z_i = k, \phi_{k,w_{i,2}}) P(e_i = 1|\eta_i) \\
&= \sum_{k=1}^{K_0} \theta_{0,k} \psi_{k,w_{i,1}} \psi_{k,w_{i,2}} (1 - \eta_i) + \sum_{k=1}^{K_1} \theta_{1,k} \phi_{k,w_{i,1}} \phi_{k,w_{i,2}} \eta_i
\end{aligned} \tag{5.1}$$

给定超参$\boldsymbol{\Delta} = \{\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \boldsymbol{\beta}_0, \boldsymbol{\beta}_1\}$，我们可以把参数$\boldsymbol{\Theta}$积掉，得到:

$$P(b_i|\boldsymbol{\Delta}) = \int\int \left(\sum_{k=1}^{K_0} \theta_{0,k} \psi_{k,w_{i,1}} \psi_{k,w_{i,2}} (1 - \eta_i) + \sum_{k=1}^{K_1} \theta_{1,k} \phi_{k,w_{i,1}} \phi_{k,w_{i,2}} \eta_i \right) \mathrm{d}\boldsymbol{\Theta} \tag{5.2}$$

考虑整个双词集合$\mathbb{B}$，其似然函数为：

$$P(\mathbb{B}|\boldsymbol{\Delta}) = \prod_{i=1}^{N_B} \int\int \left(\sum_{k=1}^{K_0} \theta_{0,k} \psi_{k,w_{i,1}} \psi_{k,w_{i,2}} (1 - \eta_i) + \sum_{k=1}^{K_1} \theta_{1,k} \phi_{k,w_{i,1}} \phi_{k,w_{i,2}} \eta_i \right) \mathrm{d}\boldsymbol{\Theta} \tag{5.3}$$

5.4.2 η_i的计算

注意，在上文中虽然对突发话题和普通话题进行分开建模，但是CBTM还不能自动区分突发话题和普通话题。为此，我们需要通过定义e_i的先验分布Bern(η_i)来指导这两类不同话题的学习。

η_i表示了在当前时间段t内双词b_i由突发话题产生的可能性，即$P(e_i=1|b_i)$，我们根据一个双词的时序次数来估计η_i。假设双词b_i在时间段$(1,...,t)$内出现的次数分别为$(n_i^{(1)},...,n_i^{(t)})$。根据我们之前的假设，一个双词的产生有两种可能，要么是由突发话题产生，要么是由普通话题所产生。首先我们不考虑数据中的噪音，在理想状态下，$n_i^{(t)}$可分解成两部分：

$$n_i^{(t)} = n_{i,0}^{(t)} + n_{i,1}^{(t)}, \tag{5.4}$$

其中$n_{i,0}^{(t)}$和$n_{i,1}^{(t)}$分别表示在时间段t内b_i由普通话题和突发话题产生的次数。如果我们已知$n_{i,0}^{(t)}$和$n_{i,1}^{(t)}$，那么η_i的计算就非常简单：

$$\eta_i = \frac{n_{i,1}^{(t)}}{n_i^{(t)}}. \tag{5.5}$$

但是，这里我们并不知道真实的$n_{i,0}^{(t)}$和$n_{i,1}^{(t)}$，下面我们利用b_i的历史次数信息来估计$n_{i,0}^{(t)}$和$n_{i,1}^{(t)}$，从而最终得到η_i的估计。

这里主要用到了普通话题和突发话题的时序特性。首先，由于普通话题的内容随时间变化不大，因此$n_{i,0}^{(t)}$的大小也应该比较稳定。其次，由于突发话题只是在短时间内内容急剧增多。因此忽略噪音的理想状态下，当t较大的时候，在大部分时间段$n_i^{(j)} = n_{i,0}^{(j)}$。于是，我们可以粗略的估计$n_{i,0}^{(t)}$为:

$$\hat{n}_{i,0}^{(t)} = \mu_i, \tag{5.6}$$

这里$\mu_i = \frac{1}{t-1}\sum_{j=1}^{t-1} n_i^{(j)}$是$b_i$的历史平均每时间段出现次数。有了$\hat{n}_{i,0}^{(t)}$之后，再根据式(5.4)，我们相应地估计出在时间段t内突发话题产生b_i的次数：

$$\hat{n}_{i,1}^{(t)} = \max(n_i^{(t)} - \hat{n}_{i,0}^{(t)}, \epsilon), \tag{5.7}$$

其中，$\epsilon \in (0,1)$ 是一个较小的数（本文中我们设为0.01），用做平滑作用，目的是为了保证不突发的双词也会有很小的概率由突发话题产生。

将式（5.6）和式（5.7）代入式（5.5），得到：

$$\eta_i = \frac{\max(n_i^{(t)} - \mu_i, \epsilon)}{n_i^{(t)}}. \tag{5.8}$$

在以上的分析中，我们忽略了双词出现次数的随机性。事实上，即使在普通话题中，双词每个时间段内的出现次数也会有所波动。在式（5.8）中，这种波动性对于高

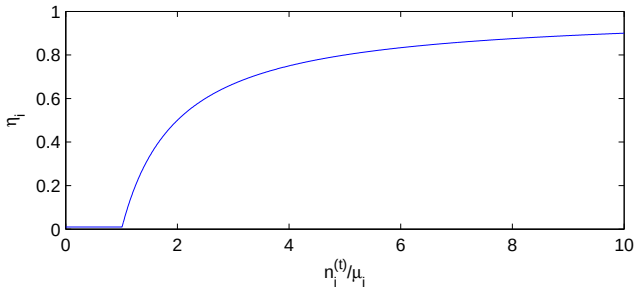


图 5.2: η_i随$n_i^{(t)}/\mu_i$的变化曲线（$n_i^{(t)} > 5$）

频的双词来说影响不大，但对于一些低频双词会导致其η_i过高。这些低频双词由于其频率较低，其实突发性较弱，因此由突发话题产生的概率应该也较小。为此，我们令每个时间段出现次数少于5的双词的$\eta_i = \epsilon$。于是，最终的η_i 计算如下：

$$\eta_i = \begin{cases} \frac{\max(n_i^{(t)} - \mu_i, \epsilon)}{n_i^{(t)}}, & n_i^{(t)} > 5 \\ \epsilon & n_i^{(t)} \leq 5 \end{cases} \tag{5.9}$$

图5.2绘出了$n_i^{(t)} > 5$时η_i随$n_i^{(t)}/\mu_i$的变化曲线。我们可以看到，当$n_i^{(t)}/\mu_i$越大（此时双词b_i的突发性越强），η_i也越大，即b_i由突发话题产生的概率也越大；相反，$n_i^{(t)}/\mu_i$越小（此时双词b_i的突发性越弱），η_i 也越小，即b_i由普通话题产生的概率也越大。这说明式（5.9）的定义很好地符合了假设1。于是，CBTM于是利用η_i来指导不同类型的话题学习过程，在下一小节我们将详述其过程。

5.5　参数估计

CBTM模型的需要估计的参数为$\boldsymbol{\Theta} = \{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\Psi}, \boldsymbol{\Phi}\}$。

5.5.1　Gibbs采样算法

在似然函数式（5.3）中，由于参数之间存在耦合关系，我们无法通过最大似然估计精确求解这两个参数。这里我们借鉴[61]中的collapsed Gibbs采样算法来近似求解。其主要思想是交替地去对待估计地随机变量进行后验采样，其中每次随机变量进行采样基于其他随机变量的赋值。具体地说，在CBTM中我们需要对每个双词b_i采样一个话

Algorithm 5: 针对CBTM的Gibbs采样算法

Input: K_0, K_1, α_0, α_1, β_0, β_1, $\mathbb{B}$
Output: $\boldsymbol{\Psi}, \boldsymbol{\Phi}, \boldsymbol{\theta_0}, \boldsymbol{\theta_1}$
Randomly initialize the topic assignments for all the biterms
for *iter* $= 1$ *to* N_{iter} **do**
  foreach *biterm* $b_i = (w_{i,1}, w_{i,2}) \in \mathbb{B}$ **do**
    Draw e_i, k from Eq.(5.10) and Eq.(5.11)
    Update $n_{e_i,k}$, $n_{e_i,k,w_{i,1}}$, and $n_{e_i,k,w_{i,2}}$
Compute the parameters by Eqs. (5.12-5.15)

题类型标示变量e_i，以及一个话题赋值z_i或c_i。采样的条件概率分布为

$$P(e_i=0, c_i=k|\mathbf{e}_{\neg i}, \mathbf{c}_{\neg i}, \mathbb{B}, \alpha_0, \beta_0, \nu_i, \mu_i) \propto (1-\eta_i) \cdot \frac{(n_{\neg i,0,k}+\alpha_0)}{(n_{\neg i,0,\cdot}+K_0\alpha_0)} \cdot \frac{(n_{\neg i,0,k,w_{i,1}}+\beta_0)(n_{\neg i,0,k,w_{i,2}}+\beta_0)}{(n_{\neg i,0,k,\cdot}+W\beta_0)^2}, \tag{5.10}$$

$$P(e_i=1, z_i=k|\mathbf{e}_{\neg i}, \mathbf{z}_{\neg i}, \mathbb{B}, \alpha_1, \beta_1, \nu_i, \mu_i) \propto \eta_i \cdot \frac{(n_{\neg i,1,k}+\alpha_1)}{(n_{\neg i,1,\cdot}+K_1\alpha_1)} \cdot \frac{(n_{\neg i,1,k,w_{i,1}}+\beta_1)(n_{\neg i,1,k,w_{i,2}}+\beta_1)}{(n_{\neg i,1,k,\cdot}+W\beta_1)^2} \tag{5.11}$$

其中各符号的解释如下：$\neg i$表示不计双词b_i，$\mathbf{c}$和$\mathbf{z}$表示所有双词的普通和突发话题赋值；n_{0,b_i}和n_{1,b_i}分别表示普通话题和突发话题产生双词b_i的次数；n_{b_i}表示双词b_i在$\mathbb{B}$中出现的总次数；$n_{0,k}$和$n_{1,k}$分别表示普通话题和突发话题中第k个话题产生的双词个数；$n_{0,\cdot}$和$n_{1,\cdot}$分别表示普通话题和突发话题产生双词的个数，即$n_{0,\cdot} = \sum_{k=1}^{K_0} n_{0,k}$和$n_{1,\cdot} = \sum_{i=1}^{K_1} n_{1,k}$；$n_{0,k,w}$和$n_{1,k,w}$分别表示普通话题和突发话题中第k个话题产生词$w$的次数；$n_{0,k,\cdot}$和$n_{1,k,\cdot}$分别表示普通话题和突发话题中第k个话题产生词的个数，即$n_{0,k,\cdot} = \sum_{i=1}^{W} n_{0,k,w_i}$和$n_{1,k,\cdot} = \sum_{i=1}^{W} n_{1,k,w_i}$。

式（5.10）和式（5.10）的含义比较直观。以式（5.10）为例，其中右边第一项η_i表示的是从时序信息上看双词b_i属于突发话题的概率；第二项$\frac{(n_{\neg i,1,k}+\alpha_1)}{(n_{\neg i,1,\cdot}+K_1\alpha_1)}$表示的是话题$k$在突发话题中的比例，最后一项表示的词$w_{i,1}$和$w_{i,2}$由突发话题$k$产生的概率。综合起来，我们可以发现双词$b_i$的话题类型$e_i$在Gibbs采样中的赋值取决于两个因素：时序先验$\eta_i$，以及$b_i$与所有突发话题$z_k$的内容相关程度。可以看到，$\eta_i$影响着双词$b_i$的话题类型判断，从而指导着突发话题的学习；但另一方面，η_i仅仅决定e_i赋值的其中一个因素，所以这种指导作用并非强制性的。正如假设1中所描述的，突发性强的双词仅仅是更可能由突发话题产生，而不是一定由突发话题所产生。所以CBTM中对双词话题类型e_i的概率建模恰恰是考虑到了这种不确定性。

针对BTM的Gibbs采样算法的详细步骤如算法5所示。首先，我们随机的分配一个话题给每一个双词作为初始状态。然后，在每次的迭代过程中，我们根据式（5.10）和

式（5.11）计算条件概率进行采样，逐个更新每个双词的话题类型标示变量与话题赋值。经过充分多的迭代次数之后，我们开始收集充分统计量$n_{e_i,k}$和$n_{e_i,k,w}$。利用这些充分统计量，我们可以估计各参数：

$$\hat{\psi}_{k,w} = \frac{n_{0,k,w}+\beta_0}{n_{0,k,\cdot}+W\beta_0}, \tag{5.12}$$

$$\hat{\theta}_{0,k} = \frac{n_{0,k}+\alpha_0}{n_{0,\cdot}+K_1\alpha_0}, \tag{5.13}$$

$$\hat{\phi}_{k,w} = \frac{n_{1,k,w}+\beta_1}{n_{1,k,\cdot}+W\beta_1}, \tag{5.14}$$

$$\hat{\theta}_{1,k} = \frac{n_{1,k}+\alpha_1}{n_{1,\cdot}+K_1\alpha_1}, \tag{5.15}$$

得到参数$\hat{\mathbf{\Theta}}=\{\hat{\boldsymbol{\theta}}_0,\hat{\boldsymbol{\theta}}_1,\hat{\mathbf{\Psi}},\ \hat{\mathbf{\Phi}}\}$之后，我们可以继续估计$\eta_i$的后验。首先，我们把式（5.12-5.15）带入式式(5.10-5.11)，可以得到：

$$P(e_i=0,c_i=k|b_i) = \frac{1}{Z_i}(1-\eta_i)\hat{\theta}_{0,k}\hat{\psi}_{k,w_{i,1}}\hat{\psi}_{k,w_{i,2}}, \tag{5.16}$$

$$P(e_i=1,z_i=k|b_i) = \frac{1}{Z_i}\eta_i\hat{\theta}_{1,k}\hat{\phi}_{k,w_{i,1}}\hat{\phi}_{k,w_{i,2}}, \tag{5.17}$$

其中，

$$Z_i=(1-\eta_i)\sum_{k=1}^{K_0}\hat{\theta}_{0,k}\hat{\psi}_{k,w_{i,1}}\hat{\psi}_{k,w_{i,2}}+\eta_i\sum_{k=1}^{K_1}\hat{\theta}_{1,k}\hat{\phi}_{k,w_{i,1}}\hat{\phi}_{k,w_{i,2}}.$$

然后，通过将式（5.17）中的z_i求和，我们就可以得到η_i的后验估计（即$P(e_i=1|b_i,\hat{\mathbf{\Theta}},\eta_i)$）：

$$\hat{\eta}_i = P(e_i=1|b_i) = \frac{1}{Z_i}\eta_i\sum_{k=1}^{K_1}\hat{\theta}_{1,k}\hat{\phi}_{k,w_{i,1}}\hat{\phi}_{k,w_{i,2}} \tag{5.18}$$

5.5.2　文档话题成分推断

学习出了语料集合中的话题之后，我们接下来讨论如何推断一个文档的话题成分，即$P(e=0,c|d)$和$P(e=1,z|d)$。和BTM一样，CBTM也没有对文档进行建模，无法直接学习出这两部分内容。下面我们采用和BTM类似的方法，即根据文档中的双词来推断文档的话题属性。该推断过程依赖于以下假设：

假设 2 给定一个双词b_i，我们假设它的话题关于其所在的文档d条件独立，即满足

$$P(e=0,c=k|b_i,d) = P(e=0,c=k|b_i) \tag{5.19}$$

$$P(e=1,z=k|b_i,d) = P(e=1,z=k|b_i) \tag{5.20}$$

假设文档d中包含N_d个双词$\{b_{d_j}|j \in [1, N_d]\}$，我们可以通过最大似然方式来估计$P(b_{d_j}|d)$，得到：

$$P(b_{d_j}|d) = \frac{n_d(b_{d_j})}{N_d}, \tag{5.21}$$

其中$n_d(b_{d_j})$是b_{d_j}在d中出现的次数。

根据式（5.19）和式(5.21)，我们可以推导出$P(e=1, c|d)$:

$$\begin{aligned} P(e=0, c=k|d) &= \sum_{j=1}^{N_d} P(e=0, c=k, b_{d_j}|d) \\ &= \sum_{j=1}^{N_d} P(e=0, c=k|b_{d_j}) P(b_{d_j}|d) \\ &= \frac{1}{N_d} \sum_{j=1}^{N_d} n_d(b_{d_j}) P(e=0, c=k|b_{d_j}), \end{aligned} \tag{5.22}$$

其中$P(e=0, c=k|b_{d_j})$的定义见式（5.16）。

类似地，我们可以推导出$P(e=1, z|d)$:

$$P(e=1, z=k|d) = \frac{1}{N_d} \sum_{j=1}^{N_d} n_d(b_{d_j}) P(e=1, z=k|b_{d_j}), \tag{5.23}$$

其中$P(e=1, z=k|b_{d_j})$的定义见式（5.17）。

最后，通过将式（5.23）中的z_i求和，我们还可以得到d中所有突发话题的比例：

$$P(e=1|d) = \frac{1}{N_d} \sum_{j=1}^{N_d} n_d(b_{d_j}) \hat{\eta}_{d_j}, \tag{5.24}$$

其中$\hat{\eta}_{d_j}$的定义见（5.18）。

5.6 实验结果与分析

本节中，我们通过实验来验证CBTM在短文本数据流上发现突发话题的效果。首先我们介绍实验中使用的数据集和基准方法。然后，我们对实验结果进行了展示和分析。

5.6.1 实验数据

我们这里仍然采用的是TREC 2011微博任务中的Tweets2011数据[2]，这也是目前的一个标准短文本数据集。该数据集包含了在2011年1月23日-2011年2月8日期间采集

[2]http://trec.nist.gov/data/tweets/

的tweets，总共约包含1600万条。我们按天将其分成17个时间段，然后逐天进行突发话题发现。

为了减少数据中的噪音，我们对原始数据做了如下的预处理: (a)过滤非英文的字符; (b)所有的英文字符都转换成了小写; (c)去除了出现次数小于10的词；(d)去除了词数少于2的文档；(e)去除重复的文档。最后剩下4230578条tweet，词汇表大小为98857。

5.6.2　基准方法

我们和以下方法进行了对比：

- **Twevent** Twevent [88]是目前最新的一个基于特征的突发事件检测方法。Twevent主要包含四个步骤：1）利用Wikipedia对微博进行切分，然后提取切分后的片段作为特征；2）提出了一种突发概率来计算特征的突发性；3）然后对突发性强的特征进行聚类得到一些话题；4）利用Wikipedia来过滤一些话题后得到事件。由于我们的目的是检验突发话题发现的效果而非突发事件，我们没有对微博进行内容切分，而仅仅用词来作为特征，并且我们在对突发特征聚类后，不再进行事件过滤。

- **oLDA** oLDA[86]是目前一种典型的基于话题学习的突发话题检测方法。oLDA使用在线LDA来计算每个时间段内的话题，然后采用后处理的方式来检测突发话题。具体地说，oLDA方法计算两个时间段内对应话题的词分布的Jensen-Shannon差异，若Jensen-Shannon差异大于一个阈值，则认为是一个突发话题。由于合理的阈值的选择并不是一件容易的事情。为了和其他方法比较方便，我们按Jensen-Shannon差异从大到小排序，取前K_1个话题，作为突发话题。oLDA中总的话题数目设为$K_0 + K_1$个。

- **BTM** 我们在每天的数据上单独的训练一个基本BTM[159]模型。BTM中总的话题数目也设为$K_0 + K_1$ 个。

除了话题个数之外，基准方法中的其余参数均采用原论文中的设置。CBTM的参数参照BTM中的参数设置，没有再特别调优，即$\alpha_0 = 50/K_0$，$\alpha_1 = 50/K_1$ 和$\beta_0 = \beta_1 = 0.01$。Gibbs采样过程中的迭代次数设为500次。根据我们之前的试探性实验，我们将普通话题的数目K_0固定为20，突发话题的数目K_1的取值包括$10, 30, 50$。

考虑到oLDA必须从第2天开始才能区分突发话题和普通话题，我们从第2天开始评测。对于Twevent、BTM和CBTM，每天的数据上都是独立地处理。对于BTM，我们先对当天和前一天话题分布按贪心策略进行匹配，然后选择最相似的K_0个作为普通话题，其余K_1个作为突发话题。由于Twevent 对突发的词聚类过程中不限定类的数目，为公平起见，我们选择包含突发词最多的K_1个和其他方法进行对比。同时，我们将每个类中的词按照其频率归一化，得到一个词分布作为该类别的表示。

方法	P@10	P@30	P@50
Twevent	0.592	0.681	0.636
oLDA	0.231	0.217	0.185
BTM	0.30	0.325	0.297
CBTM	**0.810**	**0.865**	**0.842**

表 5.2: 突发话题发现精度对比

.

5.6.3 突发话题发现

我们首先评价了CBTM在突发话题发现上的效果。由于事先并不知道微博数据流中有哪些突发话题，我们人工对各方法发现的突发话题进行了标注。我们将各方法发现的突发话题混合，然后随机提供给两个志愿者进行标注。对于每个突发话题，我们提供地信息包括：发现日期、概率最大的10个词、以及50条在该时间段内和该突发话题最相关的微博[3]。标注者可以使用Google和Twitter搜索来辅助判断。突发话题的认定标准：该话题的前10个词意义比较一致，而且确实存在突发性质。当且仅当两个志愿者都认为该话题是一个突发话题的时候，我们才认为该话题是一个准确的突发话题。最后，我们计算不同的突发话题数目K_1对应的精度（$P@K_1$）来评价各方法对突发话题发现的准确度。

表5.2列出来各方法的结果。我们发现：1）CBTM的精度一直大于0.8，明显超过其他方法，说明其能比较准确得发现突发话题。对比不同的突发话题数目K_1下的效果，我们还发现CBTM在$K_1 = 10$的时候效果稍差，主要是话题数目太少导致话题学习结果比较泛的原因。2) Twevent的效果比CBTM差，但比其他两种基准方法要好很多。这说明考虑词或双词的突发性对突发话题的发现有很大帮助。3) 和预期一致，两种采用普通话题模型的方法oLDA和BTM的结果都不高。这是因为普通话题模型没有直接考虑话题的突发性质，不能自动区分出普通话题和突发话题。我们还发现BTM的精度要稍高于oLDA。一个可能的原因是BTM在短文本上发现的话题可读性要好于oLDA。

为了对比各方法发现的突发话题的可读性，我们进一步评价这些突发话题的PMI-Score。PMI-Score的计算和前文一样，即采用500万篇Wikipedia文章作为辅助语料来计算每个突发话题中前10个词的平均PMI得分。图5.3展示各方法学习到的突发话题的PMI-Score。我们发现CBTM的PMI-Score和BTM的比较接近，显著高于oLDA（P-value$<$ 0.001）和Twevent，说明CBTM学习到的突发话题的可读性较高。BTM学习到的突发话题可读性虽然也较高，但如前面结果所述，这些话题并不一定是真正的突发话题。另外，我们还发现Twevent发现的突发话题的可读性随着K_1的增加，显著下降。

[3]对于Twevent方法，我们用jaccard稀疏找最相似的微博；对于其他方法，我们对于突发话题z找出$P(z|d)$最大的微博。

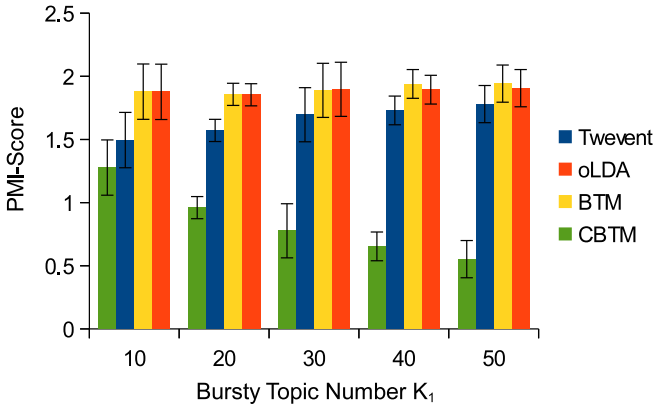


图 5.3: 各方法发现的突发话题的PMI-Score

方法	概率最大的前8个词	$\cos(z,h)$
Hashtag	#ntas win love award matt morning watching doctor	–
Twevent	#thegame malik ant melanie eastenders derwin #ntas tosh	0.18
oLDA	amazing vote award movie year listen awesome film	0.17
BTM	award shorty nominate oscar awards #ntas film win	0.26
CBTM	#ntas award awards win #nta national love tv	**0.59**

表 5.3: 各方法在2011年1月26日发现的和“#ntas”最相关的突发话题

这是由于Twevent中排在后面的突发话题中的词数越来越少，所以可读性也就越来越差。

为了进一步分析结果，我们从各方法中挑选了两个突发话题用作定性分析。为了方便比较，我们首先随机挑选了两个突发性较强且高频的hashtag，即“#ntas”（对应日期为2011/1/26）和“#tahrir”（对应日期为2011/02/04）。其中“#ntas”对应着“National Television Awards”（即国家电视奖），颁奖典礼在2011年1月26日举行；tahrir是埃及解放广场的名字，该hashtag表示的是2011年2月4日发生的埃及示威事件，可见这两个事件都比较热门。对于每个hashtag，我们先是把所有包含该hashtag的微博抽取出来，统计出其中的词频并归一化，视为一个hashtag对应的话题。然后，从每个方法的突发话题中，找出和该hashtag对应话题最相似的话题（用余弦相似度衡量）。表5.3和表5.4列出了各方法中和该hashtag对应话题最相似话题的概率最大的8个词，其中第二行表示的是hashtag对应的话题内容，第三列显示的是它们之间的相似度。

我们可以看到，CBTM中的词和该hashtag对应的词分布最相似。在表5.3中，其实CBTM发现的突发话题比hashtag本身的词分布更接近“#nta”。而Twevent中词比较生僻，同时还包含一些不相关的词，如“#thegame”等。说明突发词聚类对噪音比较敏感。oLDA中的词以常见词为主，而且只是部分词与“#nta”相关，因此相似度最低。BTM的结果和oLDA类似，里面多个公众评奖活动混合在一块，即“shorty

方法	概率最大的前8个词	$\cos(z,h)$
Hashtag	#tahrir #jan25 #egypt square mubarak people #mubarak protesters	–
Twevent	mubarak tahrir #mubarak #tahrir egyptians democracy	0.24
oLDA	people real hate talk social media talking person	0.06
BTM	news egypt obama top president world government uk	0.14
CBTM	#egypt #jan25 mubarak tahrir square #tahrir people protesters	**0.77**

表 5.4: 各方法在2011年2月4日发现的和“#tahrir”最相关的突发话题

z	The 10 most probable words	$\theta_{1,z}$
z_2	police officers shot shooting detroit twitter adam suspect year revenue (Two St. Petersburg police officers were shot and killed)	0.036
z_{11}	airport moscow police news killed people dead blast suicide explosion (Deadly suicide bombing hits Moscow's Domodedovo airport)	0.057
z_{15}	open #ausopen nadal australian murray mike tomlin cloud #cloud avril (Australian Open Tennis Championships 2011)	0.015
z_{25}	jack lalanne fitness 96 dies guru rip died age dead (Jack LaLanne: US fitness guru who last ate dessert in 1929 dies aged 96)	0.044
z_{26}	court emanuel rahm chicago ballot mayor mayoral run appellate rules (Court tosses Emanuel off Chicago mayoral ballot)	0.024

表 5.5: CBTM额外发现的5个突发话题，括号中由人工标注地相关新闻标题

award”和“oscar”。这说明，基本话题模型不能很好地区分出突发话题和普通话题。从表5.4中我们同样可以看到，CBTM的结果是和该事件最贴切的；而Twevent中包含“thugs”和“chanting”等无关词；BTM的结果和该事件相关，但同时包含了较多的高频词，如“top”和“world”；oLDA的结果和该事件的关联性最差。以上结果表明，CBTM比其他方法能更全面地学习出突发话题，而且话题的可读性更好。

此外，在表5.5中，我们还展示5个CBTM在2011年1月24日单独发现的5个突发话题（$K_1 = 30$）。在第二列中的括号中，我们通过用Google搜索话题中前几个关键词得到的新闻标题，作为该话题的标注信息。可以看到，这个突发话题和对应的新闻事件非常吻合。第三列中的$\theta_{i,z}$表示这个话题的概率，我们可以看出这几个话题的出现概率并不高，因此没有被其他方法发现。该结果表明，CBTM对突发话题的敏感度比其他方法要高。

5.6.4 普通话题与突发话题对比

接下来我们来验证CBTM对普通话题和突发话题的区分能力。由于Twevent不具备普通话题发现的能力，这里我们仅仅比较CBTM、oLDA与BTM。首先，我们给出了CBTM学习到的概率最大的5个普通话题，见表5.6。我们发现，除第4个话题是关于

c	概率最大的前10个词	$P(c\|e=0)$
c_1	time good people day today make life love back feel	0.118
c_2	shit ass fuck nigga man wit bout bitch damn dat	0.113
c_3	time today good back work day school people class home	0.067
c_4	online business social job free blog media marketing post web	0.066
c_5	love watch show movie watching video good great time tv	0.064

表 5.6: CBTM发现的概率最大的5个普通话题

商业信息之外，其余几个都是和日常生活相关。这一结果说明Twitter中关于日常生活的消息占主导地位，和Pear Analytics的调查结果[5]相符。

直观上讲，普通话题的内容应该随时间变化不大，而突发话题则相反。接下来，我们通过分析各方法中这两类不同话题的内容随时间变化情况，来判断其对普通话题和突发话题的区分度。在时间段t内，对于每一类话题，我们取其中每个话题中概率最大的10个词构造一个词集，分别用$\mathbb{W}_c^{(t)}$和$\mathbb{W}_z^{(t)}$表示。然后我们这个两个词集相比上一个时间段对应词集（即$\mathbb{W}_c^{(t-1)}$和$\mathbb{W}_z^{(t-1)}$）的重复率（overlap ratio）：

$$\text{OverlapRatio}_c(t) = \frac{|\mathbb{W}_c^{(t-1)} \cap \mathbb{W}_c^{(t)}|}{|\mathbb{W}_c^{(t)}|},$$
$$\text{OverlapRatio}_z(t) = \frac{|\mathbb{W}_z^{(t-1)} \cap \mathbb{W}_z^{(t)}|}{|\mathbb{W}_z^{(t)}|}.$$

显然，重复率越高说明该类型话题越稳定。从图5.4（a）中我们可看出，各方法学习到的普通话题的重复率比较接近，在0.7左右，说明普通话题随时间变化不大。但是在图5.4（b）中，各方法发现的突发话题的重复率差别较大。其中oLDA中的突发话题重复率与普通话题重复率很接近，说明其对新突发话题发现的能力较弱。这主要是由于oLDA使用的online LDA算法使用了较多的历史信息作为先验。CBTM中的突发话题重复率最低，反映出CBTM具有较强的新突发话题发现能力。

图5.5（a）和（b）分别展示了不同话题数目时的普通话题和突发话题的平均重复率。我们发现普通话题的重复率仍然很稳定。而增加突发话题的数目反而会提高oLDA和BTM的突发话题的重复率。说明增加突发话题数目并不能改进基本话题模型对突发话题学习的能力。相反，CBTM的突发话题重复率有下降趋势，说明此时CBTM能发现更多的新突发话题。以上结果充分说明了相比其他方法，CBTM能更好的区分普通话题和突发话题。

5.6.5　文档中的突发话题成分判断

除了发现突发话题之外，CBTM通过推断消息中的普通话题和突发话题成分。因此，CBTM可以用来对消息进行突发话题相关的语义分析，比如突发消息聚类，突发

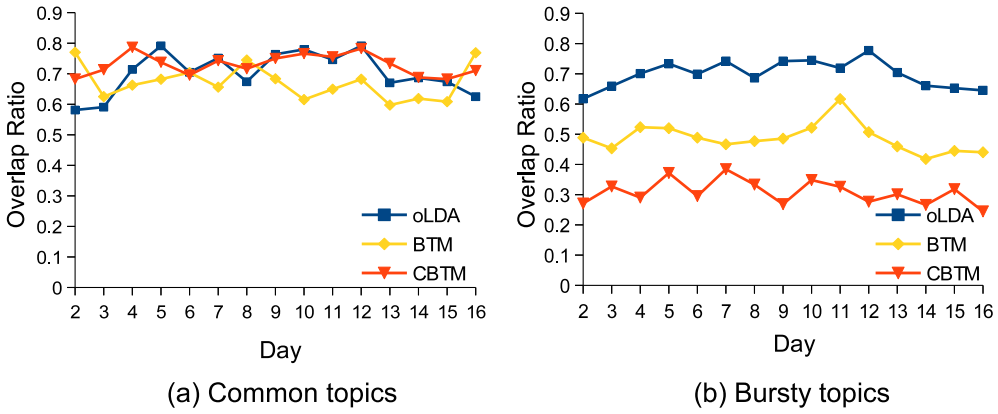


图 5.4: 不同日期中的两类话题词重复度

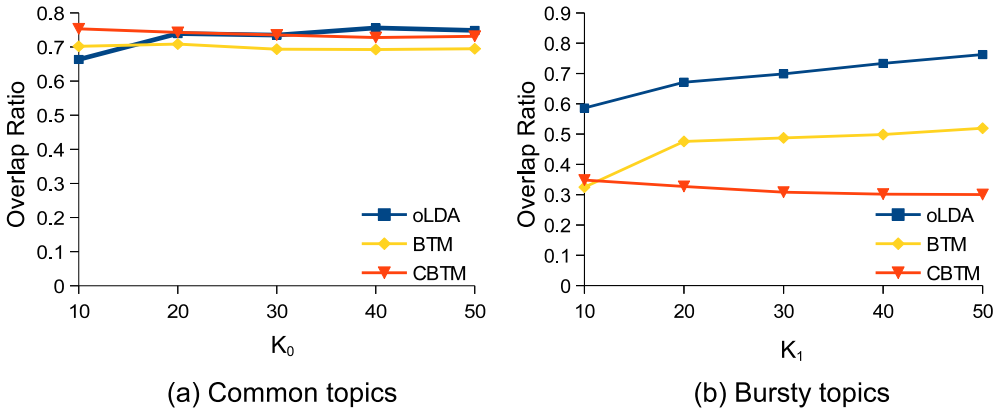


图 5.5: 词重复度随话题个数变化情况

话题相关消息查找等。接下来，我们对CBTM对文档中的突发话题成分判断进行评价。

由于微博中消息噪音多，上下文信息不足，对每条消息进行人工标注非常困难。幸运地是，在微博中部分消息中本身包含作者对消息话题的标注，即hashtag[4]。但是，并不是所有的hashtag都表示一个突发话题。为了保证减少噪音干扰，我们首先人工地从从第2-17天的微博中选择每天出现次数超过平均每天出现次数的两倍的hashtag。然后，将它们按出现次数排序，选择5个高频且意义比较明确的hashtag作为测试集中消息的类别标签。我们这些hashtag对应的文档中随机采样1/10的消息去除该hashtag后作为测试集。

这里，我们用消息聚类的方式来评价CBTM对消息中的突发话题成分判断的好坏。对于oLDA、BTM和CBTM，我们把每个突发话题设为一个类，然后把每条消息d赋予给$P(e==1|d)$的类；对于Twevent，我们按类与消息之间的Jaccard系数，把消息赋予给最相似的类。然后我们判断消息聚类效果与原hashtag标注的一致性。我们采用三种常

[4] 在Twitter中，hashtag的形式为“#keyword”；在新浪微博中，hashtag的形式为“#关键词#”

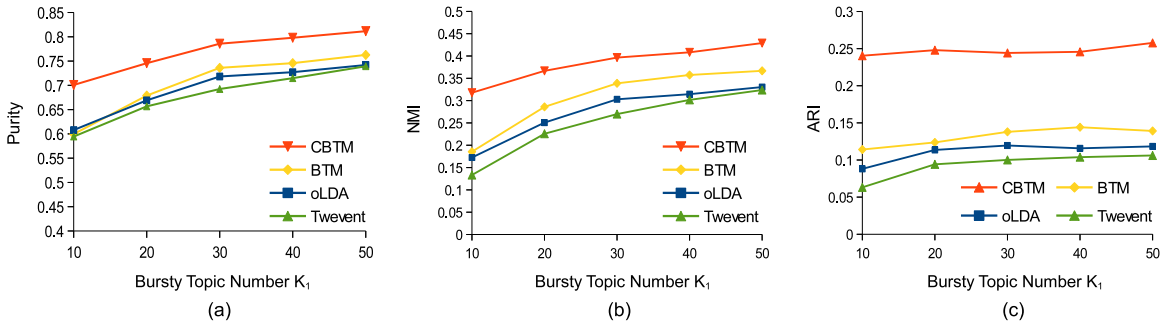


图 5.6: 消息聚类结果对比

排序	Tweet
1	@JO_ROCKS1 @Catherined38 i like ant n dec 2 lol
2	YESSSSSSSSSSSSSS easties winsssssssss :D:D:D #eastenders #NTA
3	Come on Lacey! #ntas
4	Howay Ant and Dec
5	xfactorrrr to win!!! #ntas

表 5.7: CBTM发现的和突发话题“#ntas”最相关的5条消息

用的聚类评价指标进行评价，即purity[172]，NMI[137]和ARI[76]。聚类结果如图5.6所示。我们可以看到，CBTM明显优于其他三种方法。说明CBTM对消息的突发话题成分分析最为准确。Twevent的结果最差，可能原因在于其只利用突发词来表达突发话题，难以全面准确判断突发话题和消息之间的相似度。

最后，我们展示CBTM对突发话题相关消息识别的两个示例。这里，我们仍然选用的是表5.3和表5.4列出的两个突发话题，即分别由hashtag “#ntas”和“#tahrir”，其对应的最相关的5条消息如表5.7和表5.8所示。在表5.7中，第1条和第4条消息中的“Ant and Dec”正式2011年获得NTAS将的一个组合的名称。所以，我们可以看到，这些消息和“#ntas”确实非常相关。在表5.4中列出的5条消息也很容易看出和埃及解放广场示威事件非常相关。该结果表明，CBTM能较好的识别出和突发话题相关的消息。

5.7　小结

本章中我们针对短文本数据流中的突发话题建模进行了研究。在像微博等短文本数据流中，每天都会有很多新的话题涌现，这些突发的话题比普通话题更能吸引人们的兴趣。突发话题是一种特殊的话题，传统的话题模型由于没有考虑话题的突发性，因此不能自动区分数据中的突发话题和普通话题。相关的一些工作依赖于后处理措施或者一些启发式技巧来发现突发话题，并没有直接对突发话题建模。

排序	Tweet
1	Bambuser — Tahrir square! (waelabbas) http://bit.ly/dTYj6b
2	@alaa in Tahrir http://bit.ly/gWRaJn #jan25
3	Burmese Fend for Themselves in Cairo
4	How i wish i was in Tahrir . . NOW ! #Tahrir #Jan25 #Egypt
5	@Gsquare86 will u be in tahrir with us today? #jan25

表 5.8: CBTM发现的和突发话题“#tahrir”最相关的5条消息

本章中，我们在双词话题模型的基础上提出了组合双词话题模型来解决这个问题。组合双词话题模型根据双词的突发性来对突发话题和普通话题进行有区分的建模。其基本思想是突发性强的双词更可能是由突发话题所产生，突发性弱的双词更可能由普通话题所产生。利用双词的突发性作为指导信息，该模型能自动学习出突发话题和普通话题。通过在Twitter数目上的实验验证，我们发现该模型的突发话题发现能力明显优于现有方法，其发现的突发话题更全面、可读性也更好。

本章的研究成果计划投稿到The 23th ACM International Conference on Information and Knowledge Management (CIKM 2014)，论文题目为“Modeling Bursty Topics in Microblog Stream with a Composte Biterm Topic Model”。

第六章　总结与展望

6.1　论文工作总结

随着Web2.0以及社交媒体的发展，人们越来越广泛地使用互联网来发布和分享信息。大量的用户产生内容带来了新一轮的信息爆炸。其中，大部分用户产生内容都为短文本信息，如评论、微博、状态消息等。相比以往的长文本信息，如新闻、博客，这些短文本信息的长度虽短，但规模大、更新快、内容更丰富多样。

互联网上海量的短文本信息是一座有待开采的金矿，其中蕴含着丰富的有价值信息。挖掘这些信息对舆情监控、用户行为分析、商业情报收集等领域都有重要意义。然而这些短文本信息也给现有的文本处理方法带来了严峻的挑战。首先，短文本内容非常短，内容非常稀疏，我们难以准确判断一条短文本中的语义。其次，这些短文本的动态性强，不仅规模增长快，而且内容复杂多变。

在此背景下，本文开展了针对短文本的话题建模方法的研究，以提高对互联网中短文本数据的语义分析水平。基于互联网应用当中的短文本的特点，我们展开了三方面的研究。

- 首先，我们研究了如何克服话题建模方法在短文本上所面临内容稀疏性问题。传统话题模型过于依赖文档内部的词共现关系来学习话题，在文档过短时，难以准确估计文档内部的话题结构，从而影响整体的话题学习。为了克服这个问题，我们采用了一种不同的话题学习思路，即直接通过全局丰富的词共现关系来学习话题。根据统计自然语言原理，两个词共现次数越多，其语义越相关，因此也就越可能属于同一个话题。基于这个基本假设，我们提出了一个新的话题模型——双词话题模型。双词话题模型建模的文档集合中一个双词（即一个无序共现词对）的产生过程，即从同一个全局话题中抽取出两个词构成。由于没有对文档进行建模，双词话题模型对话题的学习不受文档长度的影响。

- 其次，我们针对实际应用中短文本数据的大规模动态性特点，研究了如何在像微博这种大规模动态数据下进行话题建模。大规模动态短文本数据就要求话题学习算法不仅要具备高度可伸缩性，同时还能即时的反映数据的变化。为此，我们在双词话题模型的基础上提出了两种在线话题学习方法。这两种在线算法通过使用一小部分最近的历史数据来增量更新模型，从而使模型更新所需要的时间和内存开销降低到常数级别。此外，这两种算法还能追踪数据中话题的演化。

- 最后，我们针对短文本数据中突发话题涌现的特点，研究了短文本数据流中的突发话题建模问题。突发话题是一类特殊的话题，基本的话题模型并没有考虑到话

题的突发性，不能有效地学习到突发话题。目前的一些做法基本都是靠一些启发式的方式来找出突发话题。我们在双词话题模型的基础上提出了一个组合双词话题模型来建模突发话题。其基本假设是：突发性强的双词更可能由突发话题产生，而突发性弱的双词更可能由普通话题产生。借助双词的突发性，该模型能自动学习出短文本数据流中的突发话题。

6.2 论文主要贡献

本文的主要贡献包括:

- 我们提出了首个通用的短文本话题学习模型——双词话题模型。该模型创造性的通过建模双词的产生过程来学习话题，打破了传统话题模型必须通过文档建模来学习话题的常规，因此也避免了短文本文档过短所带来的数据稀疏性问题。该模型有效的提高了短文本话题建模的水平，也拓宽了话题建模的思路。另外，该模型比较简单且容易实现，内存复杂度较低，可以被广泛地应用到各种短文本内容分析相关领域中。

- 为了有效地对大规模动态短文本数据进行话题学习，我们基于双词话题模型提出了两种在线话题学习算法。它们有效的降低了模型更新所需要的时间和空间代价，提升了双词话题模型处理大规模数据的能力。同时，还能动态的追踪话题的演化。通过在大规模微博数据上的实验表明，这两个在线话题学习算法的效果和批处理算法相差不大，但效率要高很多。同时，我们也验证这两个算法追踪流式数据中话题演化的能力。这表明这两种算法能很好的适应在线话题建模需求。

- 针对短文本流式数据中话题的突发涌现问题，我们提出了一种对短文本数据流中突发话题建模的方法，即组合双词话题模型。该模型能自动利用双词的突发性来指导突发话题的学习。与其他方法相比，该模型无需任何后处理手段和启发式技巧，而且学习到的突发话题更前面，可解释性更好，具有较高的应用价值。该方法同时也验证双词话题模型的良好可扩展性，拓宽了其应用范围。

6.3 进一步工作

随着Web2.0以及社交媒体的兴起，短文本信息在互联网上越来越多。尽管过去文本挖掘已经有很多工作与进展，但大多是基于长文本进行的，对短文本处理的技术仍不成熟。本文对短文本话题建模技术的研究，为提高目前短文本语义分析与处理的水平作出了尝试。但现阶段对短文本语义分析的研究距离让计算机精准地去理解和处理短文本数据，还有很多工作需要去做。我们对接下来进一步的工作规划如下：

- **多源异质数据下的话题建模** 目前，我们仅仅利用到了短文本数据内部的词共现信息来建模话题，对效果的提升难免会有限。从人的角度看，人对自然语言理解

的能力正是从各种外部数据源中不断学习得到的。因此在大数据时代的背景下，如何广泛利用其他数据源中的知识，如Wikipedia，开发网页等，来进一步提高计算机短文本的理解与处理能力，是一个非常值得关注的问题。

- **深层结构话题建模** 目前，我们研究的话题模型结构都比较简单，只有一层潜在语义结构，话题的数目也很有限。这种简单结构的话题模型只能大概反映文本中的语义，难以准确全面的描述文本内容。未来，我们将考虑更复杂的话题结构，如层次甚至是图状等，来提升话题模型的表达能力。

- **多特征话题建模** 在真实的应用环境中，短文本消息还包含着大量的其他非文本上下文信息，如作者，地点，人物关系，时间等等。由于短文本的内容稀疏性，这些特征对短文本内容的理解有很大帮助。如何在特定应用场景下，有效的利用这些特征，来进一步提高对短文本处理的准确度，也是一个很有意义的问题。

- **相关应用** 短文本话题建模技术是一种基本的自动语义分析工具，可以被广泛应用到短文本挖掘的各个领域中，如微博检索，消息推荐，用户行为分析等等。本文中，我们将双词话题模型成功应用到了微博突发话题发现上。未来，我们将在其他应用领域做出更多的尝试。

附录 A　附录

A.1　BTM的Gibbs采样条件概率$P(z_i|\mathbf{z}_{\neg i},\mathbb{B})$的推导

根据链式法则，该条件概率可以写成:

$$P(z_i|\mathbf{z}_{\neg i},\mathbb{B}) = \frac{P(\mathbf{z},\mathbb{B})}{P(\mathbf{z}_{\neg i},\mathbb{B})} \propto \frac{P(\mathbb{B}|\mathbf{z})P(\mathbf{z})}{P(\mathbb{B}_{\neg i}|\mathbf{z}_{\neg i})P(\mathbf{z}_{\neg i})}. \tag{A.1}$$

在式（A.1）中，$P(\mathbb{B}|\mathbf{z})$可以通过对$\mathbf{\Phi}$积分得到：

$$\begin{aligned} P(\mathbb{B}|\mathbf{z}) &= \int P(\mathbb{B}|\mathbf{z},\mathbf{\Phi})P(\mathbf{\Phi})\mathrm{d}\mathbf{\Phi} \\ &= \int \left(\prod_{i=1}^{N_B} P(b_i|z_i,\boldsymbol{\phi}_{z_i})\right) P(\mathbf{\Phi})\mathrm{d}\mathbf{\Phi} \\ &= \int \prod_{k=1}^{K}\left(\frac{\Gamma(W\beta)}{\Gamma(\beta)^W}\prod_{w=1}^{W}\boldsymbol{\phi}_{k,w}^{n_{w|k}+\beta-1}\mathrm{d}\boldsymbol{\phi}_{\boldsymbol{k}}\right) \\ &= \left(\frac{\Gamma(W\beta)}{\Gamma(\beta)^W}\right)^K \prod_{k=1}^{K}\frac{\prod_{w=1}^{W}\Gamma(n_{w|k}+\beta)}{\Gamma(n_{\cdot|k}+W\beta)}, \end{aligned} \tag{A.2}$$

这里$\Gamma(\cdot)$是标准Gamma函数[1]，$n_{w|k}$是词w赋给话题k的次数，且$n_{\cdot|k}=\sum_{w=1}^{W} n_{w|k}$。$P(\mathbf{z})$可以通过对$\boldsymbol{\theta}$积分得到:

$$\begin{aligned} P(\mathbf{z}) &= \int P(\mathbf{z}|\boldsymbol{\theta})P(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} \\ &= \int \left(\prod_{i=1}^{N_B} P(z_i|\boldsymbol{\theta})\right) P(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} \\ &= \int \frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K}\prod_{k=1}^{K}\theta_k^{n_k+\alpha-1}\mathrm{d}\boldsymbol{\theta} \\ &= \frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K}\frac{\prod_k\Gamma(n_k+\alpha)}{\Gamma(N_B+K\alpha)}. \end{aligned} \tag{A.3}$$

类似地，我们可以计算出$P(\mathbb{B}_{\neg i}|\mathbf{z}_{\neg i})$和$P(\mathbf{z}_{\neg i})$：

$$P(\mathbb{B}_{\neg i}|\mathbf{z}_{\neg i}) = \left(\frac{\Gamma(W\beta)}{\Gamma(\beta)^W}\right)^K \prod_{k=1}^{K}\frac{\prod_{w=1}^{W}\Gamma(n_{\neg i,w|k}+\beta)}{\Gamma(n_{\neg i,\cdot|k}+W\beta)}, \tag{A.4}$$

$$P(\mathbf{z}_{\neg i}) = \frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K}\frac{\prod_{k=1}^{K}\Gamma(n_{\neg i,k}+\alpha)}{\Gamma(N_B-1+K\alpha)}, \tag{A.5}$$

[1] 详细定义参见：http//en.wikipedia.org/wiki/Gamma_function。特别地，但x为正整数的时候，Gamma函数满足$\Gamma(x)=(x-1)!$

其中$\neg i$表示不包括双词b_i。把式（A.2-A.5）代入式（A.1），同时根据$\Gamma(x+1)=x\Gamma(x)$，消掉公共因子后便可得到Gibbs采样所需的条件概率分布：

$$P(z_i=k|\mathbf{z}_{\neg i},\mathbb{B}) \propto (n_{\neg i,k}+\alpha)\frac{(n_{\neg i,w_{i,1}|k}+\beta)(n_{\neg i,w_{i,2}|k}+\beta)}{(n_{\neg i,\cdot|k}+W\beta)^2}.$$

A.2 BTM中$\phi_{k,w}$和θ_k的估计

给定超参α和β，双词集合$\mathbb{B}$以及所有双词的话题赋值$\mathbf{z}$，我们可以通过贝叶斯法则和Dirichlet-multinomial共轭性质推导出$\boldsymbol{\Phi}$和$\boldsymbol{\theta}$的概率分布：

$$P(\boldsymbol{\theta}|\mathbf{z},\alpha) = \frac{1}{Z_\theta}\prod_{i=1}^{N_B} P(z_i|\boldsymbol{\theta})P(\boldsymbol{\theta}|\alpha) = \mathrm{Dir}(\boldsymbol{\theta}|\alpha+\mathbf{n}), \tag{A.6}$$

$$P(\boldsymbol{\phi}_k|\mathbf{z},\mathbb{B},\beta) = \frac{1}{Z_{\phi_k}}\prod_{i=1}^{N_B} P(z_i|\boldsymbol{\theta})P(\boldsymbol{\theta}|\alpha) = \mathrm{Dir}(\boldsymbol{\phi}_k|\beta+\mathbf{n_k}), \tag{A.7}$$

其中向量$\mathbf{n}=\{n_k\}_{k=1}^K$，向量$\mathbf{n_k}=\{n_{w|k}\}_{w=1}^W$，$Z_{\boldsymbol{\theta}}$和$Z_{\boldsymbol{\phi}_k}$为归一化因子，$\mathrm{Dir}(\cdot)$表示一个Dirichlet分布。

注意到Dirichlet分布$\mathrm{Dir}(\mathbf{x}|\boldsymbol{\alpha})$的期望为is $\mathrm{E}(x_i)=\frac{\alpha_i}{\sum_{k=1}^K\alpha_k}$。根据式（A.6-A.7），我们可以用其期望来估计$\phi_{k,w}$和θ_k：

$$\phi_{k,w} = \frac{n_{w|k}+\beta}{n_{\cdot|k}+W\beta},$$
$$\theta_k = \frac{n_k+\alpha}{N_B+K\alpha}.$$

参 考 文 献

[1] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, pages 30–38. Association for Computational Linguistics, 2011.

[2] James Allan, Jaime G Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. Topic detection and tracking pilot study final report. pages 194–218, 1998.

[3] James Allan, Ron Papka, and Victor Lavrenko. On-line new event detection and tracking. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 37–45. ACM, 1998.

[4] Loulwah AlSumait, Daniel Barbará, and Carlotta Domeniconi. On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *ICDM*, pages 3–12. IEEE, 2008.

[5] Pear Analytics. Twitter study–august 2009. *San Antonio, TX: Pear Analytics. Available at: www. pearanalytics. com/blog/wp-content/uploads/2010/05/Twitter-Study-August-2009. pdf*, 2009.

[6] Avi Arampatzis and Jaap Kamps. A study of query length. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 811–812. ACM, 2008.

[7] Rachit Arora and Balaraman Ravindran. Latent dirichlet allocation based multi-document summarization. In *Proceedings of the second workshop on Analytics for noisy unstructured text data*, pages 91–97. ACM, 2008.

[8] Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 27–34. AUAI Press, 2009.

[9] Hagai Attias. A variational Bayesian framework for graphical models. *NIPS*, 12(1-2):209–215, 2000.

[10] Leif Azzopardi, Mark Girolami, and Keith van Risjbergen. Investigating the relationship between language model perplexity and ir precision-recall measures. In

Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pages 369–370. ACM, 2003.

[11] Arindam Banerjee and Sugato Basu. Topic models over text streams: A study of batch and online unsupervised learning. In *SDM*, volume 7, pages 437–442, 2007.

[12] Somnath Banerjee, Krishnan Ramanathan, and Ajay Gupta. Clustering short texts using wikipedia. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 787–788. ACM, 2007.

[13] Luciano Barbosa and Junlan Feng. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 36–44. Association for Computational Linguistics, 2010.

[14] Hila Becker, Mor Naaman, and Luis Gravano. Beyond trending topics: Real-world event identification on twitter. *ICWSM*, 11:438–441, 2011.

[15] Lidong Bing, Wai Lam, and Tak-Lam Wong. Using query log and social tagging to refine queries based on latent topics. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 583–592. ACM, 2011.

[16] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[17] David Blei and John Lafferty. Correlated topic models. *Advances in neural information processing systems*, 18:147, 2006.

[18] David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[19] David M Blei, Thomas L Griffiths, Michael I Jordan, and Joshua B Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *NIPS*, volume 16, 2003.

[20] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

[21] David M Blei and Jon D McAuliffe. Supervised topic models. In *NIPS*, volume 7, pages 121–128, 2007.

[22] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[23] Phil Blunsom, Edward Grefenstette, Nal Kalchbrenner, et al. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014.

[24] I. Bordino, C. Castillo, D. Donato, and A. Gionis. Query similarity by projecting the query-flow graph. In *SIGIR*, pages 515–522. ACM, 2010.

[25] Jonathan Boyd-Graber, Jordan Chang, Sean Gerrish, Chong Wang, and David Blei. Reading tea leaves: How humans interpret topic models. In *NIPS*, 2009.

[26] Jordan Boyd-Graber and David M Blei. Syntactic topic models. Technical Report arXiv:1002.4665, Feb 2010.

[27] Samuel Brody and Noemie Elhadad. An unsupervised aspect-sentiment model for online reviews. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 804–812. Association for Computational Linguistics, 2010.

[28] Deng Cai, Xiaofei He, Xiaoyun Wu, and Jiawei Han. Non-negative matrix factorization on manifold. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 63–72. IEEE, 2008.

[29] Deng Cai, Qiaozhu Mei, Jiawei Han, and Chengxiang Zhai. Modeling hidden topics on document manifold. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 911–920. ACM, 2008.

[30] Kevin R Canini, Lei Shi, and Thomas L Griffiths. Online inference of topics with latent dirichlet allocation. In *International conference on artificial intelligence and statistics*, pages 65–72, 2009.

[31] Mark J Carman, Fabio Crestani, Morgan Harvey, and Mark Baillie. Towards query log based personalization using topic models. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1849–1852. ACM, 2010.

[32] Ben Carterette and Praveen Chandar. Probabilistic models of ranking novel documents for faceted topic retrieval. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1287–1296. ACM, 2009.

[33] Mario Cataldi, Luigi Di Caro, and Claudio Schifanella. Emerging topic detection on twitter based on temporal and social terms evaluation. In *Proceedings of the Tenth International Workshop on Multimedia Data Mining*, page 4. ACM, 2010.

[34] Asli Celikyilmaz and Dilek Hakkani-Tur. A hybrid hierarchical model for multi-document summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 815–824. Association for Computational Linguistics, 2010.

[35] Asli Celikyilmaz and Dilek Hakkani-Tür. Discovery of topically coherent sentences for extractive summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 491–499. Association for Computational Linguistics, 2011.

[36] Allison June-Barlow Chaney and David M Blei. Visualizing topic models. In *ICWSM*, 2012.

[37] Jonathan Chang, Jordan L Boyd-Graber, Sean Gerrish, Chong Wang, and David M Blei. Reading tea leaves: How humans interpret topic models. In *NIPS*, volume 22, pages 288–296, 2009.

[38] Ying-Lang Chang and Jen-Tzung Chien. Latent dirichlet learning for document summarization. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 1689–1692. IEEE, 2009.

[39] Chaitanya Chemudugunta, Padhraic Smyth, and Mark Steyvers. Modeling general and specific aspects of documents with a probabilistic topic model. In *NIPS*, volume 19, pages 241–248, 2006.

[40] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi. Short and tweet: experiments on recommending content from information streams. In *CHI*, pages 1185–1194. ACM, 2010.

[41] Mengen Chen, Xiaoming Jin, and Dou Shen. Short text classification improved by learning multi-granularity topics. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 1776–1781. AAAI Press, 2011.

[42] Mengen Chen, Xiaoming Jin, and Dou Shen. Short text classification improved by learning multi-granularity topics. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 1776–1781. AAAI Press, 2011.

[43] Xi Chen, Yanjun Qi, Bing Bai, Qihang Lin, and Jaime G Carbonell. Sparse latent semantic analysis. In *SDM*, pages 474–485. SIAM, 2011.

[44] Yan Chen, Hadi Amiri, Zhoujun Li, and Tat-Seng Chua. Emerging topic detection for organizations from microblogs. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 43–52. ACM, 2013.

[45] Xueqi Cheng, Xiaohui Yan, Yanyan Lan, and Jiafeng Guo. Btm: Topic modeling over short texts. In *IEEE Transactions on Knowledge and Data Engineering*, accepted.

[46] Jen-Tzung Chien and Meng-Sung Wu. Adaptive bayesian latent semantic analysis. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(1):198–207, 2008.

[47] Sanjiv R Das and Mike Y Chen. Yahoo! for amazon: Sentiment extraction from small talk on the web. *Management Science*, 53(9):1375–1388, 2007.

[48] Hal Daumé III and Daniel Marcu. Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 305–312. Association for Computational Linguistics, 2006.

[49] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 241–249. Association for Computational Linguistics, 2010.

[50] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[51] Jean-Yves Delort and Enrique Alfonseca. Dualsum: a topic-model based approach for update summarization. In *Proceedings of the 13th Conference of the European*

Chapter of the Association for Computational Linguistics, pages 214–223. Association for Computational Linguistics, 2012.

[52] Arthur P Dempster, Nan M Laird, Donald B Rubin, et al. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal statistical Society*, 39(1):1–38, 1977.

[53] Qiming Diao, Jing Jiang, Feida Zhu, and Ee-Peng Lim. Finding bursty topics from microblogs. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 536–544. Association for Computational Linguistics, 2012.

[54] Andr e Gohr, Alexander Hinneburg, Ren e Schult, and Myra Spiliopoulou. Topic evolution in a stream of documents. In *SDM*, 2009.

[55] Paolo Ferragina and Ugo Scaiella. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1625–1628. ACM, 2010.

[56] Gabriel Pui Cheong Fung, Jeffrey Xu Yu, Philip S Yu, and Hongjun Lu. Parameter free bursty events detection in text streams. In *Proceedings of the 31st international conference on Very large data bases*, pages 181–192. VLDB Endowment, 2005.

[57] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, pages 1606–1611, 2007.

[58] Eric Gaussier and Cyril Goutte. Relation between plsa and nmf and implications. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 601–602. ACM, 2005.

[59] Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the bayesian restoration of images. *PAMI*, (6):721–741, 1984.

[60] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, pages 1–12, 2009.

[61] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, 2004.

[62] T.L. Griffiths, M. Steyvers, D.M. Blei, and J.B. Tenenbaum. Integrating topics and syntax. *NIPS*, 17:537–544, 2005.

[63] A. Gruber, M. Rosen-Zvi, and Y. Weiss. Hidden topic Markov models. *AISTATS*, 2007.

[64] Jiafeng Guo, Xueqi Cheng, Gu Xu, and Xiaofei Zhu. Intent-aware query similarity. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 259–268. ACM, 2011.

[65] Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. Named entity recognition in query. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 267–274. ACM, 2009.

[66] Aria Haghighi and Lucy Vanderwende. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370. Association for Computational Linguistics, 2009.

[67] Sanda M Harabagiu and Finley Lacatusu. Generating single and multi-document summaries with gistexter. In *Document Understanding Conferences*, 2002.

[68] Jiyin He, Edgar Meij, and Maarten de Rijke. Result diversification based on query-specific cluster ranking. *Journal of the American Society for Information Science and Technology*, 62(3):550–571, 2011.

[69] G. Heinrich. Parameter estimation for text analysis. *Technical report*, 2005.

[70] Matthew Hoffman, Francis R Bach, and David M Blei. Online learning for latent Dirichlet allocation. In *NIPS*, pages 856–864, 2010.

[71] Thomas Hofmann. Probabilistic latent semantic analysis. *The Conference on Uncertainty in Artificial Intelligence*, 1999.

[72] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[73] L. Hong and B.D. Davison. Empirical study of topic modeling in Twitter. In *Proceedings of the First Workshop on Social Media Analytics*, pages 80–88. ACM, 2010.

[74] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[75] Xia Hu, Nan Sun, Chao Zhang, and Tat-Seng Chua. Exploiting internal and external semantics for the clustering of short texts using world knowledge. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 919–928. ACM, 2009.

[76] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[77] Sampath Jayarathna, Atish Patra, and Frank Shipman. Mining user interest from search tasks and annotations. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1849–1852. ACM, 2013.

[78] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 151–160. Association for Computational Linguistics, 2011.

[79] O. Jin, N.N. Liu, K. Zhao, Y. Yu, and Q. Yang. Transferring topical knowledge from auxiliary long texts for short text clustering. In *CIKM*, pages 775–784. ACM, 2011.

[80] Wei Jin, Hung Hay Ho, and Rohini K Srihari. Opinionminer: a novel machine learning system for web opinion mining and extraction. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1195–1204. ACM, 2009.

[81] Yohan Jo and Alice H Oh. Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 815–824. ACM, 2011.

[82] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. *An introduction to variational methods for graphical models*. Springer, 1998.

[83] Shiva Prasad Kasiviswanathan, Prem Melville, Arindam Banerjee, and Vikas Sindhwani. Emerging topic detection using dictionary learning. In *Proceedings of the*

20th ACM international conference on Information and knowledge management, pages 745–754. ACM, 2011.

[84] Himabindu Lakkaraju, Indrajit Bhattacharya, and Chiranjib Bhattacharyya. Dynamic multi-relational Chinese restaurant process for analyzing influences on users in social media. In *ICDM*, pages 389–398, Washington, DC, USA, 2012. IEEE Computer Society.

[85] Hugo Larochelle and Stanislas Lauly. A neural autoregressive topic model. In *NIPS*, pages 2717–2725, 2012.

[86] Jey Han Lau, Nigel Collier, and Timothy Baldwin. On-line trend analysis with topic models:\# twitter trends detection topic model online. In *COLING*, pages 1519–1534, 2012.

[87] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[88] Chenliang Li, Aixin Sun, and Anwitaman Datta. Twevent: segment-based event detection from tweets. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 155–164. ACM, 2012.

[89] Jiwei Li, Sujian Li, Xun Wang, Ye Tian, and Baobao Chang. Update Summarization Using a Multi-level Hierarchical Dirichlet Process Model. *COLING*, 1(December):1603–1618, 2012.

[90] Tao Li and Chris Ding. The relationships among various nonnegative matrix factorization methods for clustering. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 362–371. IEEE, 2006.

[91] Wei Li and Andrew McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *ICML*, 2006.

[92] Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 375–384. ACM, 2009.

[93] Chenghua Lin, Yulan He, Richard Everson, and Stefan Ruger. Weakly supervised joint sentiment-topic detection from text. *Knowledge and Data Engineering, IEEE Transactions on*, 24(6):1134–1145, 2012.

[94] Cindy Xide Lin, Bo Zhao, Qiaozhu Mei, and Jiawei Han. PET: a statistical model for popular events tracking in social communities. In *SIGKDD*, pages 929–938. ACM, 2010.

[95] Jun S Liu. The collapsed gibbs sampler in Bayesian computations with applications to a gene regulation problem. *JASA*, 89(427):958–966, 1994.

[96] Yue Lu and Chengxiang Zhai. Opinion integration through semi-supervised topic modeling. In *Proceedings of the 17th international conference on World Wide Web*, pages 121–130. ACM, 2008.

[97] Andrew L Maas and Andrew Y Ng. A probabilistic model for semantic word vectors. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2010.

[98] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[99] Bhaskara Marthi, Hanna Pasula, Stuart Russell, and Yuval Peres. Decayed MCMC iltering. In *UAI*, pages 319–326, 2002.

[100] Michael Mathioudakis and Nick Koudas. Twittermonitor: Trend detection over the twitter stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, pages 1155–1158, New York, NY, USA, 2010. ACM.

[101] Rishabh Mehrotra, Scott Sanner, Wray Buntine, and Lexing Xie. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 889–892. ACM, 2013.

[102] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*, pages 171–180. ACM, 2007.

[103] D. Metzler, S. Dumais, and C. Meek. Similarity measures for short segments of text. *Advances in Information Retrieval*, pages 16–27, 2007.

[104] Donald Metzler, Susan Dumais, and Christopher Meek. Similarity measures for short segments of text. In *Advances in Information Retrieval*, pages 16–27. Springer, 2007.

[105] David Mimno, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 262–272. Association for Computational Linguistics, 2011.

[106] Thomas Minka and John Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 352–359. Morgan Kaufmann Publishers Inc., 2002.

[107] Arjun Mukherjee and Bing Liu. Aspect extraction through semi-supervised modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 339–348. Association for Computational Linguistics, 2012.

[108] Claudiu Musat, Julien Velcin, Stefan Trausan-Matu, and Marian-Andrei Rizoiu. Improving topic evaluation using conceptual knowledge. In *IJCAI*, 2011.

[109] N. Naveed, T. Gottron, J. Kunegis, and A.C. Alhadi. Searching microblogs: coping with sparsity and document quality. In *CIKM*, pages 183–188. ACM, 2011.

[110] David Newman, Edwin V. Bonilla, and Wray Buntine. Improving topic coherence with regularized topic models. In *NIPS*, pages 496–504. 2011.

[111] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 100–108. Association for Computational Linguistics, 2010.

[112] Xingliang Ni, Xiaojun Quan, Zhi Lu, Liu Wenyin, and Bei Hua. Short text clustering by finding core terms. *Knowledge and information systems*, 27(3):345–365, 2011.

[113] K. Nigam, A.K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine learning*, 39(2):103–134, 2000.

[114] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[115] X.H. Phan, L.M. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *WWW*, pages 91–100. ACM, 2008.

[116] Owen Phelan, Kevin McCarthy, and Barry Smyth. Using Twitter to recommend real-time topical news. In *Proceedings of the third ACM conference on Recommender systems*, pages 385–388, NY, USA, 2009. ACM.

[117] Jay M Ponte and W Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281. ACM, 1998.

[118] Ana-Maria Popescu and Orena Etzioni. Extracting product features and opinions from reviews. In *Natural language processing and text mining*, pages 9–28. Springer, 2007.

[119] D. Ramage, S. Dumais, and D. Liebling. Characterizing microblogs with topic models. In *International AAAI Conference on Weblogs and Social Media*, volume 5, pages 130–137, 2010.

[120] Xiang Ren, Yujing Wang, Xiao Yu, Jun Yan, Zheng Chen, and Jiawei Han. Heterogeneous graph-based intent learning with queries, web pages and wikipedia concepts. *Proceedings of the 7th ACM international conference on Web search and data mining - WSDM '14*, pages 23–32, 2014.

[121] A. Ritter, C. Cherry, and B. Dolan. Unsupervised modeling of Twitter conversations. In *NAACL HLT*, pages 172–180. ACL, 2010.

[122] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494. AUAI Press, 2004.

[123] Ankan Saha and Vikas Sindhwani. Learning evolving and emerging topics in social media: a dynamic nmf approach with temporal regularization. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 693–702. ACM, 2012.

[124] Mehran Sahami and Timothy D Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th international conference on World Wide Web*, pages 377–386. ACM, 2006.

[125] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM, 2010.

[126] Ruslan Salakhutdinov and Geoffrey E Hinton. Replicated softmax: an undirected topic model. In *NIPS*, volume 22, pages 1607–1614, 2009.

[127] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, November 1975.

[128] Ruhi Sarikaya, Geoffrey E Hinton, and Anoop Deoras. Application of deep belief networks for natural language understanding. *IEEE Transactions on Audio, Speech and Language Processing*, 2014.

[129] Richard Socher, Yoshua Bengio, and Christopher D Manning. Deep learning for nlp (without magic). In *Tutorial Abstracts of ACL 2012*, pages 5–5. Association for Computational Linguistics, 2012.

[130] Wei Song, Yu Zhang, Ting Liu, and Sheng Li. Bridging topic modeling and personalized search. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1167–1175. Association for Computational Linguistics, 2010.

[131] Xiaodan Song, Ching-Yung Lin, Belle L Tseng, and Ming-Ting Sun. Modeling and predicting personal information dissemination behavior. In *SIGKDD*, pages 479–488. ACM, 2005.

[132] Yangqiu Song, Haixun Wang, Zhongyuan Wang, Hongsong Li, and Weizhu Chen. Short text conceptualization using a probabilistic knowledgebase. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 2330–2336. AAAI Press, 2011.

[133] Michael Speriosu, Nikita Sudan, Sid Upadhyay, and Jason Baldridge. Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of the First workshop on Unsupervised Learning in NLP*, pages 53–63. Association for Computational Linguistics, 2011.

[134] Nitish Srivastava, Ruslan R Salakhutdinov, and Geoffrey E Hinton. Modeling documents with deep boltzmann machines. *The Conference on Uncertainty in Artificial Intelligence*, 2013.

[135] Mark Steyvers. Probabilistic topic models. *Handbook of latent semantic analysis*, 2007.

[136] Mark Steyvers, Padhraic Smyth, Michal Rosen-Zvi, and Thomas Griffiths. Probabilistic author-topic models for information discovery. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 306–315. ACM, 2004.

[137] A. Strehl and J. Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *JMLR*, 3:583–617, 2003.

[138] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[139] Yee W Teh, David Newman, and Max Welling. A collapsed variational Bayesian inference algorithm for latent dirichlet allocation. In *NIPS*, pages 1353–1360, 2006.

[140] Ivan Titov and Ryan McDonald. A joint model of text and aspect ratings for sentiment summarization. *Urbana*, 51:61801, 2008.

[141] Ivan Titov and Ryan McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th international conference on World Wide Web*, pages 111–120. ACM, 2008.

[142] Hans Van Halteren. Writing style recognition and sentence extraction. In *Workshop on Text Summarization, DUC*. Citeseer, 2002.

[143] Hanna M Wallach. Topic modeling: beyond bag-of-words. In *ICML*, pages 977–984. ACM, 2006.

[144] Chong Wang, Bo Thiesson, Christopher Meek, and David M Blei. Markov topic models. In *International Conference on Artificial Intelligence and Statistics*, pages 583–590, 2009.

[145] Dingding Wang, Shenghuo Zhu, Tao Li, and Yihong Gong. Multi-document summarization using sentence-based topic models. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 297–300. Association for Computational Linguistics, 2009.

[146] Jiwei Li1 Sujian Li1 Xun Wang and Ye Tian1 Baobao Chang. Update summarization using a multi-level hierarchical dirichlet process model. In *COLING*, pages 1603–1618, 2012.

[147] Quan Wang, Zheng Cao, Jun Xu, and Hang Li. Group matrix factorization for scalable topic modeling. *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval - SIGIR '12*, page 375, 2012.

[148] Quan Wang, Jun Xu, Hang Li, and Nick Craswell. Regularized latent semantic indexing. *SIGIR*, pages 685–694, 2011.

[149] Xuerui Wang and Andrew McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433. ACM, 2006.

[150] Xuerui Wang, Andrew McCallum, and Xing Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 697–702. IEEE, 2007.

[151] Yu Wang, Eugene Agichtein, and Michele Benzi. TM-LDA: efficient online modeling of latent topic transitions in social media. In *SIGKDD*, pages 123–131, NY, USA, 2012. ACM.

[152] Xing Wei and W Bruce Adviser-Croft. Topic models in information retrieval. *Phd thesis*, 2007.

[153] Xing Wei and W Bruce Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185. ACM, 2006.

[154] Michael J Welch, Junghoo Cho, and Christopher Olston. Search result diversity for informational queries. In *Proceedings of the 20th international conference on World wide web*, pages 237–246. ACM, 2011.

[155] J. Weng, E.P. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *WSDM*, pages 261–270. ACM, 2010.

[156] Jianshu Weng and Bu-Sung Lee. Event detection in twitter. In *ICWSM*, 2011.

[157] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR*

conference on Research and development in informaion retrieval, pages 267–273. ACM, 2003.

[158] Yang Xu, Gareth JF Jones, and Bin Wang. Query dependent pseudo-relevance feedback based on wikipedia. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 59–66. ACM, 2009.

[159] Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. A biterm topic model for short texts. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1445–1456, 2013.

[160] Xiaohui Yan, Jiafeng Guo, Shenghua Liu, Xue-qi Cheng, and Yanfeng Wang. Clustering short text using ncut-weighted non-negative matrix factorization. In *CIKM*, pages 2259–2262, NY, USA, 2012. ACM.

[161] Xiaohui Yan, Jiafeng Guo, Shenghua Liu, Xueqi Cheng, and Yanfeng Wang. Learning topics in short texts by non-negative matrix factorization on term correlation matrix. In *Proceedings of the SIAM International Conference on Data Mining*, 2013.

[162] Yiming Yang, Tom Pierce, and Jaime Carbonell. A study of retrospective and on-line event detection. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 28–36. ACM, 1998.

[163] Hongzhi Yin, Bin Cui, Hua Lu, Yuxin Huang, and Junjie Yao. A unified model for stable and temporal topic detection from social media data. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pages 661–672. IEEE, 2013.

[164] H Yu, C Ho, Y Juan, and C Lin. Libshorttext: A library for short-text classification and analysis. Technical report, Technical Report. http://www. csie. ntu. edu. tw/˜ cjlin/ papers/libshorttext. pdf, 2013.

[165] Cheng Xiang Zhai, William W Cohen, and John Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 10–17. ACM, 2003.

[166] Aonan Zhang, Jun Zhu, and Bo Zhang. Sparse online topic models. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1489–1500. International World Wide Web Conferences Steering Committee, 2013.

[167] Min Zhang and Xingyao Ye. A generation model to unify topic relevance and lexicon-based sentiment for opinion retrieval. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 411–418. ACM, 2008.

[168] Zhenzhong Zhang, Le Sun, and Xianpei Han. Learning to detect task boundaries of query session. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1885–1888. ACM, 2013.

[169] W. Zhao, J. Jiang, J. Weng, J. He, E.P. Lim, H. Yan, and X. Li. Comparing Twitter and traditional media using topic models. *Advances in Information Retrieval*, pages 338–349, 2011.

[170] Wayne Xin Zhao, Jing Jiang, Hongfei Yan, and Xiaoming Li. Jointly modeling aspects and opinions with a maxent-lda hybrid. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 56–65. Association for Computational Linguistics, 2010.

[171] W.X. Zhao, J. Jiang, J. He, Y. Song, P. Achananuparp, E.P. Lim, and X. Li. Topical keyphrase extraction from Twitter. In *ACL*, pages 379–388, 2011.

[172] Y. Zhao and G. Karypis. Criterion functions for document clustering: Experiments and analysis. 2001.

[173] Jun Zhu and Eric Xing. Sparse topical coding. In *Proceedings of the Twenty-Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-11)*, pages 831–838, Corvallis, Oregon, 2011. AUAI Press.