计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2018.20180155
55(9):1972-1986,2018

一种基于社交事件关联的故事脉络生成方法

李莹莹[1,2] 马 帅[1,2] 蒋浩谊[1,2] 刘 喆[2] 胡春明[1,2] 李 雄[3]
[1](软件开发环境国家重点实验室(北京航空航天大学) 北京 100191)
[2](北京大数据科学与脑机智能高精尖创新中心(北京航空航天大学) 北京 100191)
[3](国家计算机网络应急技术处理协调中心 北京 100029)
(liyy@act.buaa.edu.cn)

An Approach for Storytelling by Correlating Events from Social Networks

Li Yingying[1,2], Ma Shuai[1,2], Jiang Haoyi[1,2], Liu Zhe[2], Hu Chunming[1,2], and Li Xiong[3]
[1](*State Key Laboratory of Software Development Environment* (*Beihang University*), *Beijing* 100191)
[2](*Beijing Advanced Innovation Center for Big Data and Brain Computing* (*Beihang University*), *Beijing* 100191)
[3](*National Computer Network Emergency Response Technical Team/Coordination Center of China*, *Beijing* 100029)

Abstract Social networks, such as Twitter and Sina weibo, have become popular platforms to report the public event. They provide valuable data for us to monitor events and their evolution. However, informal words and fragmented texts make it challenging to extract descriptive information. Monitoring the event progression from fast accumulation of microblogs is also difficult. To this end, we monitor the event progression with a common topic from the social network. This can help us to gain an overview and a detailed documentation of the events. In this paper, we use three consecutive components to meet this end. First, we use a structure based approach to detect events from the microblog dataset. Second, we cluster the events by their topics based on their latent semantic information, and define each cluster as a story. Third, we use a graph based approach to generate a storyline for each story. The storyline is denoted by a directed acyclic graph (DAG) with a summary to express the progression of events in the story. The user experience evaluation indicates that this method can help us to monitor events and their progression by achieving improved accuracy and comprehension compared with the state of art methods.

Key words social network; event progression; storyline; cluster; topic model

摘 要 推特和新浪微博等社交网络已成为报道公共事件的重要平台,它们为监控事件及其演化提供了宝贵的数据.然而,这些数据包含的非正式词语和碎片化文本使得从中提取描述性的信息具有一定的挑战.另外,从快速生成的大量微博监控事件演化也有一定难度.提出在社交网络中监控事件并对具有相同主题的事件演化进行分析.这既可以在粗粒度水平获得事件的概述,又可以在细粒度水平获得事件的详细信息.通过3个连续的组件实现该任务.1)用结构化的方法从微博检测事件;2)基于事件的隐式语义信息对事件聚类并将聚类获得的簇定义为故事;3)用基于图的方法为每个故事生成故事脉络,故事脉络用包含摘要的有向无环图表示故事内事件的演化.用户体验评估实验表明:提出的方法比现有方法具有更高的准确性和可理解性,并能够帮助用户监控事件及其演化.

收稿日期:2018-03-06;**修回日期**:2018-06-25
基金项目:国家自然科学基金项目(U1636210&61421003);国家自然科学基金优秀青年科学基金项目(61322207)
This work was supported by the National Natural Science Foundation of China (U1636210&61421003) and the National Natural Science Foundation of China for Excellent Young Scientists (61322207).
通信作者:马帅(mashuai@buaa.edu.cn)

关键词 社交网络;事件演化;故事脉络;聚类;主题模型

中图法分类号 TP391

社交网络已被政府、公司甚至总统(如奥巴马、特朗普等)等广泛用于发布新闻和报道事件.社交网络中信息的实时性和快速传播的能力使其成为获取信息的重要媒介.短文本的表述方式也能够有效地传递关键信息.社交网络的这些特性颠覆了传统媒体在信息传播上的统治力,这使其为监控事件及其演化提供了宝贵数据.然而,社交网络中文本的快速积累、口语化的表达方式以及文本内容中的错别字使得监控事件及事件间的演化具有极大挑战.从社交网络文本中对具有同一主题的的事件及其演化进行提取能够极大地帮助我们在全景上对某一事件进行了解.例如:我们期望获得关于平昌冬奥会所有项目(即事件)的信息和这些项目的进程(即事件演化).这需要我们首先检测事件,而后对这些事件进行聚类从而获得具有同一主题的事件(即故事),并最终以一种用户友好的方式(故事脉络)呈现出来.

目前针对该问题的方法按照是否需要用户提供关键词,大致可分为2类:1)关键词检索依赖型算法,将该问题形式化为信息检索问题,依据用户提供的关键词生成故事脉络,如MetroMap[1]首先依据用户提供的关键词匹配到相关的文档,然后用其构造用于表示故事脉络的多尺度地图.再如Wang等人[2]首先依据主题相关的包含文本描述的图像集合用图像的文本和时间相似度构造带权重的图,然后通过在该图上解决最小权重支配集(minimum-weighted connected dominating set)问题选择用于表示故事脉络的对象;再如GESM[3]首先依据用户提供的关键词得到相关的微博,然后依据Wang等人[2]的算法构造故事脉络.然而,这类方法严重依赖于用户所提供的关键词,而对于用户无法提供关键词的情况,这类算法无法提供相应的结果,这限制了该类方法的应用.2)为了解决这一问题,关键词检索独立型算法能够自动生成故事脉络,如CAST[4]首先从数据流中基于微博的文本相似度和时间相似度构造微博图,并将微博图中稠密子图做为事件,然后依据事件间的相似度构造事件间关系,依据事件间关系追踪事件的上下文.StoryGraph[5]则将每天的新闻文本分到不同主题集合中,然后通过新产生的主题与已经存在的主题的Pearson相关系数决定事件的演化.

然而,故事脉络生成仍然存在2个问题:1)事件由微博集合表示且有特定主题,如何从微博集合提取与事件对应的强相关的微博集合是一个关键问题.目前,针对该问题研究者们已经提出多种解决方式,然而如何选择最优的方法是一个具有挑战的问题.2)对有关联关系的事件如何进行有效组装,并以故事脉络的形式展示是另一个关键问题.

为此,我们将该问题形式化为3个连续的步骤,即事件检测、故事组装以及故事脉络生成.本文的主要贡献有3个方面:

1) 从微博检测事件.依据事件的隐式语义信息关联事件并组装故事,为故事生成故事脉络以可视化故事的发展过程;

2) 提出用包含摘要的有向无环图描述故事脉络.该故事脉络既可以使用户了解故事,也可使用户了解故事的发展过程;

3) 利用新浪微博数据集评价我们提出的故事脉络生成方法.基于用户体验的实验表明我们方法的性能优于现有方法.

1 研究问题和系统框架

在本节中,我们首先介绍术语的定义;然后,我们陈述所研究的问题;最后,我们描述系统框架.

1.1 术语定义

定义1. 微博.一个微博m由二元组$\langle M, T_m\rangle$表示,其中,1)M是微博的内容;2)T_m是微博的产生时间.

定义2. 事件.一个事件e是在某时间和地点发生的事件[6],例如:"正确的中国国旗赶制完成预计11日运抵里约"是一个事件.其由六元组(式)$\langle T_e, Microblog_set, C_e, L_e, P_e, D_e\rangle$表示.其中,1)$T_e$表示检测到事件的时间;2)$Microblog_set$表示事件的微博集合;3)$C_e$表示记录事件主要信息的核心词集合;4)$L_e$表示事件的地点;5)$P_e$表示事件的参与者集合;6)$D_e$表示事件的描述,该描述由一个短句子表示.我们基于微博集合$Microblog_set$识别L_e,P_e和D_e特征.

定义3. 故事.一个故事s定义为属于相同主题的事件集合,例如"2016里约奥运会"是一个故事,其由五元组(式)$\langle Event_set, T_s, C_s, L_s, P_s\rangle$表示.其中,1)$Event_set$表示故事的事件集合;2)$T_s$表示故

事的时间段;3)C_s 表示故事的核心词集合;4)L_s 表示故事的地点集合;5)P_s 表示故事的参与者集合. 我们基于故事的事件集合 $Event\_set$ 识别 T_s,C_s,L_s 和 P_s 特征.

定义 4. 故事脉络(*storyline*). 用于可视化故事的发展过程,其由二元组〈*skeleton*, *summary*〉表示,其中,1)*skeleton* 是展示故事内事件间演化的有向无环图;2)*summary* 是描述故事大意的短句子.

例 1. 图 1 中故事"2016 巴西奥运"的故事脉络(部分)用于可视化该故事的发展过程. 圆结点代表事件,事件的描述和检测时间(UTC+8)在该结点的右侧. 事件结点的索引号表示该事件在时间轴上的顺序,索引号越大表示事件的时间越靠后. 从事件结点 e_i 到事件结点 e_j 的有向边表示他们之间的时序演化关系. 该故事脉络有 3 个分支:分支 A、分支 B 和分支 C. 分支 A 与"巴西里约奥运俄罗斯部分运动员被禁赛"相关;分支 B 与"巴西里约奥运中国国旗有误"相关;分支 C 与"巴西里约奥运美国女子 4×100 接力掉棒"相关. 故事脉络中的多个分支展示了故事"2016 巴西奥运"的发展过程. 故事摘要(summary)展示在上方的矩形框里. 该摘要由各分支摘要合并而成,可帮助用户了解故事概述.

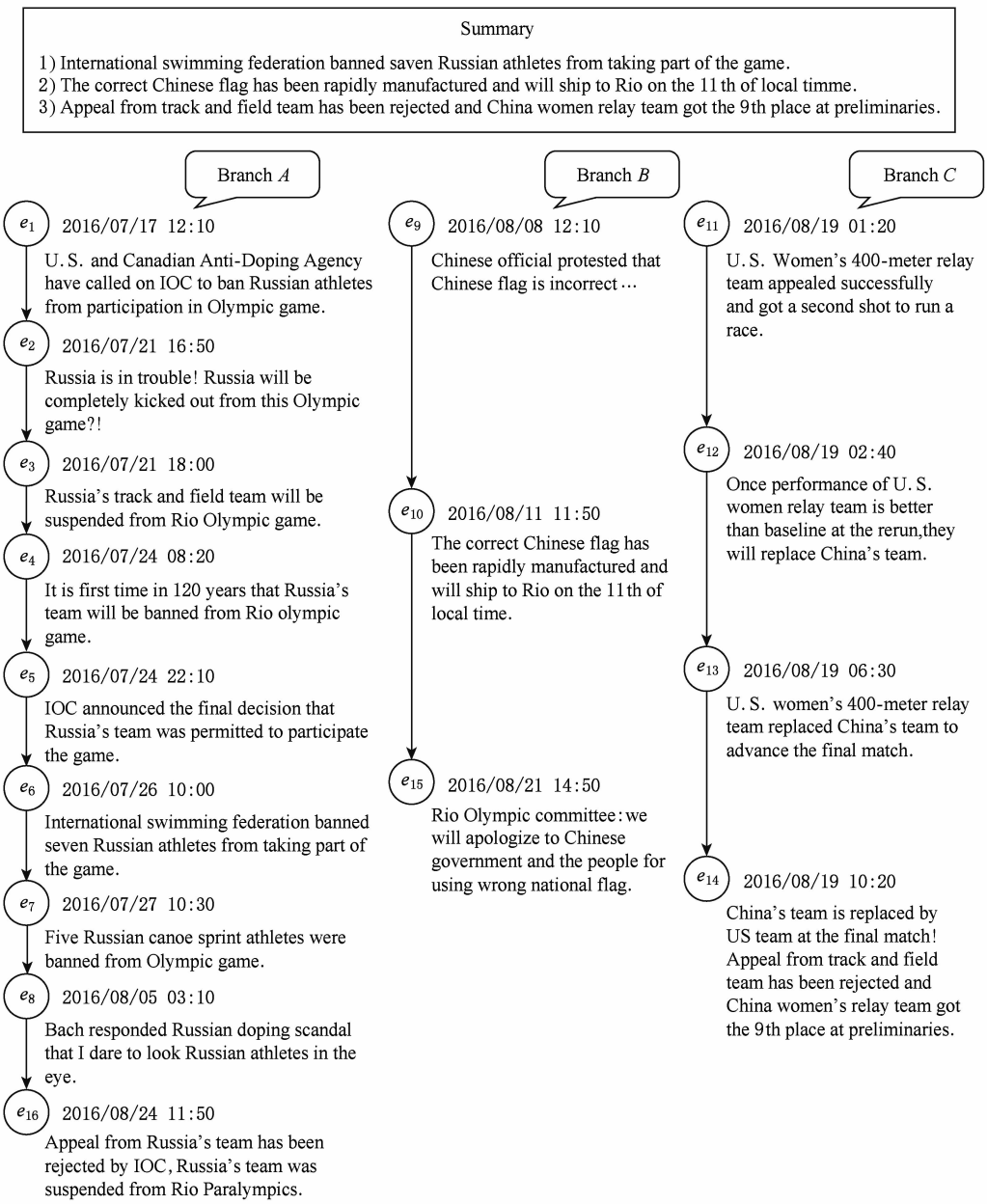


Fig. 1 The storyline in a story ("2016 Rio Olympic Games")

图 1 "2016 巴西奥运"故事的故事脉络

1.2 问题陈述

对于微博集($\{M_1, M_2, \cdots, M_t\}$),其中 M_t 是时间片 t 的微博集合.我们的目标是:1)从微博集中检测事件($\{E_1, E_2, \cdots, E_t\}$),其中 E_t 是时间片 t 检测的事件集合;2)依据事件的隐式语义信息有效的关联事件并组装故事($S=\{s_1, s_2, \cdots, s_{N_s}\}$),其中 s_i 表示一个故事;3)为每个故事生成一个用于可视化故事发展过程的故事脉络.

1.3 系统框架描述

我们用包含 3 个组件的框架(如图 2 所示)解决故事脉络生成问题.首先,我们从微博集中检测事件;然后,我们通过关联事件组装故事;最后,我们为每个故事生成描述故事发展过程的故事脉络.

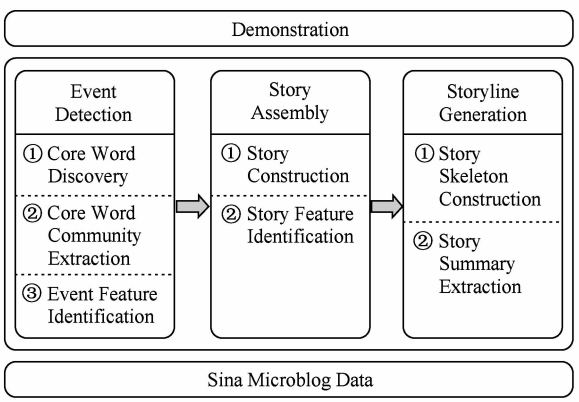


Fig. 2 System framework

图 2 系统框架

1.3.1 事件检测

我们从微博集中检测事件.首先,从微博集得到由表示事件的核心词和核心词间共现关系构成的核心词图;然后,发现核心词图中紧密连接的子图并将子图做为事件的核心词集合;最后,为事件识别其他的特征 T_e, L_e, P_e, D_e 和 $Microblog\_set$.

1.3.2 故事组装

我们依据主题对事件分组,将事件组装成故事.首先,我们依据事件的隐式语义信息对事件聚类,将一个簇认为一个故事;然后,我们为每个故事识别其他的特征 T_s, L_s, P_s 和 C_s.

1.3.3 故事脉络生成

我们为每个故事生成故事脉络,该故事脉络由包含摘要的事件有向无环图表示.首先,我们从故事的事件集基于弱连通分量和最大生成树构造有向无环图($skeleton$);然后,我们基于故事的所有事件描述提取短文本作为故事的摘要.

2 系统组件

针对在第 1 节中形式化的 3 个步骤,即事件检测、故事组装和故事脉络生成,我们在本节具体介绍所对应的实现方法.

2.1 事件检测

在事件检测步骤,我们旨在从微博数据中检测事件.为帮助用户理解故事的发展过程,我们认为故事中的事件应该可以使用户剖析故事的细节,即事件应属于特定主题且具有细粒度性.

表示事件的词、核心词,在使用频率和与其他词的共现模式上较于该词的历史时刻有异常的变化[7].单个核心词表示的事件粒度较粗,不足以表达事件的全部信息.例如单个核心词、辅警,只能表示该事件与辅警有关.紧密连接的核心词集合可以详细地表达事件信息,增加事件内容覆盖率.例如,紧密连接的核心词集合,江苏省、沭阳、追授、牺牲和辅警,可详细表述"江苏省政府追授沭阳因公牺牲辅警孙孟涛见义勇为英雄称号"事件.用核心词集合表示的事件不利于用户理解讲述的内容,我们用事件的结构化表示帮助用户理解事件.

我们用 3 个连续的模块完成事件检测任务.首先,用热点发现算法[7]发现表示事件的具有异常出现频率的词(核心词);然后,用重叠的社区检测算法[8]提取紧密连接的核心词集合对事件进行详细描述;最后,从微博识别事件其余特征,方便用户理解事件.下面描述的 3 个连续的模块完成事件检测任务,我们采用 Ring[9]实现的事件检测算法.

2.1.1 核心词发现

在核心词发现阶段,我们发现表示事件的核心词.表示事件的词在使用频率和与其他词的共现模式上较于该词的历史时刻有异常变化[7].我们用 HOTSPOT[7]检测能描述事件的词.该算法首先依据微博数据构造词共现图;然后检测有异常变化的词,即核心词,并输出核心词及核心词间共现关系构成的图(核心词图).

2.1.2 核心词社区提取

在核心词社区提取阶段,我们提取表示事件的核心词集合.事件的核心词通常紧密连接.依据上一步输出的核心词图,我们用社区检测算法[8]检测紧密连接的核心词社区,即由词(点)和词间共现关系(边)构成的稠密子图.核心词社区对应事件的核心词集合,其能够有效地描述一个事件.我们依据图 3(a)

展示的核心词图检测出图 3(b)的核心词社区,该核心词社区表示“江苏省政府追授沭阳因公牺牲辅警孙孟涛见义勇为英雄称号”事件.

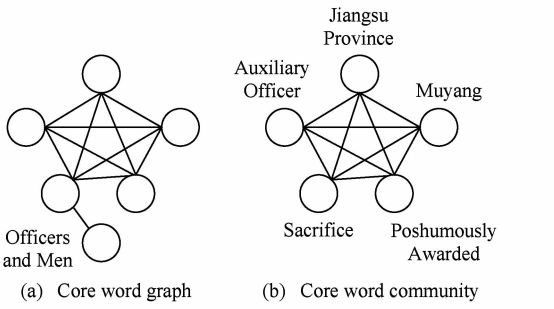


Fig. 3 A core word community in the core word graph

图 3 核心词图的一个核心词社区

2.1.3 事件特征识别

在事件特征识别阶段,我们将事件的数据进行结构化以增加事件的描述信息. 仅用核心词集合表示事件存在不足,如碎片化和易读性差. 我们将用核心词集合表示的事件扩充为事件六元组,过程如下:

1) 我们将时间 T_e 赋值为事件被检测的时间(每 10 min);

2) 我们依据核心词集合寻找包含事件所有核心词的微博集合 *Microblog_set*;

3) 我们将描述 D_e 赋值为事件的微博集合中包含核心词集合中的词最多的句子;

4) 我们从事件的微博集合中识别所有的命名实体(named entity),包括地名、人名和机构名等;

5) 我们将地点 L_e 赋值为事件的微博集合中最频繁出现的地名;

6) 我们将参与者集合 P_e 赋值为事件的微博集合中出现的人名和机构名.

2.2 故事组装

在故事组装步骤,我们旨在通过关联事件组装故事. 为帮助用户从全景了解故事,故事应囊括该主题下所有事件,即故事组装需有效组装有关联的事件.

依据事件词的相似度,即事件的显式语义信息,将有关联关系的事件聚成簇是简单直观的方式. 但基于显式语义信息聚类得到的簇粒度较细,即只能将词相似度较高的事件聚到相同的簇. 考虑到相同故事的事件可能包含较少的共有词,如表 1 所示,事件 e_{10} 和事件 e_{11} 仅包含“Rio”和“Olympic”两个共有词,基于显式语义信息的聚类不能有效组装有关联关系的事件. Latent Dirichelet Allocation(LDA)为数据集中的数据,例事件集合中的事件,生成有利于相似性和相关性判断的主题分布[10]. 我们通过用 LDA 生成事件的主题分布发现事件 e_{10} 和事件 e_{11} 的主题分布很相似. 为方便说明,以下称事件的主题分布为隐式语义信息. 我们基于 LDA 挖掘的隐式语义信息将相同主题的事件聚成簇.

Table 1 Two Eevents From the Story (“2016 Rio Olympic Games”)

表 1 故事“2016 巴西奥运”中 2 个事件

Feature	e_{10}	e_{11}
Time	2016/08/11 11:50	2016/08/19 01:20
Location	Brazil	US
Participants	Chinese Olympic Committee, the Xinhua News Agency	The US team, Brazilian team
Core words	protest, rapidly manufacture, ship to	playback, take-over, appeal, American team, Felix
Description	The correct Chinese flag has been rapidly manufactured and will ship to Rio on the 11th of local time.	US women 400-meter relay team of appealed successfully and got a second shot to run a race.
Microblog Set	M1: The correct Chinese flag has been rapidly manufactured and will ship to Rio on 11th. According to Xinhua News Agency, Chinese officials repeatedly protested incorrect flag was used during medal ceremonies and Rio Olympic finally agreed to rapidly manufacture Chinese flag. contractor, located in Sao Paulo, spent 30 hours to complete the task and will ship the flags to Rio on 11th. M2: …	M1: # Rio Olympic Go for it # [The women 400-meter relay team of US granted the rerun.] US team has filed an appeal claim that US athlete was bumped by a Brazilian athlete and lead to drop baton. After taking the appeal from US team, Brazilian Olympic officials disqualified the Brazilian teams and granted the second chance to ran a race alone at night to decide whether they can advance to final. M2: …

我们用 2 个连续的模块完成故事组装任务. 首先,我们依据事件的隐式语义信息关联事件,将事件分到不同的故事;然后,我们依据事件的特征,识别故事的特征,生成故事的结构化表示,以便用户查询.

2.2.1 故事构造

在故事构造阶段,我们将含有相同主题的事件聚成簇,称为故事. LDA 生成的隐式语义信息包含相似性和相关性判断等任务的必要统计关系[10],是

一个有效且被很多学者使用的模型.我们用 LDA 建模事件所属的故事.每个事件 e_i 被建模成故事(主题)的概率分布,用故事向量($s_1^i, s_2^i, \cdots, s_{N_s}^i$)表示.其中,$s_k^i$ 表示事件 e_i 属于故事(主题)s_k 的概率,N_s 是参数初始故事数.观察发现,较于不相关的事件,相同故事下的事件有更多的共有词.用该先验知识初始化 LDA 中故事的词分布可减少 LDA 的搜索空间.

我们用预聚类和细聚类的方式组装故事.首先,我们用聚类算法 DBSCAN[11] 实现预聚类,即依据显式语义信息对事件分组;然后,我们用预聚类的结果初始化 LDA 中故事的词分布,并依据 LDA 生成的隐式语义信息构造故事.

1) 预聚类.预聚类依据事件的显式语义信息对事件分组.目前有很多成熟且应用广泛的聚类算法.我们从成熟聚类算法中选择适合我们任务的算法.基于密度的聚类算法 DBSCAN 有 3 个优势:①能处理带噪音的数据;②不需要指定类别;③容易适应单遍(single-pass)聚类,即只需遍历一遍数据集即可完成聚类.我们采用 DBSCAN 进行预聚类.

首先,我们为事件集合 E 中每个事件 e 构造词向量 $\boldsymbol{w}_e$.若第 k 个词在事件 e 中,$w_{e,k}=1$;否则 $w_{e,k}=0$.然后,我们依据词向量用 DBSCAN 将事件聚到类成员 P 中,其中 $P=\{P_1, P_2, \cdots, P_I\}$,$P_i$ 是包含一个事件集合的预簇.DBSCAN 使用的距离函数:

$$dis(e_i, e_j)=1-\cos(\boldsymbol{w}_{e_i}, \boldsymbol{w}_{e_j}), \quad (1)$$

其中,$\boldsymbol{w}_{e_i}$ 和 $\boldsymbol{w}_{e_j}$ 分别是事件 e_i 和事件 e_j 的词向量.

最终,我们将事件集合 E 和基于 DBSCAN 的聚类结果作为细聚类的输入.

2) 细聚类.细聚类基于预聚类的结果挖掘事件的隐式语义信息,依据事件的隐式语义信息关联事件,并将事件赋值到故事.LDA 生成的隐式语义信息有利于相关性判断.用预聚类的结果初始化 LDA 中故事的词分布可减少 LDA 的搜索空间.

首先,我们依据预聚类结果初始化 LDA 中故事的词分布,给定预聚类结果 P,我们将相同的预簇中事件的词赋给相同的故事;然后,我们用 Gibbs Sampling 推断 LDA 的参数、事件的故事向量;最后,我们依据选择标准将事件赋给故事.

选择标准.我们假设每个事件属于且仅属于一个故事.我们将事件赋给概率最高的故事.

算法 1. 故事构造算法.

输入:事件集合 $E=\{e_1, e_2, \cdots, e_n\}$、初始故事数 N_s;

输出:故事集合 $S=\{s_1, s_2, \cdots, s_m\}$($m \leqslant N_s$).

```
Construct. Story (E, N_s);
① S←{s_1, s_2, ···, s_{N_s}};
② {P_1, P_2, ···, P_I}←DBSCAN(E);
③ for i=1 to I do
④   if i≤N_s then
⑤     k←i;
⑥   else
⑦     k←random(1, N_s);
⑧   end if
⑨   将预簇 P_i 中所有事件的所有词赋给故事 s_k 的词列表;
⑩ end for
⑪ for iter=1 to N_iter do
⑫   for each event e∈E do
⑬     for each word w∈e do
⑭       for each story s∈S do
⑮         计算事件 e 中词 w 属于故事 s 的概率;
⑯       end for
⑰       基于词 w 的故事概率分布抽样词所属的故事;
⑱     end for
⑲   end for
⑳ end for
㉑ for each event e∈E do
㉒   为事件 e 计算故事向量;
㉓ end for
㉔ 基于选择标准将事件赋给故事;
㉕ 移除故事集合 S 中空故事;
㉖ return S.
```

故事构造的伪代码如算法 1 所示,给定事件集 E,故事构造算法构造并返回故事集 S.首先,我们用聚类算法 DBSCAN 预聚类(行②);然后,我们用 DBSCAN 的预聚类结果初始化 LDA(行③~⑩);随之,我们用 Gibbs Sampling 推断 LDA 的参数,包括推断事件的故事向量(行⑪~㉓);而后,我们依据选择标准将事件分到故事中并去掉不包含事件的故事(行㉔~㉕);最后,我们返回非空的故事集 S(行㉖).

时间复杂度分析.DBSCAN 需计算任意 2 个事件间的距离,这需要 $O(|E|^2)$.LDA 需为各事件的各词抽样故事,这需要 $O(N_s|E|\bar{l})$,其中 $\bar{l}$ 是事件中词的平均长度.总时间复杂度为 $O(|E|^2+N_s|E|\bar{l})$.

2.2.2 故事特征识别

在故事特征识别阶段,我们将故事的数据进行结构化以便于用户查询故事.事件检测组件为事件生成结构化表示,用事件的结构化表示生成故事的结构化表示既能充分利用相关微博的信息,也可以提高效率.我们基于事件六元组将用事件集合表示的故事扩充为故事五元组,过程如下:1)故事的时间段 T_s 的开始时间和结束时间分别被赋值为故事的事件集中事件的最早时间和最晚时间;2)故事内包含事件的特征越多,越能帮助用户查询故事,故事的地点集合 L_s、参与者集合 P_s 和核心词集合 C_s 分别被设为故事的事件集中相应特征的并集.

算法 2. 故事特征识别算法.

输入:故事集合 $S=\{s_1,s_2,\cdots,s_m\}$;

输出:故事集合 $S=\{s_1,s_2,\cdots,s_m\}$.

Identify. Story. Feature (S);

① for each story $s\in S$ do

② $T_s.start\leftarrow\min(\{T_e|e\in Event\_set_s\})$;

③ $T_s.stop\leftarrow\max(\{T_e|e\in Event\_set_s\})$;

④ $L_s\leftarrow\bigcup_{e\in Event\_set_s}L_e$;

⑤ $P_s\leftarrow\bigcup_{e\in Event\_set_s}P_e$;

⑥ $C_s\leftarrow\bigcup_{e\in Event\_set_s}C_e$;

⑦ end for

⑧ return S.

故事特征识别的伪代码如算法 2 所示.给定故事集 S,故事特征识别算法为故事集 S 中每个故事识别特征并返回故事集 S.首先,故事的开始时间被设为事件集中的事件的最早时间(行②),故事的结束时间被设为事件集中事件的最晚时间(行③);然后,故事的地点集合、参与者集合和核心词集合分别被设为事件集中地点、参与者集合和核心词集合的并集(行④~⑥);最后,故事集 S 被返回(行⑧).

时间复杂度分析.为故事生成特征时,我们需遍历该故事的事件集,这需要 $O(\bar{l}_e)$,其中 $\bar{l}_e$ 指故事中事件的平均长度.因此,总的时间复杂度为 $O(|S|\bar{l}_e)$.

2.3 故事脉络生成

在故事脉络生成步骤,我们旨在为故事生成包含摘要的有向无环图以可视化故事的发展过程.为有更好的用户体验,故事脉络应兼顾准确性和理解性.准确性指故事脉络准确地展示事件的发展过程.理解性指故事脉络便于用户快速的了解故事.

故事可能包含多个相对独立的部分.例"2016巴西奥运"故事包含"巴西里约奥运俄罗斯部分运动员被禁赛"和"巴西里约奥运美国女子 4×100 接力掉棒"等多个相对独立的部分.我们用弱连通分量提取故事中多个相对独立的部分.为方便描述,下面称相对独立的部分为分支.

分支内的事件有较强的关联关系.复杂的图结构表示的分支不便于用户快速的理解[15].如图 4(a)中图结构表示的分支,虽然其充分地表达了事件间的关联关系,但其也引入一些不必要连接,如事件 e_3 到事件 e_7.为折中准确性和理解性,我们用最大生成树生成分支的树结构,如图 4(b)所示.

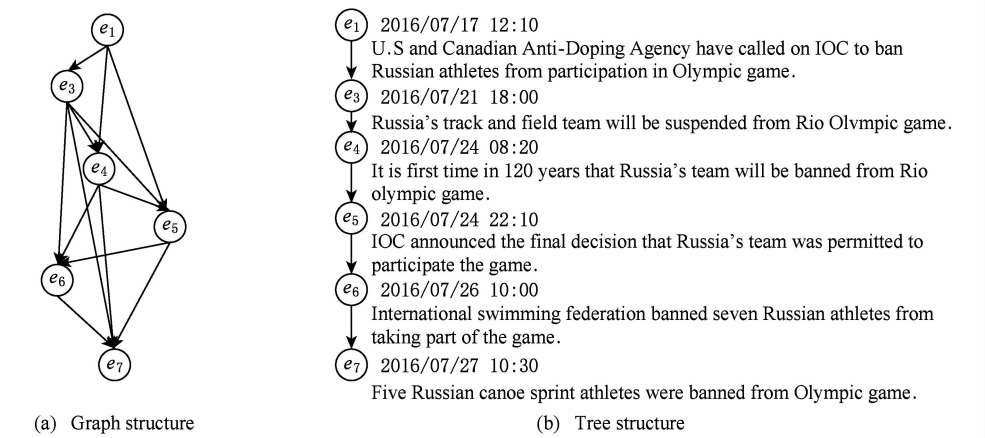


Fig. 4 A branch represented by a graph or tree structure

图 4 由图或树结构表示的分支

我们用 2 个连续的模块完成故事脉络生成任务.首先,我们从故事的事件集中基于弱连通分量和最大生成树构造故事骨架;然后,我们用基于图的方法提取短文本做为故事的摘要.

2.3.1 故事骨架构造

在故事骨架构造阶段,我们依据故事的事件集构造用于描述故事发展过程的有向无环图.首先,我们计算任意 2 事件间的权重,依此生成有向边,构造

一个事件图;然后,我们依据事件图识别故事中的分支,即识别该图中所有的弱连通分量,并形成弱连通分量集合;最后,我们为弱连通分量集合中每个弱连通分量构造一个最大生成树,即用树结构表示的分支.这些用树结构表示的分支构成故事的骨架:

$$w(e_i,e_j)=I(T_{e_i},T_{e_j})sim_l(e_i,e_j)\times(c_p sim_p(e_i,e_j)+c_c sim_c(e_i,e_j)),\quad(2)$$

其中,e_i 和 e_j 表示 2 个事件,$I(T_{e_i},T_{e_j})$表示事件间的时间关系;sim_l,sim_p 和 sim_c 表示 2 事件地点、参与者集合和核心词集合的相似度;c_p 和 c_c 是权重系数,该权重系数在满足 $c_p+c_c=1$ 的条件下可以被调整.

演化关系包含着事件的时间关系.有向边只能从先发生的事件指向后发生的事件.若 $T_{e_i}<T_{e_j}$,$I(T_{e_i},T_{e_j})=1$;在其他情况下,$I(T_{e_i},T_{e_j})=0$.

相同地点发生的事件更可能属于相同的分支.sim_l用于度量 2 事件地点间的相似度.$sim_l(e_i,e_j)=1$,若事件 e_i 和事件 e_j 的地点相同;$sim_l(e_i,e_j)=0.5$,若事件 e_i 的地点地理位置上属于事件 e_j 的地点,例如地点“中国北京”在地理位置上属于地点“中国”;$sim_l(e_i,e_j)=0$,在其他情况下.

事件间的参与者和核心词的相似度同样能反映事件间的演化关系.$sim_p(e_i,e_j)$度量 2 事件的参与者集合的 Jaccard 系数,$sim_c(e_i,e_j)$度量 2 事件的核心词集合的 Jaccard 系数.

事件的微博集合由包含事件所有核心词的微博构成.事件的地点和参与者由微博集合中的命名实体构成.因此事件的核心词、地点和参与者包含了微博集合的主要信息.

我们在 3 个组装的故事上调节权重系数 c_p 和 c_c.首先,我们使用多组权重系数构造故事骨架.然后,我们依据骨架是否反应故事的发展过程对多个故事骨架排序,并依据排序结果设定 $c_p=0.3$ 和 $c_c=0.7$.

算法 3. 故事骨架构造算法.

输入:故事 s 的事件集 $Event\_set=\{e_1,e_2,\cdots,e_{|Event\_set|}\}$;

输出:故事的骨架 *skeleton*.

Construct. Story. Skeleton(*Event_set*);

① 基于 *Event_set* 创建一个按时间升序排列的事件列表 *event. list*;
② *skeleton*←null;
③ *event2branch*←null; /* 事件到分支映射 */
④ for $i=0$ to *event. list. size*−1 do
⑤ *event. parent*←null; /* 父事件结点 */
⑥ *edge. weight*←0; /* 与父事件结点边的权重 */
⑦ for $j=0$ to $i-1$ do
⑧ *j2 i. weight*←compute. weight(*event. list*,*j*,*i*); /* 依据式(2)计算事件 *j* 到 *i* 的有向边权重 */
⑨ if *j2i. weight*>*edge. weight* then
⑩ *event. parent*←*event. list. get*(*j*);
⑪ *edge. weight*←*j2i. weight*;
⑫ end if
⑬ end for
⑭ if *event. parent*≠null then
⑮ *branch*←*event2branch. get*(*event. list. get*(*j*));
⑯ *branch. add*(edge(*event. list*,*j*,*i*));
⑰ *event2branch. put*(*event. list. get*(*i*), *branch*);
⑱ else
⑲ 创建一个新的分支 branch;
⑳ *branch. add*(*event. list. get*(*i*));
㉑ *event2branch. put*(*event. list. get*(*i*), *branch*);
㉒ *skeleton. add*(*branch*);
㉓ end if
㉔ end for
㉕ return *skeleton*.

故事骨架构造的伪代码如算法 3 所示.给定故事 s 的事件集 *Event_set*,算法 3 为故事 s 构造并返回故事骨架 *skeleton*.算法计算弱连通分量的同时构造弱连通分量的最大生成树.首先,我们依据事件的时间升序排列事件(行①).然后,我们遍历事件(行④~㉔).我们计算事件 *event* 与任意时间在 *event* 之前的事件间的有向边权重,并寻找最大的权重和对应的父事件 *event. parent*(行⑤~⑬).若存在父事件,则事件 *event* 属于事件 *event. parent* 所在的分支,并在分支中添加从事件 *event. parent* 到事件 *event* 的边(行⑮~⑰),否则,构造新的只包含事件 *event* 的分支 *branch*(行⑲~㉒).最后,我们返回故事骨架 *skeleton*(行㉕).

时间复杂度分析.升序排列事件花费的时间为 $O(|Event\_set|\,\mathrm{lb}(|Event\_set|))$.构造弱连通分量集和最大生成树需计算任意 2 事件间有向边权重,花费的时间为 $O(|Event\_set|^2)$.总花费时间为 $O(|Event\_set|^2)$.

2.3.2 故事摘要提取

在故事摘要阶段,我们依据故事的事件集为故事提取便于用户了解故事概述的摘要.为使用户从摘要了解各分支内容,故事摘要应包含各分支内容.事件描述便于用户理解事件,因此,我们基于故事的事件集提取几个事件描述作为故事摘要.首先,我们用 TextRank[12] 为各分支提取摘要;然后,我们将各分支的摘要合并为故事摘要.

算法 4. 故事摘要提取算法.

输入:故事骨架 *skeleton*;

输出:故事摘要 *story_summary*.

Extract. Story. Summary (*skeleton*);

① *story_summary*←null;

② for each *branch*∈*skeleton* do

③ *branch _description*←*merge. description* (*branch*); /* 将分支内所有的事件描述合为分支描述 */

④ *branch _summary*←*TextRank* (*branch _description*);

⑤ *story_summary. add*(*branch_summary*);

⑥ end for

⑦ return *story_summary*.

故事摘要提取的伪代码如算法 4 所示.给定故事 s 的故事骨架 *skeleton*,故事摘要提取算法为故事 s 提取并返回故事摘要 *story_summary*.我们生成各分支摘要,并将各分支摘要合并成故事摘要(行②~⑥).首先,我们将分支 *branch* 中所有的事件描述合并为分支描述 *branch _description*(行③),并用 TextRank 从文章 *branch_description* 中提取分支摘要 *branch_summary*(行④);然后,我们将分支摘要 *branch _summary* 合并到故事的摘要 *story _summary* 中(行⑤),并将其返回(行⑦).

时间复杂度分析.为一个分支提取摘要花费 $O(|\bar{b}|^2)$,其中 $|\bar{b}|$ 是分支的事件数.为所有 n_b 个分支提取摘要花费 $O(n_b\times|\bar{b}|^2)$.

3 实验与结果

针对第 2 节中提出的方法,我们在本节进行实验验证.首先,我们介绍实验设置;然后,我们评价事件检测、故事组装和故事脉络生成 3 个组件的性能,并展示我们提出的方法较于已有方法的优势.

3.1 实验设置

实验运行在 2 个 Intel Xeon E5-2650 v3 CPUs,64 GB 内存的机器(64 b Windows7 旗舰版系统)上.新浪微博数据集包含从 2016-06-01—2016-08-31 的共 2.16 亿条微博.我们将时间片设为 10 min,即每 10 min 检测一次事件.在该微博集我们共检测 19.8 万个事件.

3.2 事件检测实验结果及分析

本节主要评价事件检测的性能.首先,我们构造测试集;然后,我们评价 2 个事件检测算法 Ring[9] 和 MetroMap[1] 的性能.

算法 MetroMap 依据微博构造词共现图,基于词共现图用社区检测算法检测紧密连接的词社区,用词社区表示事件.

我们提出的框架(事件检测、故事组装和故事脉络生成)需要关联事件.框架是否合理依赖检测的事件间是否存在关联.事件核心词集合的重合度可以反映事件间的关联性.我们使用冗余度指数 *redundancy-ratio* 计算事件集 E 含有关联事件的事件所占的百分比:

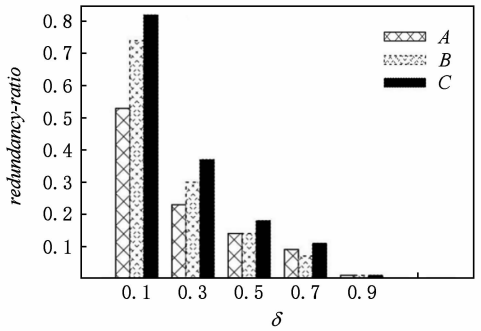
$$redundancy\text{-}ratio(E,\delta)=\frac{\sum_{e_i\in E} I(e_i,E,\delta)}{|E|},\quad(3)$$

其中,E 表示事件集;δ 是 0 到 1 间的实数,表示阈值;e_i 表示事件;$I(e_i,E,\delta)$是示性函数,表示事件集 E 是否存在与事件 e_i 相关联的事件.若事件集 E 存在事件 $e_j(e_i\neq e_j)$,且 $\cos(coreword_{e_i},coreword_{e_j})>\delta$,则 $I(e_i,E,\delta)=1$;否则 $I(e_i,E,\delta)=0$.

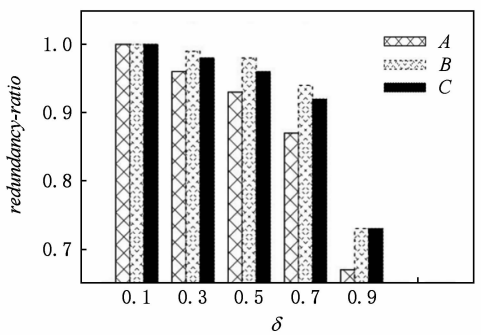
事件检测使用的数据集包含 2016-08-11—2016-08-13 共 3 d 780 万条微博.我们用该数据集构造 3 个测试集.其中,测试集 A 由 2016 年 8 月 13 日的微博构成,测试集 B 由 2016-08-11—2016-08-12 的微博构成,测试集 C 由 2016-08-11—2016-08-13 的微博构成.

Ring 和 MetroMap 的冗余度指数如图 5 所示.在测试集 C 上,当阈值 $\delta=0.1$ 时,Ring 的冗余度指数大于 82%,MetroMap 的冗余度指数是 100%.Ring 和 MetroMap 检测的大部分事件至少有一个关联事件.这说明我们提出的自底向上的框架(事件检测、故事组装和故事脉络生成)的合理性.

在图 5(b)中,MetroMap 的冗余度指数随着阈值的增加轻微地减小,这说明 MetroMap 检测的事件存在大量重复事件.在图 5(a)中,Ring 的冗余度指数对阈值很敏感,这说明较于 MetroMap,Ring 检测到事件的多样性更好.同时,事件可能与另一天的事件相关联.因此,当数据集变大时,冗余度指数也变大.



(a) *redundancy-ratio* of Ring



(b) *redundancy-ratio* of MetroMap

Fig. 5 *redundancy-ratio* at different datasets and thresholds

图 5 不同数据集和阈值上的冗余度指数

3.3 故事组装实验结果及分析

本节主要评价故事组装的性能.首先,我们构造用于评价的故事集,即金标准;然后,我们利用金标准评价我们的故事组装算法、LDA[10]、BTM[13]、GSDMM[14]、DBSCAN[11]和 Story Forest[15]的性能.

1) LDA.首先,该方法用主题模型 LDA 生成事件的主题分布;然后,该方法依据选择标准将事件赋给主题,一个主题对应一个故事.

2) BTM.首先,该方法用主题模型 BTM 生成事件的主题分布;然后,该方法依据选择标准将事件赋给主题,一个主题对应一个故事.

3) GSDMM.首先,该方法用主题模型 GSDMM 生成事件的主题分布;然后,该方法依据选择标准将事件赋给主题,一个主题对应一个故事.

4) DBSCAN.该方法用 DBSCAN 聚类,一个簇对应一个故事.事件间的距离用 1 减去事件间词的 cosine 值表示.

5) Story Forest.该方法依据事件核心词与故事核心词的 Jaccard 系数判定事件是否属于某故事.

构造金标准.我们请 2 个志愿者将事件检测算法检测的 19.8 万个事件分组,一个组对应一个故事.首先,一个志愿者对 2016-06-01—2016-07-15 的事件分组,另一个志愿者对 2016-07-16—2016-08-31 的事件分组.然后,一个志愿者查看另一个志愿者的分组结果,被 2 个志愿者认可的分组结果才会被保留.最后,为分析有演化过程的故事,我们移除包含 4 个及以下事件的故事.最终我们构造了共包含 1 011 个事件的 41 个故事.

我们使用已有的评价方法[16-17]评价故事组装的性能.我们将人工标注的故事称为金标准故事,将故事组装算法组装的故事称为组装故事.对任意一个金标准故事 g,我们计算该金标准故事与任意组装故事 a 的相似度,并将有最高相似度的组装故事 a_g 映射到金标准故事 g:

$$\mathrm{sim}(g,a)=\frac{|Event\_set_g \cap Event\_set_a|}{\sqrt{|Event\_set_g|\times|Event\_set_a|}}, \quad (4)$$

其中,$\mathrm{sim}(g,a)$是金标准故事 g 和组装故事 a 的相似度;$Event\_set_g$ 是金标准故事 g 的事件集;$Event\_set_a$ 是组装故事 a 的事件集;$|\cdot|$代表集合中元素的个数;$\cap$代表集合的交集.

然后,我们计算 $F1$ 值:

$$P=\frac{\sum_{g\in G}|Event\_set_g \cap Event\_set_{a_g}|}{\sum_{g\in G}|Event\_set_{a_g}|}, \quad (5)$$

$$R=\frac{\sum_{g\in G}|Event\_set_g \cap Event\_set_{a_g}|}{\sum_{g\in G}|Event\_set_g|}, \quad (6)$$

$$F1=\frac{2\times P\times R}{P+R}, \quad (7)$$

其中,g 是金标准故事,$Event\_set_g$ 是金标准故事 g 的事件集.a_g 是映射到金标准故事 g 的组装故事,$Event_set_{a_g}$是组装故事 a_g 的事件集,G 是金标准.

3.3.1 参数调节

我们调节 6 种方法:我们的故事组装算法、LDA、BTM、GSDMM、DBSCAN 和 Story Forest,调整参数的方式为:

1) LDA,BTM 和 GSDMM.我们在标注故事集上调节 3 个方法的参数,$alpha$,$beta$ 和初始故事数 N_s.首先,我们固定 $beta$ 和 N_s,从 0.1~1 之间,以 0.1 为步长调节 $alpha$,并取得最优值;然后,我们选择取得最优值的 $alpha$ 并固定 N_s,从 0.01~0.1 之间,以 0.01 为步长调节 $beta$,并取得最优值;最后,我们选择取得最优值的 $alpha$ 和 $beta$,从 50~500 之间,以 50 为步长调节 N_s,并选择取得最优值的 N_s.当 2 个参数设置取得相似结果时,我们参考文献报道的 BTM 和 GSDMM 中所使用的参数设置.

2) 本文方法. 我们使用最优的 LDA 配置($alpha=0.1, beta=0.03$),并用网格搜索调节初始故事数 N_s、最小点数 $minpts$ 和半径 $radius$ 参数. 其中 $N_s \in \{50,100,150,200,250,300,350,400,450,500\}$, $minpts \in \{2,3,4\}$, $radius \in \{0.6,0.65,0.7,0.75,0.8\}$.

3) DBSCAN. 我们用网络搜索调节最小点数 $minpts$ 和半径 $radius$ 参数. 其中 $minpts \in \{2,3,4\}$, $radius \in \{0.6,0.65,0.7,0.75,0.8\}$.

4) Story Forest. 我们从 0.01～0.1 以 0.01 为步长调节阈值. 当阈值增加时,性能下降. 因些,我们没有测试阈值大于 0.1 的性能.

3.3.2 性能评价

6 种方法在不同初始故事数的性能($F1$)如图 6 所示. 因为空间限制,我们只展示在 5 个不同的故事数($\{50,150,250,350,450\}$)的结果. 使用显式语义信息组装故事的 Story Forest 和 DBSCAN 有较差的性能,这说明显式信息不能有效地组装故事. DBSCAN 最优的结果准确率为 0.706,召回率为0.459,这证明 DBSCAN 聚类得到的簇粒度较细,即只能将词相似度较高的事件聚到相同的簇. 当初始故事数在[50,450]时,我们的方法和 LDA 的性能优于 BTM 和 GSDMM. 我们的方法使用 DBSCAN 降低 LDA 初始化时的随机性,有比 LDA 更好的性能.

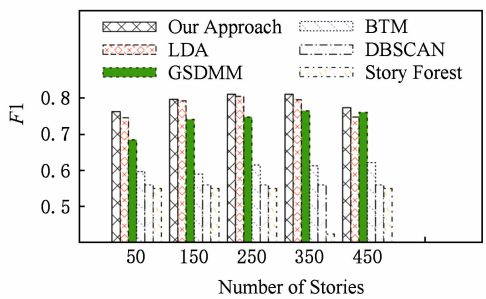


Fig. 6 Performances at different story number N_s

图 6 各算法在不同初始故事数 N_s 的性能

我们对比我们的方法和 LDA 在不同的参数(初始故事数 N_s、最小点数 $minpts$ 和半径 $radius$)的性能. 当 $minpts$ 在[2,4]、$radius$ 在(0.65,0.75)和 N_s 在[100,500]时,我们的方法取得了比 LDA 更大的 $F1$ 值,实验结果见附录 A.

$$convergence_{iter} = \frac{\sum_{e \in E} dis(story\_d_e^{iter}, story\_d_e^{iter-1})}{|E|}, \tag{8}$$

其中,dis 表示 2 个向量的欧氏距离,$story\_d_e^{iter}$ 表示第 $iter$ 次迭代时事件 e 的故事分布.

不同的 Gibbs Sampling 迭代次数 N_{iter} 的性能如图 7 所示. 当 $N_{iter}=1\,000$ 时,4 个方法都达到最优性能;当 $N_{iter}=200$ 时,LDA 没达到最优性能,而我们的方法达到最优性能. 我们依据式(8)计算 LDA 和我们的方法在 Gibbs Sampling 迭代过程的收敛值,并展示在不同的收敛阈值下 LDA 和我们的方法需要的运行时间. 我们的方法用 DBSCAN 降低 LDA 初始化时的随机性,比 LDA 收敛更快,实验结果见附录 B.

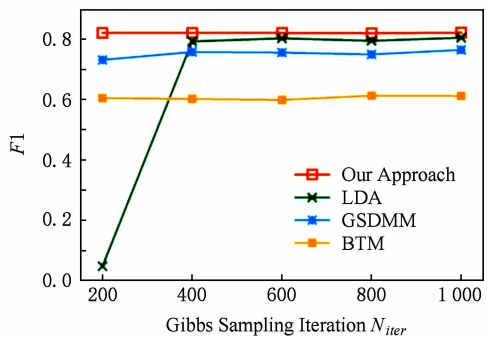


Fig. 7 Performances at different iteration number N_{iter}

图 7 各算法在不同迭代次数 N_{iter} 的性能

在相同的迭代次数(1 000),我们在改变数据集(事件数)和参数(初始故事数)的情况下对比运行时间. 效果最好的 3 个方法,即本文方法、LDA 和 GSDMM 的运行时间如图 8 所示. GSDMM 所需的

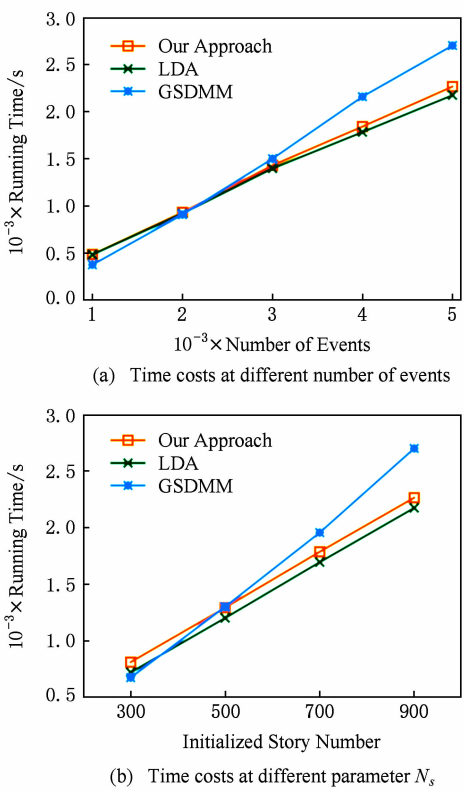


Fig. 8 Running time of different story assembly methods

图 8 不同故事组装方法的运行时间

时间最多.我们的方法用DBSCAN的聚类结果初始化LDA,在相同的迭代次数下本文方法比LDA的运行时间略高.本文方法比LDA收敛快,在实际运行中可通过减小迭代次数缩短运行时间.

综上所述,本文方法有最优的$F1$值;本文方法较LDA收敛快,即实际运行过程中本文方法比LDA需要的运行时间少.

3.4 故事脉络生成实验结果及分析

本节主要评价故事脉络生成的性能.我们基于3个组装故事(Case 1,Case 2,Case 3)对比我们的方法、Timeline[18]和Story Forest[15]的性能.

1) Timeline.该方法基于事件的时间先后关系线性的关联事件.

2) Story Forest.该方法首先判断新事件与已发生事件是否重复,若重复则将2事件合并.该方法然后为非重复的新事件选择父节点.该方法计算新事件与已有事件的连接强度(作者自定义的函数).如果最大的连接强度小于阈值,则新事件的父亲为故事的根结点,否则新事件的父亲为连接强度最大的事件.

Case 1. 中国维和部队遇袭.北京时间2016年6月,联合国在巴里加奥的多层面综合稳定特派团营地被汽车炸弹袭击.中国维和部队中1人牺牲、多人受伤.同年7月,中国维和部队在南苏丹执行任务时被炮弹击中,有2人牺牲、多人受伤.

Case 2. 万科股权之争.中国A股市场上规模最大的并购与反并购攻防战.2015年12月17日,万科股权之争正式进入正面肉搏阶段.

Case 3. 2016里约奥运会.2016年里约热内卢奥运会于2016-08-05—2016-08-21在巴西里约执内卢举行.

我们基于用户体验的方式评价性能.首先,我们将12个志愿者随机平均分成6组.然后,我们将3个组装故事在3个不同方法下的结果①(共9个故事脉络)随机呈现给6组志愿者,并请志愿者在准确性(该脉络是否描述故事的发展过程)和理解性(该脉络是否有助于用户理解故事)2方面对3个方法排序,即针对一个结果志愿者对其进行排序,即最好为1,次好为2,最差为3.最后,我们将排序的算数平均值做为评价故事脉络生成性能的指标.准确性从微观上评价故事脉络,即故事脉络中事件间的连接是否合理.理解性从宏观上评价故事脉络,即故事脉络是否从大体上易于用户理解故事的主要内容.

基于试点用户体验的故事脉络生成的性能评价如表2和表3所示.相同算法在不同案例下的评分不一样,这说明故事脉络的评价存在主观性.我们的方法在3个案例下的准确性和理解性都有最靠前的排名.这说明,较于Timeline和Story Forest,用户倾向我们方法生成的故事脉络.

Table 2 Accuracy by a Pilot User Experience Study

表2 基于试点用户体验的故事脉络生成的准确性

Case	Our Approach	Timeline	Story Forest
Case 1	**1.33**	2.75	1.83
Case 2	**1.50**	2.58	1.75
Case 3	**1.33**	2.75	1.83

Table 3 Comprehension by a Pilot User Experience Study

表3 基于试点用户体验的故事脉络生成的理解性

Case	Our Approach	Timeline	Story Forest
Case 1	**1.58**	2.33	2.08
Case 2	**1.33**	2.58	2.08
Case 3	**1.83**	2.75	2.00

附录C列出2位志愿者对不同方法的评价.从故事脉络和志愿者评论可看出,我们的方法兼顾了准确性和理解性.我们的方法既可区分故事中相对独立的分支以便于用户理解故事,又在分支内能较好体现事件的关联关系.

4 相关工作

故事脉络生成问题在社交网络[18]和传统媒体[19-21]都有相关研究.传统媒体的内容严谨而完整,一篇新闻可完整地描述事件.社会网络的文本短小精悍,具有碎片化和语法不标准等特性.一条微博可能只包含事件的碎片化信息.基于文章可描述事件的方法[19,21]直接用于社交网络可能得不到理想的效果.

解决故事脉络生成问题的方法大致分为2类:分步法和整合法.分步法将问题形式化为多个组件,即事件检测、故事组装和故事脉络生成;整合方法则尝试构造一个统一模型来解决该问题.

1) 分步法.这里我们分别对事件检测、故事组装和故事脉络的相关工作进行回顾.①事件检测.Lee等人[22]将社交网络流建模为动态微博网络,并

① https://github.com/liyingrenjie/storyline2

将网络中紧密连接的微博集合做为事件. Story Forest[15]对新闻文本流聚类,并将簇做为事件. ②故事组装. Story Forest[15]依据事件与已有故事的语义距离将事件分配到特定故事. ③故事脉络生成. Lee 等人[22]用事件间的 Jaccard 系数追踪事件间的演化关系. Story Forest[15]在故事内依据自定义的函数生成故事脉络. Lee 等人[22]用关键词集合表示事件,不利于用户理解事件. Lee 等人[22]依据事件的微博相似度是否大于阈值判定事件的演化关系;这种方法存在 2 个问题:①只能连接相似度较高的事件;②会引入一些不必要的连接.

2) 整合法. CHARCOAL[23]用提出的概率图模型对新闻文章间的联系(link)建模,对故事的进展(progress),并通过新闻文章间的联系生成故事脉络. 单个微博可能不包含事件的所有关键信息(例如地点和参与者),因此 CHARCOAL 不能直接用于社交网络. DSEM[24]和 DSDM[25]用非参数化的生成模型同时提取事件的结构化表示和事件在连续时间片的演化模式. MEP[26]用基于非负矩阵分解的主题模型同时检测事件和连续时间片的事件的演化. 然而,这些模型只能追踪连续变化的模式,难以对时间跨度大、不连续的故事内事件间的演化进行追踪.

5 总结与展望

在社交网络通过故事脉络对事件及事件间的演化建模具有重要意义且极具挑战. 我们将故事脉络生成问题形式化为事件检测、故事组装和故事脉络生成 3 个连续的组件,并提出了解决框架. 我们提出用包含摘要的有向无环图可视化故事的发展过程. 新浪微博数据集上进行的实验表明我们的方法能有效展示故事的发展过程. 社交网络存在大量的用户关系、用户行为和用户画像等信息,且用户是社交网络的主要参与者,这些信息对于故事脉络生成有重要参考价值. 社交网络每天产生大量的数据,online 的故事脉络生成方法便于大规模部署. 如何将方法修改为 online 的模式并融合用户信息是下一步的研究工作.

参 考 文 献

[1] Shahaf D, Yang J, Suen C, et al. Information cartography: Creating zoomable, large-scale maps of information [C] // Proc of the 19th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2013: 1097-1105

[2] Wang Dingding, Li Tao, Ogihara M. Generating pictorial storylines via minimum-weight connected dominating set approximation in multi-view graphs [C] //Proc of the 26th AAAI Conf on Artificial Intelligence. Menlo Park, CA: AAAI, 2012: 683-689

[3] Lin Chen, Lin Chun, Li Jiangxuan, et al. Generating event storylines from microblogs [C] //Proc of the 21st ACM Int Conf on Information and Knowledge Management. New York: ACM, 2012: 175-184

[4] Lee P, Lakshmanan L V S, Milios E. CAST: A context-aware story-teller for streaming social content [C] //Proc of the 23rd ACM Int Conf on Information and Knowledge Management. New York: ACM, 2014: 789-798

[5] Khurdiya A, Dey L, Raj N, et al. Multi-perspective linking of news articles within a repository [C] //Proc of the 22nd Int Joint Conf on Artificial Intelligence. San Francisco: Morgan Kaufmann, 2011: 2281-2286

[6] Xie Lexing, Sundaram H, Campbell M. Event mining in multimedia streams [J]. Proceedings of the IEEE, 2008, 96 (4): 623-647

[7] Yu Weiren, Aggarwal C C, Ma Shuai, et al. On anomalous hotspot discovery in graph streams [C] //Proc of the 13th Int Conf on Data Mining. Piscataway, NJ: IEEE, 2013: 1271-1276

[8] Palla G, Derényi I, Farkas I, et al. Uncovering the overlapping community structure of complex networks in nature and society [J]. Nature, 2005, 435(7043): 814-818

[9] Lu Zhongyu, Yu Weiren, Zhang Richong, et al. Discovering event evolution chain in microblog [C] //Proc of the 17th IEEE Int Conf on High Performance Computing and Communications. Piscataway, NJ: IEEE, 2015: 635-640

[10] Blei D M, Ng A Y, Jordan M I. Latent Dirichlet allocation [J]. Journal of Machine Learning Research, 2003, 3: 993-1022

[11] Ester M, Kriegel H P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise [C] //Proc of the 2nd Int Conf on Knowledge Discovery and Data Mining. Menlo Park, CA: AAAI, 1996: 226-231

[12] Mihalcea R, Tarau P. Textrank: Bringing order into text [C] //Proc of the 2004 Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2004: 404-411

[13] Yan Xiaohui, Guo Jiafeng, Lan Yanyan, et al. A biterm topic model for short texts [C] //Proc of the 22nd Int Conf on World Wide Web. New York: ACM, 2013: 1445-1456

[14] Yin Jianhua, Wang Jianyong. A dirichlet multinomial mixture model-based approach for short text clustering [C] // Proc of the 20th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2014: 233-242

[15] Liu Bang, Niu Di, Lai Kunfeng, et al. Growing story forest online from massive breaking news [C] //Proc of the 2017 ACM on Conf on Information and Knowledge Management. New York: ACM, 2017: 777-785

[16] Kalyanam J, Mantrach A, Saez-Trumper D, et al. Leveraging social context for modeling topic evolution [C] //Proc of the 21st ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2015: 517-526

[17] Chen Yong, Zhang Hui, Wu Junjie, et al. Modeling emerging, evolving and fading topics using dynamic soft orthogonal nmf with sparse representation [C] //Proc of the 2015 IEEE Int Conf on Data Mining. Piscataway, NJ: IEEE, 2015: 61-70

[18] Wang Zhenhua, Shou Lidan, Chen Ke, et al. On summarization and timeline generation for evolutionary tweet streams [J]. IEEE Trans on Knowledge and Data Engineering, 2015, 27(5): 1301-1315

[19] Hua Ting, Zhang Xuchao, Wang Wei, et al. Automatical storyline generation with help from Twitter [C] //Proc of the 25th ACM Int on Conf on Information and Knowledge Management. New York: ACM, 2016: 2383-2388

[20] Huang Lifu, Huang Lianen. Optimized event storyline generation based on mixture-event-aspect model [C] //Proc of the 2013 Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2013: 726-735

[21] Li Fenghuan, Zheng Dequan, Zhao Tiejun. Dynamic incremental analysis of sub-topic evolution [J]. Journal of Computer Research and Development, 2015, 52(11): 2441-2450 (in Chinese)
(李风环, 郑德权, 赵铁军. 动态增量式子主题事件演化分析 [J]. 计算机研究与发展, 2015, 52(11): 2441-2450)

[22] Lee P, Lakshmanan L V S, Milios E E. Incremental cluster evolution tracking from highly dynamic network data [C] //Proc of the 30th IEEE Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2014: 3-14

[23] Tang Siliang, Wu Fei, Li Si, et al. Sketch the storyline with CHARCOAL: A non-parametric approach [C] //Proc of the 24th Int Joint Conf on Artificial Intelligence. San Francisco: Morgan Kaufmann, 2015: 3841-3848

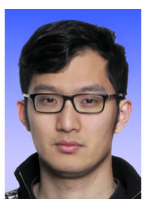
[24] Zhou Deyu, Xu Haiyang, Dai Xinyu, et al. Unsupervised storyline extraction from news articles [C] //Proc of the 25th Int Joint Conf on Artificial Intelligence. San Francisco: Morgan Kaufmann, 2016: 3014-3021

[25] Zhou Deyu, Xu Haiyang, He Yulan. An unsupervised Bayesian modelling approach for storyline detection on news articles [C] //Proc of the 2015 Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2015: 1943-1948

[26] Kalyanam J, Velupillai S, Conway M, et al. From event detection to storytelling on microblogs [C] //Proc of the 2016 IEEE/ACM Int Conf on Advances in Social Networks Analysis and Mining. Piscataway, NJ: IEEE, 2016: 437-442

Li Yingying, born in 1992. Master. Her main research interests include social data analysis and data mining.

Ma Shuai, born in 1975. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include database theory and systems, social data analysis, and data intensive computing.



Jiang Haoyi, born in 1990. Master candidate. His main research interests include automatic summarization and data mining.

Liu Zhe, born in 1984. PhD, lecturer. His main research interests include machine learning, data mining and their applications in genomics and biomedical data.

Hu Chunming, born in 1977. PhD, associate professor. Member of CCF. His main research interests include distributed systems, system virtualization, large scale data management and processing systems.

Li Xiong, born in 1982. PhD, senior engineer. His main research interests include hybrid generative discriminative learning, probabilistic graphical model and social natural language understanding.

附录 A

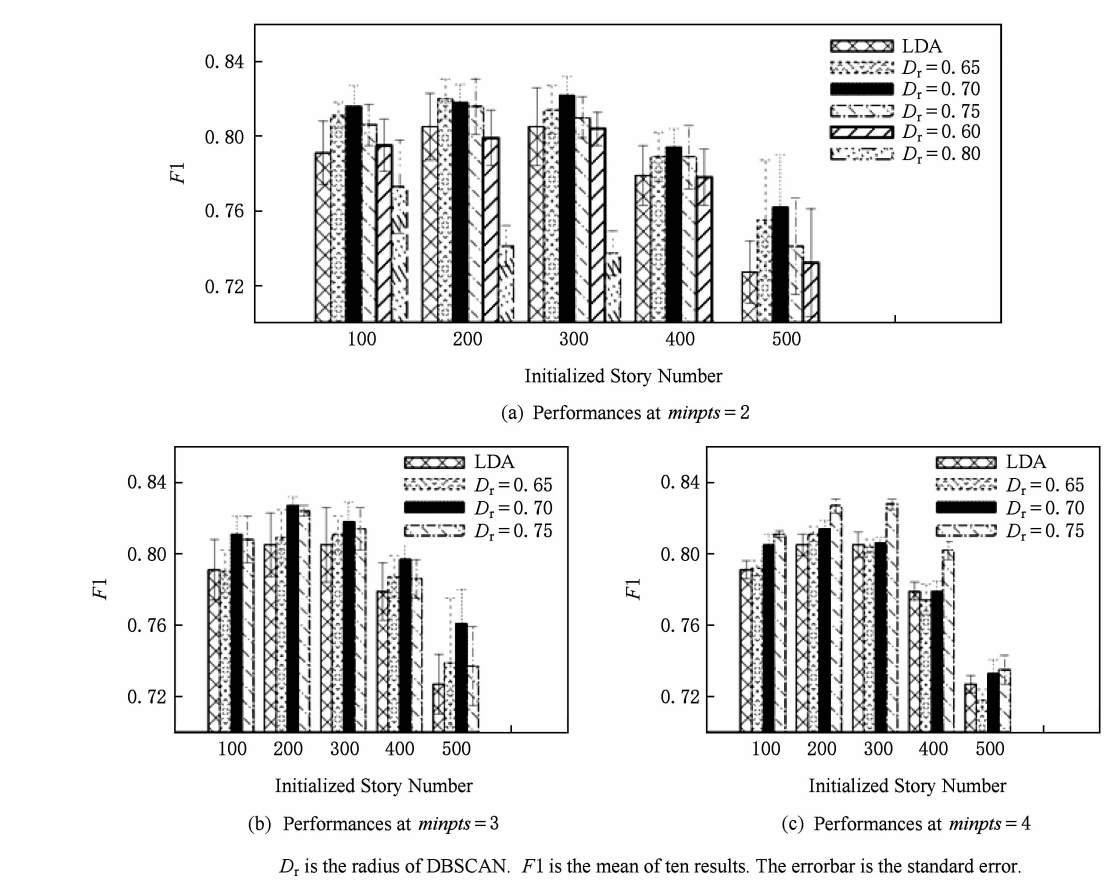


Fig. A1 The performance evaluation $F1$ of our approach and LDA

图 A1 我们的方法和 LDA 的性能评价 $F1$

附录 B

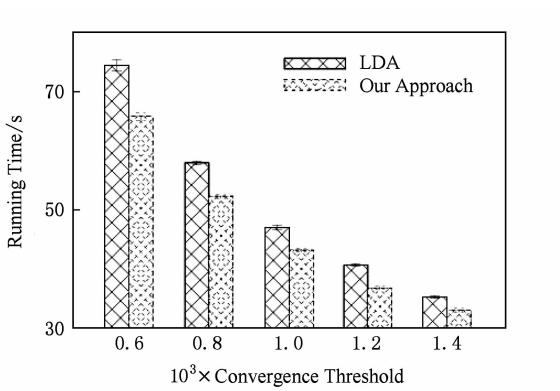


Fig. B1 Time costs at different convergence threshold

图 B1 不同收敛值的运行时间

附录 C

1) 志愿者 1 的评论

Our approach:如果有多个独立的分支,且分支在时间上有交叉(例巴西奥运),则我们方法的可读性会比较好.基本上该有的连接我们的方法都有.我们的方法有时会出现一些不必要的连接(在一些 case 下用户可区分哪些连接是不必要的).

Timeline:如果事件在时间上可以自然地聚成几堆(例中国维和部队遇袭),则 Timeline 的可读性好.但是需要手动区分前后 2 个事件(分支).在正确表达事件关系上效果最差.

Story Forest:该方法的连接太少,以至于跟 Timeline 的可读性没什么区别.

2) 志愿者 2 的评论

Our approach:虽然在多个分支上内容有交叉,但主要内容还是看的出来的.

Timeline:事件都交叉在一起,感觉乱.

Story Forest:该方法连接的不准确,增加了读的难度.