软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2012,23(4):735−747 [doi: 10.3724/SP.J.1001.2012.04031]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

短文本信息流的无监督会话抽取技术*

黄九鸣[1+], 吴泉源[1], 刘春阳[2], 张 旭[2], 贾 焰[1], 周 斌[1]

[1](国防科学技术大学 计算机学院,湖南 长沙 410073)

[2](国家计算机网络应急技术处理协调中心,北京 100029)

Unsupervised Conversation Extraction in Short Text Message Streams

HUANG Jiu-Ming[1+], WU Quan-Yuan[1], LIU Chun-Yang[2], ZHANG Xu[2], JIA Yan[1], ZHOU Bin[1]

[1](College of Computer, National University of Defense Technology, Changsha 410073, China)

[2](National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China)

+ Corresponding author: E-mail: naicky@gmail.com, http://www.nudt.edu.cn

Huang JM, Wu QY, Liu CY, Zhang X, Jia Y, Zhou B. Unsupervised conversation extraction in short text message streams. *Journal of Software*, 2012,23(4):735−747. http://www.jos.org.cn/1000-9825/4031.htm

Abstract: Short text message streams are produced by Short Message Service, Instant Messager and BBS, which are widely used. Each stream usually contains. Extracting the conversations in the streams is helpful to various applications including business intelligence, investigation of crime and public opinion analysis. Existing research mainly based on text similarity encounter challenges such as the anomaly, dynamics, and the sparse eigenvector of short text message. This paper proposes an innovative conversation extraction method to cover the challenges. Firstly, the study detects the conversation boundary of short text message streams using temporal feature; secondly, contextually correlative degree is introduced to replace similar degree, and an instance-based machine learning method is proposed to compute the correlative degree. Finally, the study designs Single-Pass based conversation extraction algorithm SPFC (single-pass based on frequency and correlation), which combines the temporal and contextually correlative characteristics. Experimental results on a large real Chinese dataset show that this method SPFC improves the performance by 30% when compared with the best existing variation algorithm in terms of $F1$ measure.

Key words: conversation extraction; short text message; short text message stream; unsupervised; temporal feature; contextually correlative degree

摘 要: 文本会话抽取将网络聊天记录等短文本信息流中的信息根据其所属的会话分检到多个会话队列,有利于短文本信息的管理及进一步的挖掘.现有的会话抽取技术主要对基于文本相似度的聚类方法进行改进,面临着短文本信息流的特征稀疏性、奇异性和动态性等挑战.针对这些挑战,研究无监督的会话抽取技术,提出了一种基于信息流时序特征和上下文相关度的抽取方法.首先研究了信息流的会话生命周期规律,提出基于信息产生频率的会话边界检测方法;其次提出信息间的上下文相关度概念,采用基于实例的机器学习方法计算该相关度;最后综合信息产生频率和上下文相关度,设计了基于 Single-Pass 聚类模型的会话在线抽取算法 SPFC(single-pass based on frequency

* 基金项目: 国家自然科学基金(60933005, 60873204); 国家高技术研究发展计划(863)(2001AA012505); 国家 242 信息安全计划课题(2009A90)

收稿时间: 2010-11-04; 定稿时间: 2011-03-21

CNKI 网络优先出版: 2011-07-04 15:45, http://www.cnki.net/kcms/detail/11.2560.TP.20110704.1545.002.html

and correlation).真实数据集上的实验结果表明,SPFC 算法与已有的基于文本相似度的会话抽取算法相比,*F*1 评测指标提高了 30%.

关键词: 会话抽取;短文本;短文本信息流;无监督;时序特征;上下文相关度

中图法分类号: TP391 **文献标识码:** A

短文本信息流存在于当今广泛使用的手机短信、互联网即时通信、论坛和微博等系统中.一个短文本信息流通常包含多个会话,涉及多个话题.会话抽取任务旨在根据短文本信息讨论的话题以及信息间的对话关系,将信息分检到多个队列,每个队列是一段主题明确的会话.以会话组织的短文本信息数据,比原始的按时间顺序组织的短文本信息流更便于内容管理和进一步的挖掘.然而,人工在海量的文本信息流中抽取会话是一项费时费力、甚至不可能完成的艰巨任务.很多国家已在网络聊天记录的自动化分析上做出尝试,并有一定进展[1,2].

短文本信息流的特征稀疏性、奇异性、动态性[3]和交错性等特点,给会话抽取任务提出重大挑战.短文本信息的长度短、信息量少,因此,以词为维度的向量空间模型呈现出高维稀疏的特点.奇异性是指网络聊天语言中广泛存在的谐音词和简写词,如“稀饭”代表“喜欢”;动态性反映出短文本信息流上的流行词语随着时间不断变化,并不断有新词出现;交错性是指短文本信息流中的会话交错出现,相邻的信息可能讨论不同的话题,隶属于不同的会话.

关于文本会话抽取的研究始于 2000 年 Smith 等人[4]对会话树和聊天线程的研究,近年来更成为研究热点.在已有的研究中,有利用知识库来扩展短文本的特征向量以解决特征稀疏性问题;有利用信息的时序对词的重要性加权以适应信息流的动态性;有利用信息的语言特征改善奇异性带来的影响;还有一些采用了基于规则的机器学习方法.然而,现有的方法大部分只对基于文本相似度的聚类方法进行特征扩展,忽视了文本信息间的交互性,即上下文相关性.只简单地以信息的时间顺序对特征向量值进行加权,忽视了会话深层的时序特征.

针对这些挑战和已有研究的不足,本文创新性地提出了基于时序特征和上下文相关度的短文本文本信息流会话抽取方法.该方法是一种无监督的机器学习方法.首先,利用信息流中会话的生命周期规律,基于信息产生频率初步检测出会话边界;其次,定义了信息间的会话上下文相关度的概念,并采用基于实例的无监督机器学习方法计算这一相关度;最后,提出了短文本信息流的在线话题分检算法 SPFC(single-pass based on frequency and correlation).该算法动态更新相关度的训练语料,解决了信息流的语言动态性问题.本文的方法用信息间的上下文相关度代替相似度,更合理而有效,能够解决特征稀疏性、奇异性带来的影响.在一个时间跨度长达一个月的 Linux 技术讨论 QQ 群聊天记录上,SPFC 算法与 SP_{NN},SP_{WC},SP_{WNN}[5]这 3 种基于文本相似度的改进算法相比,*F*1 评测指标提高了 30%.

本文第 1 节对相关研究进行介绍.第 2 节给出问题定义.第 3 节给出基于时序特征和上下文相关度的短文本信息流会话抽取方法.第 4 节通过实验表明算法的有效性,测试算法的运行效率.最后给出总结和展望.

1 相关研究

计算语言学很早就开始研究文本会话.Grice[6]提出自然语言有其独特的逻辑关系,会话的最高原则是合作,称为合作原则.在这个原则下,人们遵守数量、质量、关联、方式这 4 项准则;文献[3]研究了中文网络聊天语言的奇异性和动态性,指出网络聊天用语经常是不规范的,是一种包含很多简写、谐音字、新词的网络非正规语言.

与会话抽取任务相似的传统文本挖掘技术是话题检测与跟踪(topic detection and tracking,简称 TDT)[7],它的主要任务是标识出文本集合中的文档所属的话题,主要的方法分为在线话题检测和回顾话题检测两类.与本文工作相似的是在线话题检测,这方面的经典算法是 Single-Pass[8].已有的 TDT 算法都假定每篇文档有足够的信息表明它所属的话题,在新闻报道、学术文章等长文本上已比较有效.然而,传统的 TDT 技术没有考虑短文本信息的特征稀疏性、时序性、交互性、奇异性和动态性,导致计算出的短文本信息间的相似度都很低,难以区分短文本信息的差异程度,因此,信息所属会话也难以判断.

文本会话边界检测是文本会话抽取早期研究的目标,主要有 3 条技术途径:第 1 条是采用统计和监督学习的方法[9];第 2 条是基于词的一致性,通过已有的外部知识源构建的词汇链来检测信息的一致性[10];第 3 条是综合了统计方法和相似度测量方法[11].

近期的研究主要尝试利用文本信息流中的用户信息和时间信息,更深入地从语义和语法层面改进会话抽取的效果.有些算法是在 Single-Pass 聚类算法的基础上进行了改进.Shen[6]的方法基于向量空间模型,用信息产生的时间顺序对特征向量进行加权,分别用 *K*NN 和中心向量两种方法判断信息与会话的相似度.此外,还引入了语言特征来计算信息间的上下文相关性.特征之一是信息中使用的句型,另一个是个人的拼写习惯,通过统计训练语料中各种句型组合是否属于同一会话的概率,给信息间的相似度加上一个系数,以改善文本会话抽取的效果.Wang[12]在文献[5]的工作基础上,利用知网扩展短文本的特征项设计了缓存相似文本信息的内存结构,用以对信息进行会话分组,并提出了双时间窗口的聚类算法,使得会话抽取可应用于在线高速文本信息流.针对短文本信息特征稀疏的缺点,采用特征扩展可在一定程度上加以克服.然而,特征扩展的效果依赖于知识库,而网络聊天数据存在奇异性和动态性,要维护一个全面的知识库,人力代价巨大.文献[13]在新闻组风格的会话中研究了隐含线程结构的发现,其方法与 Shen 的方法很相似,假设信息间的时间靠得越近,文本相似度越高,则越有可能存在父子关系.

除了基于 Single-Pass 方法以外,基于规则的机器学习算法也被用于会话抽取.Wu[14]最先从语法层面研究聊天室的会话挖掘,采用基于错误驱动的布尔逻辑规则学习算法.但这种方法依赖于专家制定的规则,当聊天数据的内容所属领域不一样时,需要制定新的规则集合,维护代价较高.

近来十分流行的 LDA 模型也被应用到会话抽取任务中,比如文献[15],提出了一个名为 SMSS 的稀疏编码模型,同时对会话的语义和结构进行建模.该模型将每条信息映射到一个话题空间,并通过线性组合同一会话先前的信息来度量每条信息与会话的相似度.

2 问题定义

本节首先给出短文本信息和会话的定义,然后明确定义会话抽取的任务.

定义 1(文本信息,text message). 一条文本信息是指使用数字化终端参与会话的用户一次发言的文本片段.用户使用数字化终端编写,通过网络发送文本信息与其他用户进行对话.

信息又可以分为起始信息、回复信息和终止信息.一条起始信息引发一个新信息序列,这个信息序列关注某个特定的主题;回复信息是对前面某条信息的某种回应,比如肯定、强调或补充等;一条终止信息结束对某个主题的讨论.我们将信息发送的时刻称为信息的产生时刻,信息的字数称为信息长度.

定义 2(短文本信息,short text message). 短文本信息是指信息长度较短(字数一般不超过 30 个字)、信息含量较小的那类文本信息,如论坛回帖、即时聊天消息、手机短信等.

定义 3(文本会话,text conversation). 一个文本会话是一个围绕某个特定主题的信息序列,开始于一条起始信息,结束于一条终止信息,中间有若干条回复信息.

例 1:图 1 右侧中,方框底色为白色的 4 条信息构成了一个会话,底色为灰色的 3 条信息构成另一个会话.每个会话只围绕一个特定的具体主题,并且会话中的每条信息是对同一会话中产生时刻更早的某条信息的回复.

文本会话的粒度小于 TDT 任务中话题的粒度,有利于下一步基于会话展开更深入的研究,比如基于会话研究用户间的社交关系、研究文本信息流中的社区网络.

定义 4(会话抽取,conversation extraction). 会话抽取将文本信息流中的每条信息映射到其所属的会话,并在文本信息流中发现新会话.如图 1 所示,会话抽取将图中左侧按时间顺序排列的信息流,按文本会话将信息重新组织为右侧所示的两个会话队列.

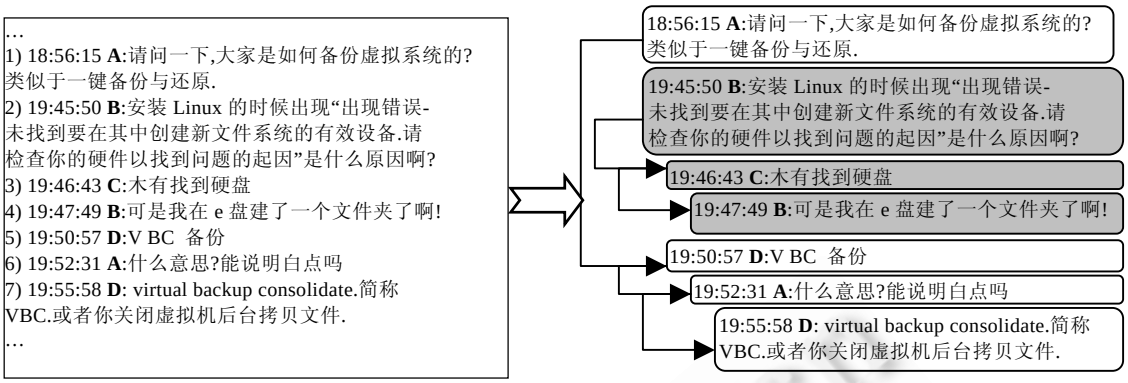

Fig.1 Messages organized in threads

图 1 文本会话抽取任务示意

3 基于时序特征和上下文相关度的文本信息流会话抽取

会话抽取在本质上是一个聚类问题.本文提出的会话抽取方法首先利用会话的生命周期性将文本信息流切分为粒度较小的会话片段;再利用信息间的会话上下文相关性,对第 1 步切分的细粒度会话片段进行聚合,得到最终的文本会话.

第 3.1 节利用文本信息流的时序特征——会话生命周期性,提出基于信息产生频率的会话边界检测方法;第 3.2 节定义了会话上下文相关度,提出了一种基于机器学习方法的相关度计算方法;第 3.3 节给出支持训练语料动态更新,综合信息产生频率和上下文相关度的在线会话抽取算法.

3.1 基于信息产生频率的会话边界检测

首先以一个例子说明该方法的思路.

例 2:根据文本会话的定义,我们对一个 Linux 技术交流 QQ 群进行了人工标注,得到一个熟语料.图 2 所示是从这个语料中随机截取的聊天片段,横轴为时间,纵轴为信息条数,实线为单位时间产生的信息条数(即信息产生的速率),虚线表示文本会话的边界.可见,大部分情况下,会话边界处于信息产生速率曲线的波谷.通过观察大量的熟语料我们发现,信息流中会话的边界点与信息产生的速率有关.

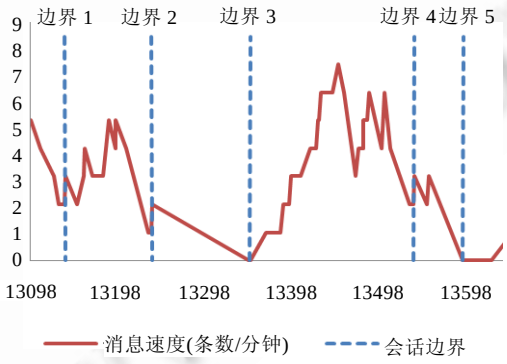

Fig.2 Evolution of message produce speed

图 2 信息产生速率变化图

性质 1. 人们的会话过程符合事物发展的一般生命周期规律,经历起源、发展、高潮、衰弱、消亡几个阶段.在文本会话中,这一规律体现在特定信息属性的变化上.例如,信息产生的速率逐渐变大(发展)并达到最大值,

保持一段平稳期(高潮)后,开始变小(衰弱)直至为 0(消亡).

图 2 所示例子中信息产生速率的变化趋势与大部分会话边界的关系符合性质 1.会话边界 4 是一个例外,观察数据发现,这是因为人们讨论的主题发生了漂移.应用性质 1 的规律,若暂不考虑文本信息流的主题交错性,则会话抽取任务简化为求解会话边界的时刻.因此有如下定义和方法.

定义 5(信息产生时刻). 函数 $\tau(M)$的值为信息 M 的产生时刻距离 1970 年 1 月 1 日 0 时 0 分的毫秒数.

定义 6(信息流的信息条数). 对于某个短文本信息流 S,用函数 $\phi_S(t)$表示信息流 S 从诞生时刻至时间点 t 所产生的信息条数.

$\phi_S(t)$的一阶导数 $\phi'_S(t)$ 表示时间点 t 的信息产生频率.我们所求的会话边界时刻是 $\phi'_S(t)$ 的值由大变小,而后由小变大(或持续为 0)的转折点.令 $\phi''_S(t)$ 为 $\phi_S(t)$的二阶导数,公式(1)所示的方程组的解就是会话边界点:

$$\begin{cases}\lim\limits_{\varepsilon\to 0}\phi''_S(t-\varepsilon)<0\\ \phi''_S(t)=0\\ \lim\limits_{\varepsilon\to 0}\phi''_S(t+\varepsilon)\geqslant 0\end{cases} \tag{1}$$

信息的产生时间是离散的,因此 $\phi'_S(t)$ 和 $\phi''_S(t)$ 须采样拟合.为了使拟合更加准确,我们设定每条信息产生的时刻为采样点.另外,如图 2 所示,信息产生频率的微小波动不能认为是会话的边界.对信息产生频率这个时序数据进行平滑,可减小微小波动带来的影响.采用 n 阶移动平均法来进行平滑,对任意信息 M_i,可用如下公式求得 M_i 产生时刻的产生频率 $\phi'_S(\tau(M_i))$:

$$\phi'_S(\tau(M_i))=\sum_{k=i-v}^{i+u}\frac{\phi_S(\tau(M_k))-\phi_S(\tau(M_k)-\Delta t)}{\Delta t\times(u+v)} \tag{2}$$

其中,Δt 是一个可配置的参数;v 指的是时间期间$[\tau(M_i)-\omega,\tau(M_i)]$内产生的信息条数,$u$ 指的是时间期间$[\tau(M_i),\tau(M_i)+\omega]$内产生的信息条数,$\omega$是一个可配置的参数.由于$(\tau(M_k)-\Delta t)$不一定在采样点上,可用距$(\tau(M_k)-\Delta t)$最近的采样点的信息条数来替代$\phi_S(\tau(M_k)-\Delta t)$的值.

在求得$\phi_S(t)$的基础上,$\phi''_S(\tau(M_i))$ 的值用公式(3)逼近:

$$\phi''_S(\tau(M_i))=\frac{\phi'_S(\tau(M_i))-\phi'_S(\tau(M_{i-n}))}{\tau(M_i)-\tau(M_{i-n})} \tag{3}$$

其中,n 是一个可配置的常量.

同样,$\phi''_S(t)$ 为 0 的时间点也不一定在采样点上,故公式(1)的方程组简化为公式(4)所示的方程组:

$$\begin{cases}\phi''_S(\tau(M_{i-1}))<0\\ \phi''_S(\tau(M_i))\geqslant 0\end{cases} \tag{4}$$

该方程组所求得的 M_i 就是区分会话边界的信息.

3.2 基于实例的上下文相关度计算方法

从信息内容的层面来判断两条信息是否存在会话上下文关系,已有的方法主要基于信息间的内容相似度.然而真实对话中,构成回复关系的两条信息内容可能完全不同.例如,“感谢你们的帮助”与“不用客气”这两条信息.为此,我们提出信息会话上下文相关度(简称相关度)的概念,度量信息间构成对话关系的可能性大小.用会话上下文相关度来对信息进行聚类以得到会话,比简单地根据相似度进行聚类更合理和有效.

基于实例的无监督机器学习方法计算信息间的上下文相关度,主要思想是利用历史信息中相邻的信息往往更可能属于同一个会话这一性质,从历史信息中分别找出与待判定相关度的两条信息相似的信息集合,一方面作为待判定信息的特征扩展数据,另一方面进一步从相似信息集合找出前导信息集合,计算待判定信息与前导信息集合的相似度.最后,综合待判定信息间的相似度和与前导信息集合间的相似度,计算出最终的相关度.

定义 7(会话上下文相关度). 信息 M_i 和 M_k 间的相关度ρ表示这两条信息构成会话上下文关系的可能性,用二元函数$\rho(M_i,M_k)$表示,值越大表示可能性越大.

通过大量观察我们发现,网络聊天记录中时间顺序上紧临的两条信息,在大部分情况下,时间晚的信息是对

时间早的信息的回复.并且,时间顺序上不紧邻的两条信息,如果它们产生的时间间隔较小,也有可能构成对话.

性质 2. 同一个短文本信息流中的两条信息 M_i 和 M_k 相距越近,这两条信息的会话上下文相关度越大.即

$$\frac{1}{|i-k|} \propto \rho(M_i, M_k) \tag{5}$$

自然语言难以直接进行相似度的度量,通常做法是把文档分解成语法成分、词和长度等特征项,用特征项的权重构成的向量来表示文档.为便于表述,我们采用类似于文档向量的方法,不同之处在于我们忽略了权重为 0 的特征项.一条信息可以看作一个集合,其元素是权重不为 0 的特征项.

定义 8(邻接共现频率). 特征项 w 和 w'相继在相邻的两条信息出现,称 w 和 w'邻接共现.给定信息流片段 S,二元函数$\chi(w,w')$表示 w 和 w'在 S 中邻接共现的频率,其定义如下:

$$\chi(w,w') = \frac{|\{i \mid w \in M_i \wedge w' \in M_{i+1} \wedge M_i \in S\}|}{|S|} \tag{6}$$

定理 1. 任意特征项 w,w'和信息 M,M',若 $w \in M$ 且 $w' \in M'$,则$\chi(w,w') \propto \rho(M,M')$成立.

证明:由性质 2 可知,文本信息流中时序上紧邻的两条信息较有可能构成会话上下文关系.换言之,文本信息流中,大部分紧邻的信息之间构成会话上下文关系.而信息是特征项的集合,如果特征项 w 和 w'在信息流中经常邻接共现,说明 w 和 w'较有可能在同一个会话的上下文里同时出现.因此,若 $w \in M$ 且 $w' \in M'$,那么信息 M,M'构成会话关系的可能性也就越大,即$\chi(w,w')$和$\rho(M,M')$成正比. □

大部分情况下,单个特征项只是信息某个维度的特征,不是任意特征项都能决定信息间的上下文关系.比如,“感谢你们的帮助”与“不用客气”这两条信息,起决定作用的是“感谢”、“帮助”和“客气”这 3 个特征项.因此,特征项集合 W,W'在信息流中邻接共现的次数越多,W,W'的基数越大,就越能决定信息间是否有上下文关系,如推论 1 所示.

推论 1. 给定信息流片段 S,对任意特征项集合 W,W',若 $W \subseteq M$ 且 $W' \subseteq M'$,则有如下公式成立:

$$|\{i \mid W \subseteq M_i \wedge W' \subseteq M_{i+1} \wedge M_i \in S\}| \times |W| \times |W'| \propto \rho(M,M') \tag{7}$$

根据定理 1 和推论 1,针对某个信息流,可指定某个历史片段为训练语料,计算出各个特征项的 IDF 值、所有特征项的各种组合间的邻接共现率,从而计算信息间的相关度.但是,这种方法的时间开销巨大.并且,由于网络聊天语言的动态性,我们必须经常地更新训练语料.因此,采用基于实例的机器学习方法,实时地从历史信息流中学习出信息间的相关度更为合适.

首先,为作为训练语料的信息流片段 S 构建特征项到信息的倒排索引.当信息 M 到达时,根据 M 的特征项,在倒排索引中搜索出最相似的前μ条信息,组成信息集合α.然后,构建如公式(8)所示的前导信息集合β:

$$\beta = \bigcup_{M_i \in \alpha} \{M_j \mid i-k < j < i, M_j \in S\} \tag{8}$$

其中,β表示集合α中的每条信息在 S 中的 k 条前导信息组成的集合.以特征项为维度,以特征项的 TF-IDF 值为维度的值构成信息的特征向量.集合的中心向量[5]为集合中所有信息向量的和的正规化向量.由定理 1 和推论 1 可知,信息 M'中权重不为 0 的特征项在β的中心向量 $\vec{\beta}$ 中的权重越高,M'与 M 的相关度就越高.另外,讨论同一主题的信息经常出现相同的关键词,两条信息的相似度越高,相关度也越高.因此,定义 M'与 M 的相关度为 M'的特征向量 $\overrightarrow{M'}$ 与 $\vec{\beta}$ 的角余弦相似度,以及 M'与 M 的相似度的综合值,如公式(9)所示:

$$\rho(M,M') = \cos(\overrightarrow{M'}, \vec{\beta}) \times (1 - \cos(\overrightarrow{M'}, \vec{\alpha})) + \cos(\overrightarrow{M'}, \vec{\alpha}) \tag{9}$$

其中,$\vec{\alpha}$ 表示将α作为信息 M 的特征扩展后得到的向量,即$\{M\} \cup \alpha$的中心向量.

3.3 在线会话抽取算法SPFC

本节综合第 3.1 节和第 3.2 节的两种方法,基于 Single-Pass 聚类模型设计了在线会话抽取算法 SPFC.单独使用第 3.1 节或第 3.2 节的方法都不能达到较好的效果.基于信息产生频率的会话边界检测利用文本信息流的时序特性有效地将一个文本信息流切分成多个会话片段.但是,它不能处理交错性问题.如果切分粒度较粗,那么交错在一起的多个会话无法被正确地区分开来;如果粒度较细,则会将原本属于同一会话的信息切成多个片

段,导致召回率降低.另一方面,虽然基于实例的上下文相关度计算方法利用历史信息,有效地将属于不同会话却交错出现的信息正确地分检到其所属队列,但却没有考虑信息出现的时序特性,受短文本特征稀疏性影响很大.相关度方法虽然比相似度方法具有更高的准确率,但仍难以区分一些应用过于广泛的日常用语(如“是”、“好的”)与上下文的关系.此外,相关度计算过程需要搜索历史记录,计算开销也远大于其他计算方法.

因此,为了较好地处理交错性问题,有效利用会话的时间特性,并面向海量数据实现高效处理,我们提出SPFC 算法(如算法 1 所示):对文本信息流,先采用基于信息产生频率的边界检测方法将其切分为多个较细粒度的会话片段;再用基于实例的上下文相关度计算方法计算各个细粒度的会话片段间的相关度,采用 Single-Pass 聚类模型,聚合细粒度的会话片段得到最终的文本会话.

算法 1. SPFC.

输入:文本信息流 S,训练语料 G;

输出:文本会话.

```
1.  while message M_i arriving in S do
2.  begin
3.    T_L=Thread in TW which contains M_{i-1};
4.    if M_i satisfy the equations (4) then
5.    begin
6.      maxP=0;
7.      for every thread T_J in TW do
8.        if ρ(M_i,T_J)>maxP then
9.        begin
10.         maxP=ρ(M_i,T_J);
11.         maxT=T_J;
12.       end;
13.     if maxP>ζ then
14.       add M_i to T_J;
15.     else
16.       add M_i to T_L;
17.   end;
18. end;
```

由于会话的数量很多,SPFC 算法的实现采用了双时间窗口机制:只检测最新的 t_W 个会话,每个会话只检测最新的 d_W 条信息.SPFC 算法定义了一个大小可配置的会话队列 TW.TW 队列采用先进先出的原则,其大小称为会话窗口 t_W.每到达一条信息,算法先使用公式(4)检测该信息是否为会话边界:若不是,则加入到最近的一个会话中;若是,则计算其与 TW 中每个会话的相关度,若大于阈值ζ,则将该起始信息加入到相关度最大的会话,否则创建新的会话.会话相关度以输入的训练语料 G 为学习实例,采用下面第 4.2 节所述方法计算.训练语料 G 为当前文本信息流的某个历史片断的倒排索引.实际应用中,训练语料可动态更新以解决动态性问题.

假设每条信息的产生频率检测时间复杂度为 C_F,上下文相关度计算的复杂度为 C_R.最坏情况下,信息产生频率频繁波动,每隔一条信息就要计算一次相关度,此时,SPFC 算法的时间复杂度为 $O(|S|\times(C_F+C_R)/2)$.

4 实 验

SPFC 去掉上下文相关度计算部分,即为基于信息产生频率的会话抽取算法 SP_F;去掉信息产生频率检测的部分,即为基于上下文相关度的会话抽取算法 SP_C.我们使用 ICTCLAS[16]对文本信息进行分词,使用 Lucene[17]构建训练语料的倒排索引并提供检索,实现上述提出的 3 种算法及基准算法.实验结果以 SP_{WC},SP_{NN},SP_{WNN}[5]为

基准算法,以 F 度量值为评价指标,对比验证 SP_F,SP_C 和 SPFC 的性能与时间开销.此外,本节还测试参数变化对算法的影响,并分析参数对算法性能产生影响的原因.

3 种基准算法是目前比较有效的基于文本相似度进行改进的文本信息流会话抽取算法.其中,SP_{WC} 采用中心向量法,根据信息出现的时间顺序对特征向量进行加权计算得到各会话的中心向量后与待判定的信息一起计算相似度;SP_{NN} 用待判定信息与各会话的 k 个最近邻信息的相似度作为其与会话的相似度;SP_{WNN} 与 SP_{NN} 相似,但在计算待判定信息与会话的 k 个最近邻信息的相似度时,根据信息的时间顺序对特征向量进行加权.

4.1 实验数据

实验数据采集自我们的 QQ 群聊天记录.我们人工标注出一个名为 Linux 技术交流的群的部分聊天记录文本会话,形成实验数据集 D.其余未标注的聊天记录(来自多个 QQ 群)组成训练集 G,作为 SP_C 和 SPFC 的训练语料.D 的起始时间为 2009 年 10 月 5 日,结束时间为 2009 年 11 月 5 日;G 的所有信息流的结束时间都早于 D 的起始时间.数据的语言大部分为中文,夹杂着一些英文和网络非正规语言.过滤掉一些只包含杂乱符号的信息后,D 剩下 44 991 条信息,G 剩下 126 027 条信息.

为验证各种算法的通用性,我们将 D 拆分成如表 1 所示的 $D1$ 和 $D2$ 两个子数据集,分别测试各种算法在 $D1$ 和 $D2$ 这两个不同数据集上的性能,以及 $D1$ 和 $D2$ 的并集,即数据集 D 上的性能.3 个数据集的信息平均长度都在 13 个字符左右,是一个典型的短文本信息流.

Table 1 Data set

表 1 测试数据集

数据集	D	$D1$	$D2$
信息条数	44 991	22 011	22 980
会话个数	321	131	190
时间范围	10 月 5 日~11 月 5 日	10 月 5 日~10 月 20 日	10 月 21 日~11 月 5 日
平均信息长度	13	12	14

4.2 评测方法

评测方法采用与文献[5]相似的准确率、召回率和 F 度量值.首先,对于算法抽取出的每个文本会话,我们计算其与人工标注的各个文本会话之间的准确率、召回率以及 F 度量的值.具体来说,对于算法抽取出的会话 j 和人工标注的真实会话 i,准确率 $Precision$、召回率 $Recall$ 和 F 度量的值由以下 3 个公式给出:

$$Recall(i,j)=n_{ij}/n_i \tag{10}$$

$$Precision(i,j)=n_{ij}/n_j \tag{11}$$

$$F(i,j)=\frac{2\times Precision(i,j)\times Recall(i,j)}{Precision(i,j)+Recall(i,j)} \tag{12}$$

其中,n_{ij} 是指真实会话 i 和算法抽取的文本会话 j 之间相同的信息的条数,n_i 是指真实会话 i 的信息条数,n_j 是指算法抽取的会话 j 的信息条数.$F(i,j)$是指真实会话 i 和算法抽取的会话 j 之间的 F 度量值.

算法会话抽取结果总的 F 度量值由下式算出:

$$F=\sum_i \frac{n_i}{n}\max_j(F(i,j)) \tag{13}$$

其中,max 函数扫描所有检测结果,查找与真实会话 i 有最大 F 度量值的会话 j;n 是指信息的总条数.

4.3 实验结果

以 0.04 的间隔遍历 t_{sim} 的各种取值,用贪心策略调整 t_W 和 d_W 的值,测试基准算法 SP_{WC},SP_{NN},SP_{WNN} 的性能.表 2~表 4 列出了 F 度量达到最大值的部分测试结果,同等性能时,t_W 和 d_W 取最小值.结果表明,SP_{WNN} 性能最好,SP_{NN} 次之,SP_{WC} 最差.最佳 t_{sim} 值远小于文献[5]的实验结果,是因为本文的测试数据来自真实的网络聊天记录,文本信息的特征稀疏性、奇异性更为突出,导致了文本信息间的相似性很差.

Table 2 Performance of base-line algorithms on *D*

表 2 数据集 *D* 上基准算法性能

t_{sim}	SP_{WC}			SP_{NN}			SP_{WNN}		
	F	t_W	d_W	F	t_W	d_W	F	t_W	d_W
0.002	**0.453**	4	17	**0.471**	3	19	**0.486**	6	11
0.004	0.444	4	17	**0.471**	3	19	0.487	6	16
0.006	0.436	4	17	0.467	3	19	0.478	6	19
0.008	0.427	4	16	0.462	3	19	0.469	9	16

Table 3 Performance of base-line algorithms on *D*1

表 3 数据集 *D*1 上基准算法性能

t_{sim}	SP_{WC}			SP_{NN}			SP_{WNN}		
	F	t_W	d_W	F	t_W	d_W	F	t_W	d_W
0.002	**0.446**	4	17	0.456	3	15	0.484	4	14
0.006	0.431	4	17	0.457	3	15	**0.493**	9	14
0.010	0.417	4	19	**0.459**	4	19	0.458	9	14
0.014	0.403	7	20	0.444	4	17	0.416	8	8

Table 4 Performance of base-line algorithms on *D*2

表 4 数据集 *D*2 上基准算法性能

t_{sim}	SP_{WC}			SP_{NN}			SP_{WNN}		
	F	t_W	d_W	F	t_W	d_W	F	t_W	d_W
0.002	**0.461**	4	9	**0.485**	3	19	**0.489**	5	20
0.004	0.458	4	18	**0.485**	3	19	0.486	3	20
0.006	0.446	4	10	0.477	3	19	0.484	5	19
0.008	0.443	4	13	0.466	3	19	0.472	5	13

SP_F 的测试结果见表 5~表 7,性能比 SP_{WNN} 提高了 27.9%.表 5~表 7 所示参数值的单位为 s;单元格的数值是 ω和Δt 取相应表头的值时,测试取得的 F 值.SP_F 算法是一种边界检测算法,其性能与会话窗口 t_W、文档窗口 d_W 的大小无关,但与Δt 和ω的取值关系密切.当$\Delta t \leqslant 135$ 时,信息产生频率采样的时间窗口较小,产生频率的变化较大,导致会话切分过细,性能表现不佳.测试还发现,当采样时间窗口大于 3min 小于 20min 时,该参数的变化对算法性能影响不大,且性能较好.这说明,这个 QQ 群的大部分会话持续了 20min 以上.ω是计算信息产生频率移动平均值的时间窗口,从测试结果来看,ω取值在 40~60 期间,性能较佳.

Table 5 Performance of SP_F on *D*

表 5 数据集 *D* 上 SP_F 算法性能

ω / Δt	20	40	60	80	100	120
135	0.575	0.584	0.595	0.593	0.601	0.604
225	0.590	0.605	0.613	0.608	0.604	0.599
315	0.587	0.611	0.610	0.598	0.597	0.605
405	**0.612**	**0.622**	0.618	**0.612**	0.606	0.610
495	0.604	0.602	0.615	0.603	**0.610**	0.604
585	0.600	0.607	**0.620**	0.606	**0.610**	0.599

Table 6 Performance of SP_F on *D*1

表 6 数据集 *D*1 上 SP_F 算法性能

ω / Δt	20	40	60	80	100	120
135	0.606	**0.624**	0.622	0.600	0.613	**0.620**
225	0.617	0.617	**0.626**	0.611	0.602	0.600
315	0.602	0.617	0.623	0.604	0.601	0.602
405	**0.619**	0.617	0.601	0.602	0.594	0.602
495	0.611	0.594	0.604	0.585	0.604	0.593
585	0.603	0.600	0.625	**0.616**	**0.623**	0.617

Table 7 Performance of SP_F on *D*2

表 7 数据集 *D*2 上 SP_F 算法性能

ω / Δt	20	40	60	80	100	120
135	0.552	0.563	0.589	0.596	0.600	0.597
225	0.589	0.616	0.621	0.617	**0.616**	0.607
315	0.592	0.620	0.607	0.604	0.597	0.611
405	**0.616**	**0.632**	**0.631**	**0.618**	0.611	**0.614**
495	0.608	0.615	0.620	0.616	0.605	0.602
585	0.607	0.616	0.617	0.594	0.589	0.578

基于训练语料 *G* 测试 SP_C 算法的性能,在 3 个数据集上的平均性能比 SP_{WNN} 高 10.9%.同样,采用贪心策略来调整 SP_C 的各项参数,*F* 值达到最大时各项参数见表 8.

Table 8 Performance of SP_C

表 8 SP_C 算法性能

F(*D*)	*F*(*D*1)	*F*(*D*2)	t_W	d_W	ζ	μ	*k*
0.539	0.534	0.540	2	19	0.02	16	7
0.522	**0.546**	0.504	2	19	0.024	4	6
0.533	0.519	**0.548**	2	14	0.04	16	6
0.530	0.512	0.540	2	20	0.056	16	6

SP_C 比基准算法对参数变化有较好的适应性.当相关度阈值ζ的取值在[0.02,0.056]期间时,SP_C 的性能变化不大,*F* 值均能保持在 0.5 以上,不似基准算法对 t_{sim} 那样十分敏感,因此,更具实用性.当前导信息窗口 *k* 为 6 时,各数据集上的效果最佳.这与无监督训练语料上会话的平均信息条数有关.当搜索窗口μ=4,ζ=0.024 时,数据集 *D*1 上的性能最佳,μ值变大后数据集 *D*1 的性能略有下降,但只要相关度阈值合适,仍能达到较好的性能.图 3 展示了ζ=0.056 时,μ值变化对性能的影响.其中,横轴为搜索窗口μ,纵轴为 *F* 度量值.μ大于 10 后,其改变对性能的影响不大.当 3 种基准算法和 SP_C 算法取得最大性能时,t_W 都在 2~4 左右,说明在大多数情况下,交错进行的会话不超过 4 个,这与我们对数据集的观察相一致.当 d_W>10 时,d_W 的改变对算法性能影响不大.为了使会话的特征向量更加稠密,d_W 通常设为一个较大的值.

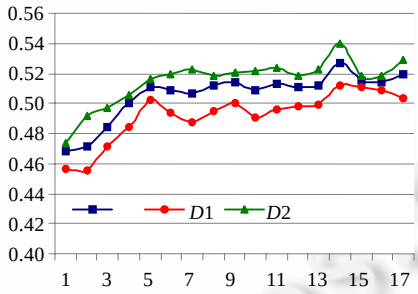

Fig.3 Performance with different search window μ

图 3 搜索窗口μ变化对性能的影响

SPFC 性能比 SP_{WNN} 高 30%,略高于 SP_F.*F* 度量取得最大值时的各项参数指标见表 9.

Table 9 Best performance of SPFC

表 9 SPFC 的最佳性能

数据集	*F*	Δt	ω	t_W	d_W	ζ	μ	*k*
D	0.629	1 140	40	3	20	0.48	13	8
*D*1	0.645	1 140	40	2	17	0.1	6	2
*D*2	0.636	1 140	40	3	29	0.54	19	8

SPFC 算法相比 SP_F 和 SP_C 的优势在于,它既利用了时序特性,又能处理会话的交错性.如第 3.3 节所述,SP_F 算法只考虑利用时序特性判定会话边界,所以即使参数设置合适,面对交错性严重的数据,其召回率仍然成为影响性能的瓶颈.应用 SPFC 时,应将时间窗口参数设置为较小的值,使得根据时序特征切分时能够得到较细粒度的初步会话,再利用相关度聚合,从而达到比 SP_F 和 SP_C 更高的性能.相反,如表 10 所示,若时间窗口属性设得较大,交错的会话无法被检测到,准确率下降,导致 F 值降低.

此外,SPFC 对参数变化具有更好的适应性.因为信息产生频率的波动并不一定是会话边界,SP_F 算法对频率的微小波动进行了平滑,所以ω参数的变化对 SP_F 的性能影响较大.SPFC 在 SP_F 的基础上引入了内容相关度的判定,对 SP_F 切分的会话进一步进行聚合.所以,即使Δt,ω参数取值不当,对产生频率的变化过分敏感,将信息流切分成过碎的片段,但 SPFC 算法可以将这些片段进行聚合,仍可达到较好性能.实验测试ω>10,ω<Δt<1140 时的各种组合,发现Δt,ω对性能几乎没有影响.t_W取 2 或 3 时,d_W取大于 20 的任何值,对算法性能也几乎没有影响.SPFC 算法的性能表现主要取决于相关度阈值ζ.数据集 $D1$ 最合适的相关度阈值ζ是 0.1,而数据集 $D2$ 最合适的相关度阈值ζ是 0.54,相差较大.这是因为 $D2$ 中的信息长度比 $D1$ 长(见表 1),内容更加丰富完整,所以 $D2$ 上的信息更容易在 G 中找到相似信息,信息间的相关度普遍较高,所以需要一个较高的相关度阈值;否则,SPFC 会将大部分会话片段聚合起来,导致准确率下降.不过,即使参数的取值不当,SPFC 也有不错的表现,性能仍然明显高于 SP_C 和 SP_{WNN} 的最佳性能.由于人们日常的对话中同时进行的会话不可能太多,t_W值可设为 2,设 d_W为任意大于 20 的值.测试其他参数的所有组合,SPFC 在各数据集上的最差性能及相关参数见表 10.

Table 10 Worst performance of SPFC

表 10 SPFC 的最差性能

数据集	F	Δt	ω	t_W	d_W	ζ
D	0.562	1 140	190	8	10	0.08
$D1$	0.602	1 140	190	2	6	0.04
$D2$	0.557	135	40	2	6	0.08

随着文本信息流规模的增长,各算法的时间开销呈线性增长.图 4 展示了各种算法的时间开销.横轴为信息条数,纵轴为时间(单位为 ms).SP_C 算法由于要频繁搜索训练语料,速度最慢;SPFC 的耗时大于 SP_{WNN},但远小于 SP_C,处理 1 万条信息只需 4s 左右时间.

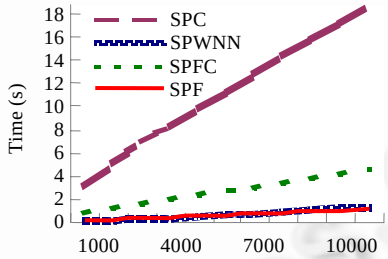

Fig.4 Overhead of each algorithm

图 4 各种算法的时间开销

图 5 对比了 3 种基准算法以及本文提出的 SP_F,SP_C,SPFC 这 3 种算法的最佳性能对比.可见,SPFC 算法性能最优.上述关于参数的讨论表明,SP_F 对参数设置较为敏感,SP_C 计算量较大.SPFC 则综合了两者的优势,具有较高的实用性,适用于高速文本信息流的处理.

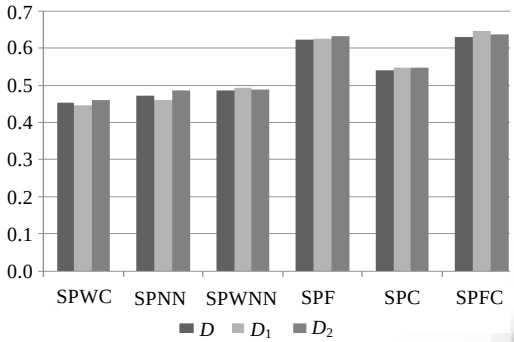

Fig.5 Best performance of each algorithm

图 5 各种算法最佳性能对比

5 结论和展望

网络聊天、微博数据等短文本信息流的会话抽取,是短文本信息流挖掘的一项重要任务.本文给出了文本会话抽取的定义,从文本信息流的时序特性和文本会话上下文相关性两个角度出发,发现了文本信息流中会话的生命周期性和相邻信息的上下文相关性规律,提出了基于信息产生频率的会话边界检测方法和基于上下文相关度的会话抽取方法,以及这两种方法的综合算法 SPFC.SPFC 是一种无监督的机器学习算法,它从历史语料中学习特征项间的会话相关度.在中文数据集上进行的大量实验表明所提方法是有效的,SPFC 的性能比基于文本相似度的方法提高了 30%.并且算法的时间开销较小,适用于高速文本信息流的在线处理.

但是,文本会话抽取的性能还有很多改进空间:首先,SP_F 算法假设大部分信息产生频率的突变代表了会话边界,但仍有一些会话主题在信息产生频率持续上升或下降时产生了漂移,SP_F 算法暂时没有考虑这类会话边界;其次,SPFC 算法是在 SP_F 算法切分的会话上利用上下文相关度对文本会话进行进一步聚合,对于交错性特别严重的信息流,仍有可能无法完全正确地区分交错的会话.下一步,我们将尝试改进 SP_C 算法的性能,引入现有的知识库,提高上下文相关度计算的准确率;尝试改进 SP_C 算法的时间开销,使 SPFC 算法在每条信息到达时都进行一次相关度判断,从而检测出不符合信息产生频率突变性质(即第 3.1 节所述的性质 1)的会话边界.

致谢 感谢北京邮电大学方滨兴院士对本文工作的支持和指导,感谢国防科学技术大学计算机学院网络中间件教研室文本挖掘课题组全体同学对本文工作的建议和帮助.

References:

[1] Bengel J, Gauch S, Mittur E, Vijayaraghavan R. Chattrack: Chat room topic detection using classification. Lecture Notes in Computer Science, 2004,3073:266−277.

[2] Adams P, Martel C. Conversational thread extraction and topic detection in text-based chat [MS. Thesis]. Monterey: Naval Postgraduate School, 2008. [doi: 10.1002/9780470588222.ch6]

[3] Xia YQ, Huang JH, Zhang J. Toward anomalous and dynamic nature of the Chinese network chat language. Journal of Chinese Information Processing, 2007,21(3):83−91 (in Chinese with English abstract).

[4] Smith M, Cadiz JJ, Burkhalter B. Conversation trees and threaded chats. In: Proc. of the 2000 ACM Conf. on Computer Supported Cooperative Work. New York: ACM Press, 2000. 97−105. [doi: 10.1145/358916.358980]

[5] Shen D, Yang Q, Sun JT, Chen Z. Thread detection in dynamic text message streams. In: Proc. of the 29th Annual Int'l ACM SIGIR Conf. New York: ACM Press, 2006. 35−42. [doi: 10.1145/1148170.1148180]

[6] Grice HP. Logic and conversation. In: Cole P, Morgan JL, eds. Syntax and Semantics, Vol.3: Speech Acts. New York: Academic Press, 1975. 41−58.

[7] Allan J, Carbonell J, Doddington G, Yamron J, Yang YM. Topic detection and tracking pilot study final report. In: Proc. of the DARPA Broadcast News Transcription and Understanding Workshop. 1998. http://repository.cmu.edu/compsci/341/

[8] Yang YM, Pierce T, Carbonell J. A study of retrospective and on-line event detection. In: Proc. of the 21st Annual Int'l ACM SIGIR Conf. New York: ACM Press, 1998. 28−36. [doi: 10.1145/290941.290953]

[9] Passonneau RJ, Litman DJ. Discourse segmentation by human and automated means. Computational Linguistics, 1997,23(1): 103−139.

[10] Galley M, McKeown K, Fosler-Lussier E, Jing HY. Discourse segmentation of multi-party conversation. Association for Computational Linguistics, 2003,29(1):562−569. [doi: 10.3115/1075096.1075167]

[11] Hearst MA. TextTiling: Segmenting text into multi-paragraph subtopic passages. Computational Linguistics, 1997,23(1):33−64.

[12] Wang L, Jia Y, Chen YW. Conversation extraction in dynamic text message stream. Journal of Computers, 2008,3(10):86−93.

[13] Wang YC, Joshi M, Cohen W, Rose CP. Recovering implicit thread structure in newsgroup style conversations. In: Proc. of the Association for the Advancement of Artificial Intelligence, ICWSM. 2008. 152−160.

[14] Wu TH, Khan FM, Fisher TA, Shuler LA, Pottenger WM. Error-Driven Boolean-logic-rule-based learning for mining chat-room conversations. Technical Reports, Lehigh CSC, 2002.

[15] Lin C, Yang JM, Cai R, Wang XJ, Wang W, Zhang L. Simultaneously modeling semantics and structure of threaded discussions: A sparse coding approach and its applications. In: Proc. of the SIGIR 2009. New York: ACM Press, 2009. 131−138. [doi: 10.1145/1571941.1571966]

[16] Zhang HP, Yu HK, Xiong DY, Liu Q. HHMM-Based Chinese lexical analyzer ICTCLAS. In: Proc. of the 2nd SIGHAN Workshop on Chinese Language Processing, Vol.7. New York: ACM Press, 2003. 184−187. [doi: 10.3115/1119250.1119280]

[17] Hatcher E, Gospodnetic O, Mccandless M. Lucene in Action. Manning Publications, 2009. http://www.imamu.edu.sa/dcontent/IT_Topics/java/luceneinaction.pdf

附中文参考文献:

[3] 夏云庆,黄锦辉,张普.中文网络聊天语言的奇异性与动态性研究.中文信息学报,2007,21(3):83−91.

黄九鸣(1981－),男,福建安溪人,博士生,CCF 学生会员,主要研究领域为文本挖掘,信息检索.

张旭(1983－),女,硕士,主要研究领域为互联网信息安全.

吴泉源(1942－),男,教授,博士生导师,主要研究领域为人工智能,分布计算,数据库管理.

贾焰(1960－),女,博士,教授,博士生导师,CCF 高级会员,主要研究领域为数据库,数据挖掘,人工智能.

刘春阳(1962－),男,博士,副研究员,主要研究领域为互联网信息安全.

周斌(1971－),男,博士,副研究员,CCF 会员,主要研究领域为文本挖掘,分布计算,Web 服务.