基于主题模型的 Mashup 标签推荐方法

刘建勋 石 敏 周 栋 唐明董 张婷婷
(湖南科技大学知识处理与网络化制造湖南省普通高校重点实验室 湖南 湘潭 411201)

摘 要 Web 2.0 时代,标签作为 Web 资源管理和检索的有效方式已成为近年的热点研究对象.开发者通常为新的 Mashup 人工指定若干与功能性相关的标签,以便于用户理解、检索以及实现 Mashup 资源的分类管理.然而,手动指定标签十分繁琐且费时,自动生成 Mashup 标签十分必要但缺乏有效方法.针对该问题,文中提出一种基于主题模型的方法进行 Mashup 标签的自动推荐.该方法首先建立 Mashups 与 Web Application Programming Interfaces(APIs)的描述文档以及 Mashups 与 APIs 之间的组合关系模型,然后寻找与待推荐标签 Mashup 的描述文档主题分布相似的 Web APIs,并将它们与该 Mashup 直接组合的 APIs 合并,采用一种带权重的 PageRank 算法,从中挑选出最重要的 APIs,最后将它们已有标签推荐给该 Mashup.同时,针对所提方法文中设计实现一种标签排序算法,该算法优先推荐那些与 Mashup 主题最相关的标签.根据使用从 ProgrammableWeb 收集的真实数据进行实验可知,文中所提出的方法明显优于其他自动化标签推荐方法.

关键词 Mashup;Web APIs;标签推荐;主题模型;PageRank;云计算
中图法分类号 TP301 **DOI 号** 10.11897/SP.J.1016.2017.00520

Topic Model based Tag Recommendation Method for Mashups

LIU Jian-Xun SHI Min ZHOU Dong TANG Ming-Dong ZHANG Ting-Ting
(*Hunan Provincial College Key Laboratory of Knowledge Processing & Networked Manufacturing, Hunan University of Science & Technology, Xiangtan, Hunan* 411201)

Abstract In the Web 2.0 era now, tags served as an effective way for web resources management and retrieval, has been a hotspot research topic. Developers usually associate manually several relevant tags to the newly created Mashup, which makes it easy to understand and retrieve for Mashup users, as well as facilitates the classification and management of Mashup resources. However, it is extraordinary time-consuming and tedious to create manual tags for new Mashups. There have not an effective method which can recommend relatively accurate tags for new Mashups automatically so far though it is very important. In this paper we propose a method for Mashup tag recommendation based on a topic model. The model simultaneously takes the description documents for Mashups and Web Application Programming Interfaces (APIs) as well as the composition relationships between them into account. Based on the model, our approach first selects the most similar APIs of the target Mashup. Subsequently, those chosen similar APIs and composed APIs of this Mashup are combined into a single APIs set. We select several most important APIs from this APIs set based on a weighted PageRank algorithm. Finally, tags of these important APIs are recommended to the Mashup. We also design an algorithm to rank tags

收稿日期:2016-01-08;在线出版日期:2016-07-02.本课题得到国家自然科学基金(61572187,61300129,61272063,61572186)、国家科技支撑计划(2015BAF32B01)、教育部留学回国人员科研启动基金(教外司留[2013]1792)、湖南省教育厅资助科研项目(16K030)、湖南省研究生科研创新项目(CX2016B573)资助.**刘建勋**,男,1970 年生,博士,教授,博士生导师,主要研究领域为服务计算与云计算、工作流管理的理论与应用等.E-mail:ljx529@gmail.com.**石 敏**,男,1991 年生,硕士研究生,主要研究方向为信息检索与服务计算.**周 栋**,男,1979 年生,副教授,硕士生导师,主要研究方向为信息检索、自然语言处理、机器学习等.**唐明董**,男,1978 年生,教授,硕士生导师,主要研究领域为服务计算与云计算.**张婷婷**,女,1991 年生,硕士研究生,主要研究方向为服务计算与云计算.

recommended according to their topic relevance with the target Mashup. The experimental results on a real world dataset collected from ProgrammableWeb prove that our approach obviously outperforms other tag recommendation methods.

Keywords　Mashup; Web APIs; tag recommendation; topic model; PageRank; cloud computing

1　引　言

随着 Web 2.0 技术的兴起，标签(Tag)作为资源管理和检索的有效方式成为近些年的热点研究对象[1-4]. 互联网上的各种资源比如网页、文献、博客、图片、音乐、视频等通常允许作者或其他用户为其关联若干描述性的词汇，也就是标签. 标签和关键词相似，都体现了对标注资源的描述和概括. 但是，标签是用户根据自己的理解为资源指定的描述性词汇，它具有随意性和语义模糊性等特点，这为标签的自动推荐带来了巨大的挑战[1-2].

近些年，许多网站如 ProgrammableWeb① 和 Seekda② 已经允许用户为 Mashup 资源关联若干标签，以便于人们理解、检索以及实现 Mashup 资源的分类管理. 如在 ProgrammableWeb 中，为 Mashup 指定若干相关的标签将有助于用户快速检索出需要的 Mashup 服务. 图 1 所示为 ProgrammableWeb③ 上名为“Mashit”的 Mashup 实例，该 Mashup 有 4 个标签和一个描述文档. 为进一步说明标签的作用，取其中两个标签“Color” 与 “Fonts”作为关键词检索 ProgrammableWeb 的 Mashups 资源库，图 2 所示为检索结果. 从结果可以看出，名称为“Mashit”的 Mashup 排在最前面. 然而，人工为 Mashups 资源指定标签是一件比较繁琐且浪费时间的任务. 因此需要一种方法为 Mashups 资源自动推荐标签，同时规范标签的使用. 目前已经存在一些为 Web 服务进行标注的方法[3-6]. 在这些标签推荐方法中，基于词共现的方法被广泛使用[7-8]. 例如为了解决 Web 服务标签分配不均匀等问题，Chen 等人[7-8] 使用基于词共现的方法为标签数较少的 Web 服务扩充标签. 基于词共现的方法挖掘已有标签数据中标签之间的关联关系，用以有效地扩充 Web 服务的现有标签. 但当服务标签较少或者当 Web 服务没有标签时，基于词共现的方法通常工作效果不好[9-10]. 基于聚类的方法也被用于 Web 服务的标签推荐[4,6,11]，例如 Fang 等人[4] 提出了一种自动 Web 服务标签推荐方法，该方法使用了两种策略：标签扩展和标签抽取. 他们首先基于 WSDL 文档将 Web 服务进行聚类，然后将同一个类中其他服务的标签推荐给指定服务. 但是仅仅使用基于 WSDL 的匹配技术忽略了文档的隐含语义信息[6] 且存在词汇稀疏的问题[12]. 因此有方法研究基于主题模型来发现 Web 服务的隐含主题信息. 例如 Aznag 等人[5] 首先基于主题模型从 Web 服务中抽取出若干候选标签，随后这些标签被用于训练一个分类器，然后基于该分类器将最相关的标签推荐给新的服务. 然而，大部分基于主题模型的标签推荐方法仅仅依赖于 Web 服务的文本描述信息[1-2,4]. 当描述信息较少或者标签矩阵很稀疏时，往往无法取得较好的效果[9]. 因此需要其他辅助信息以提高推荐标签的准确率. 众所周知，Web 服务之间往往存在链接(或组合)关系. 研究表明，如果两个文档之间存在链接关系，它们则很可能具有相似的主题分布信息[13-14]，而隐含主题分布相似的 Web 服务则表明它们的功能性描述文档相似，进一步说明它们可能共享某些相同或者相似的标签. 所以，在服务描述信息或者标签信息较少时，充分利用这些服务之间构成的网络信息有助于提高标签推荐的准确度. 基于此，本文提出了一种同时将 Mashups 文档描述信息以及 Mashups 与 APIs 之间的链接关系进行建模的方法用于 Mashups 标签的自动推荐.

图 1　ProgrammableWeb 上名为“Mashit”的 Mashup 实例

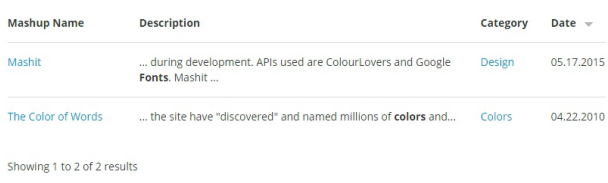

图 2　用关键词 “Color” 与 “Fonts”检索得到的结果

① http://www.programmableweb.com/category/all/mashups
② http://webservices.seekda.com/
③ http://www.programmableweb.com/mashup/Mashit

图 3 展示了 Mashups、Tags、APIs 三者之间的链接关系网，实线表示标签与 Mashups 或者 APIs 之间存在标注关系，虚线表示 Mashups 由若干 APIs 组合而成. 从图中可以看出，某些 Mashups 及 APIs 由一些共同的标签进行标注，当某一 Mashup 和某一 API 包含共同的标签时，该 Mashup 和 API 很有可能是组合关系. 以往的研究表明，当 Mashup 与 API 存在组合关系时，它们的功能性描述文档是相似的[15]. 也就是说，当某一 Mashup 和 API 具有相同或者相似的功能性描述文档时，很有可能包含有相同的标签. 基于以上观察，本文提出了一种基于主题模型的 Mashup 标签推荐方法，该主题模型同时将 Mashups 与 APIs 的描述文档以及 Mashups 与 APIs 之间的组合关系进行建模. 基于该模型，该方法首先寻找与指定 Mashup 描述文档主题分布相似的若干 APIs(可能不包含与指定 Mashup 直接组合的 APIs). 接下来，将这些 APIs 和那些与该 Mashup 直接组合的 APIs 合并为一个 APIs 集合，采用一种带权重的 PageRank 算法，从该 APIs 集合中进一步挑选出若干最重要的 APIs(最可能与指定 Mashup 共享标签的若干 APIs). 最后，将这些 APIs 已有标签推荐给该 Mashup. 下面举例说明该过程. 如图 3 所示，任务为给名称为“Where Aml. At”的 Mashup 推荐若干标签(起初该 Mashup 并没有任何标签)，首先系统选择名称为“Shooping. com”的 API 作为该 Mashup 的相似 APIs，接着将这个相似 API 与名称为“Google Maps”与“Flickr”的两个 APIs 合并为一个 APIs 集合(该实例集合中只包含 3 个 APIs，实际中可能包含多个 APIs)，显而易见，这两个 APIs 与 Mashup“Where Aml. At”是直接组合关系(虚线表示). 然后使用改进的 PageRank 算法进一步选择出最重要的 APIs，最后将这些 APIs 的标签推荐给“Where Aml. At”. 本文在寻找与 Mashup 相似的 APIs 时，使用了一种概率主题模型方法，该模型同时考虑 Mashups 的文本描述信息以及 Mashups 与 APIs 之间的链接关系. 同时，由于待推荐标签集较大，而 Mashups 真实标签数目通常有限，因此本文设计实现了一种标签排序算法用于进一步排序待推荐的标签集合，该算法优先推荐那些与 Mashup 主题最相关的标签. 使用从 Programmable-Web 收集的真实数据进行实验表明，本文所提出的方法明显优于其他标签推荐方法.

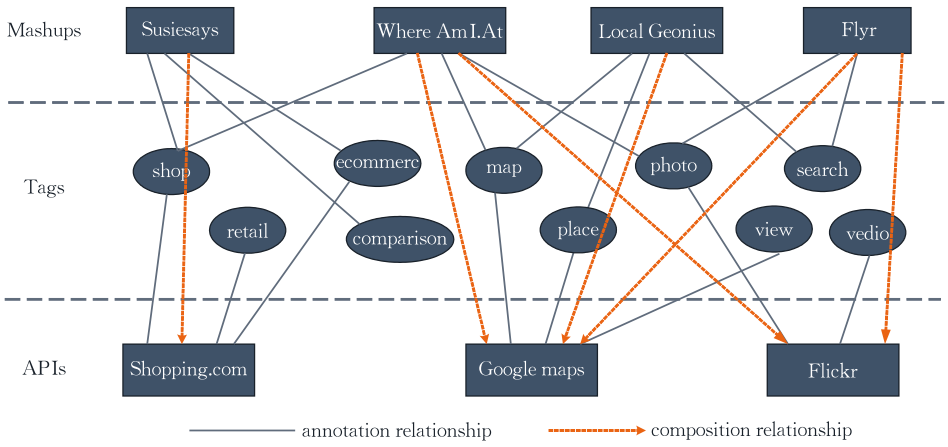

图 3 Mashups、Tags、APIs 三者之间的链接关系网

2 相关工作

本小节主要介绍相关的标签推荐技术及本文研究所涉及的 PageRank 技术.

在 Web 2.0 背景下，标签被广泛地用于对 Web 资源进行有效的组织和管理. 目前针对社会化标签推荐方面的研究较多，总体来说，可分为如下 3 类[16]：基于内容的标签推荐方法、基于词共现的标签推荐方法以及混合标签推荐方法. 基于内容的推荐方法仅仅利用物品(如 Mashup)的文本描述信息[17-18]，然后从中抽取标签. 这些方法通常能够推荐非常相关的标签，但是当物品没有丰富的文本内容时效率将不会很好[19]. 基于词共现的方法通过挖掘标签之间的共现关系来扩展物品的标签集[18,20]. 然而仅仅利用标签的共现信息不能很好地处理标签的主题漂移问题[21]. 混合的标签推荐方法同时利用物品的内容信息和物品-标签矩阵信息[9,22]，通常能取得较满意的效果. 但是当缺乏物品的内容信息或者当物品-标签矩阵很稀疏时，推荐效果通常不会很好，因此需要利用其他辅助信息如社会网络(例如 Web 服务之间的组合关系)等来进一步改善标签推

荐的准确度.在混合标签推荐方法中,充分利用物品的隐含主题信息对于提高推荐结果的准确度至关重要[9].因此基于主题模型的标签推荐方法被广泛采用[1-2,10].如 Krestel 等人[1]提出一种基于主题模型的标签推荐方法,该方法寻找与物品已有标签主题最相关的标签用于推荐.Si 等人[2]基于 Latent Dirichlet Allocation(LDA)模型提出一种新的主题模型 Tag-LDA 用于标签推荐.虽然 Tag-LDA 推荐效果优于 LDA,但是它没有考虑社会网络信息(如 Web 服务之间相互链接关系).

对 Web 服务进行标签推荐也是一个研究热点[3-6].在这些标签推荐方法中,基于词共现的方法被广泛使用[7-8].例如为了解决 Web 服务标签分配不均匀等问题,Chen 等人[7-8]基于词共现的方法为标签数较少的 Web 服务扩充标签.基于词共现的方法挖掘已有标签数据中的关联规则关系,用以有效地扩充 Web 服务的现有标签.但当服务标签较少或者当 Web 服务没有标签时,基于词共现的方法通常工作效果不好[9-10].基于聚类的 Web 服务标签推荐方法过去也被广泛采用[4,6,11].例如 Fang 等人[4]提出了一种自动 Web 服务标签推荐方法,该方法使用了两种策略:标签扩展和标签抽取.他们首先基于 WSDL 文档将 Web 服务进行聚类,然后将同一个类中其他服务的标签推荐给指定服务.但是仅仅使用基于 WSDL 的匹配技术忽略了文档的隐含语义信息[6]且存在词汇稀疏的问题[12].过去一些机器学习技术也被用于解决标签推荐的问题[5-6].例如 Aznag[5]等人首先基于主题模型从 Web 服务中抽取出若干候选标签,随后这些标签被用于训练一个分类器,然后基于该分类器将最相关的标签推荐给新的服务.然而,大部分基于主题模型的标签推荐方法仅仅依赖于物品的文本描述信息[1-2,4].当物品描述信息较少或者物品的标签矩阵很稀疏时,往往无法取得较好的效果[9].因此需要其他辅助信息,比如 Web 服务之间的组合关系,以改善推荐标签的准确度.在有些推荐系统中,物品之间常常存在着某种联系,如网页之间的互相链接的关系、Mashups 与 APIs 之间互相组合的关系以及科学论文之间的互相引用关系等.当物品之间存在链接关系时,它们的描述文档可能具有相似的主题分布[13-14].因此,充分利用上述社会网络等信息将有助于标签推荐性能的提高.

PageRank 算法主要用于计算网页的重要性,在搜索引擎中应用十分广泛[23].PageRank 算法通过研究网页的相互链接来确定网页的排序情况.其基本思想是:当网页 A 有一个链接指向网页 B,就认为 B 获得了 A 对它贡献的分值,该分值的多少取决于网页 A 本身的重要程度,即网页 A 的重要性越大,网页 B 获得的贡献值就越高.因此一个网页的重要性由 3 个因素决定[24]:链接指向该网页的网页数量、链接指向该网页的网页质量以及从这些网页链接出的网页数量.然而,传统的 PageRank 算法依据网页的链出结构,将 PageRank 值均匀分配给链出的网页节点,使得权威性高的网页不能快速上浮.Xing 等人[25]综合考虑网页的链入、链出结构提出一种带权重的 PageRank 算法(Weighted PageRank,WPR),即为网页的链入和链出赋予相应的权重因子.实验表明 WPR 改善了传统 PageRank 算法平均分配的不足,能有效改善排序的结果[26].

3　Mashup 标签推荐方法

本节首先定义标签推荐问题,然后分别介绍使用的两种主题模型技术和 PageRank 算法,继而详细描述提出的 Mashup 标签推荐方法.为了阅读方便,表 1 总结了本文所使用的常用符号及其含义说明.

表 1　常用符号及其含义说明

符号	含义说明
m	表示一个 Mashup
m'	表示一个待推荐标签的新 Mashup
a	表示一个 API
$W^{(m)}$	表示 m 的功能性描述文档
$W^{(a)}$	表示 a 的功能性描述文档
$\boldsymbol{\theta}^{(m)}$	$W^{(m)}$ 的主题分布向量
$\boldsymbol{\theta}^{(a)}$	$W^{(a)}$ 的主题分布向量
T	表示主题模型中主题的数目
$z^{(m)}$	大小为 T 的主题集合,用于 $W^{(m)}$ 中词汇的主题抽样
$z^{(a)}$	大小为 T 的主题集合,用于 $W^{(a)}$ 中词汇的主题抽样
N	语料库中词汇的总数目
$Iter$	主题模型训练时的迭代次数
$\boldsymbol{\phi}$	维度为 T 的一维向量,表示对应主题下所有词汇的概率分布
u	有向带权图中的顶点(网页或 API)
λ	本文提出的增强相似度计算公式中惩罚项的惩罚力度
$PR(u)$	顶点 u 的 PageRank 值
B_u	链向顶点 u(网页或 API)的顶点集合
d	PageRank 概率参数
t	表示一个待推荐标签
$P(t)$	标签 t 的流行度
μ	标签得分计算公式的平滑参数
A	与 m' 最相似的 API 集合,该集合可能不包含全部 Mashup 直接组合的 APIs
$tags_r$	推荐的标签集合
$tags_m$	Mashup m 的真实标签集合
S	未排序的待推荐标签集
S'	已排序的待推荐标签集

3.1　问题定义

通常一个 Mashup m 由一个简短描述文档对其提供的服务进行描述,描述文档可表示为 $W^{(m)}=\{w_1,w_2,\cdots,w_{|W^{(m)}|}\}$,其中 w_i 表示文档中第 i 个词

汇，$|W^{(m)}|$表示描述文档的词汇数目. 用于标注 Mashup 的标签集合表示为$R^{(m)}=\{t_1,t_2,\cdots,t_{|R^{(m)}|}\}$，其中 t_i表示第 i 个标签，$|R^{(m)}|$表示 Mashup m 的标签个数. 相似的，一个 API a 由一个简短的文档对其提供的功能服务进行描述，描述文档表示为 $W^{(a)}=\{w_1,w_2,\cdots,w_{|W^{(a)}|}\}$，其中 w_i 表示文本中第 i 个词汇，$|W^{(a)}|$表示描述文本的词汇数目. 用于标注 API a 的标签集合表示为 $R^{(a)}=\{t_1,t_2,\cdots,t_{|R^{(a)}|}\}$，其中 t_i 表示第 i 个标签，$|R^{(a)}|$表示 API a 的标签个数. 在对某一 Mashup m'进行标签推荐的过程中，开发者用一个简短的文本 $W^{m'}$对 Mashup 所提供的功能服务进行描述，然后 Mashup 标签推荐系统能自动为该新创建的 Mashup 推荐若干相关的标签.

3.2 概率主题模型

本文在寻找与 Mashup 文档描述相似的 APIs 时分别使用了两种主题模型技术，下面分别对它们进行详细介绍.

3.2.1 LDA 模型

LDA 模型[27]是一种文档主题生成模型，包含文档、主题和词三层结构. LDA 假设每篇文档由若干隐含的主题组成，每个主题下有一系列与主题相关的词汇. 当要生成一篇文档时，是通过以一定的概率选择某个主题，然后再以一定的概率选择主题下的某个词语. LDA 是一种非监督机器学习技术，可以用来发现大规模文档集合或者语料库中隐含的主题分布信息. 与关键词匹配技术相比较，LDA 主题模型更关注文档或语料的语义信息，它将文档归纳出若干主题，然后根据文档主题分布向量计算文档之间的相似性，因此，LDA 主题模型是一种语义层次的匹配技术. LDA 模型可以使用蒙特卡罗马尔科夫链等方法学习得到. 本文在学习 LDA 模型时使用的是吉布斯抽样方法[28].

图 4 为 LDA 的模型图. 假设 T 表示待训练主题的数目，N 表示语料库中词汇的总数目. 则 $\boldsymbol{\theta}^{(m)}$是一个维度为 T 的一维向量，表示文档 $W^{(m)}$的主题分布向量. $\boldsymbol{\phi}$是一个维度为 N 的一维向量，表示对应主题下所有词汇的概率分布. 将所有 Mashups 与 APIs 的描述文档作为输入，参数 $\boldsymbol{\theta}^{(m)}$、$\boldsymbol{\phi}$和 $z^{(m)}$可通过吉布斯抽样方法训练得到[28]. LDA 训练过程中，吉布斯抽样方法为描述文档中的每个词进行多次迭代，每次迭代都以下列式(1)为词 $W_i^{(m)}$采样一个新的主题，直到 LDA 模型的参数收敛为止.

$$p(z_i^{(m)}=j\,|\,\boldsymbol{w},\boldsymbol{z}_{\neg i}^{(m)})\propto\frac{\boldsymbol{C}_{j,\neg i}^{(W_i^{(m)})}+\beta}{C_{j,\neg i}^{(\cdot)}+N\beta}\times\frac{\boldsymbol{C}_{j,\neg i}^{(W^{(m)})}+\alpha}{C_{\cdot,\neg i}^{(W^{(m)})}+T\alpha}\tag{1}$$

其中，$\boldsymbol{C}_{j,\neg i}^{(W_i^{(m)})}$与 $\boldsymbol{C}_{j,\neg i}^{(W^{(m)})}$表示两个矩阵，分别记录了词语 $W_i^{(m)}$归入主题 j 的次数和文档 $W^{(m)}$中归入主题 j 的词汇的数目；$C_{j,\neg i}^{(\cdot)}$表示归入主题 j 的词汇数量；$C_{\cdot,\neg i}^{(W^{(m)})}$表示文档 $W^{(m)}$中所有词汇的数量；$\boldsymbol{z}_{\neg i}^{(m)}$表示除当前主题外的主题向量；$\boldsymbol{w}$表示语料库中所有的词汇向量. α和 β是狄利克雷先验的超参数，用于平滑模型. 式(1)的后验概率可以由下列式(2)和式(3)计算得出

$$p(z_i^{(m)}=j\,|\,W^{(m)})=\frac{C_j^{(W^{(m)})}+\alpha}{C_{\cdot}^{(W^{(m)})}+T\alpha}\tag{2}$$

$$p(W_i^{(m)}\,|\,z_i^{(m)}=j)=\frac{C_j^{(W_i^{(m)})}+\beta}{C_j^{(\cdot)}+N\beta}\tag{3}$$

式(2)表示文档的主题分布，式(3)表示主题的词分布. 通过式(2)、(3)可确定 Mashups 和 APIs 描述文档的主题分布及各主题下词的分布.

3.2.2 RTM 模型

关系主题模型(Relation Topic Model，RTM)最早由 Chang 等人[29]提出，是一种层次概率主题模型. RTM 不仅仅考虑文档的内容，同时还考虑文档之间的链接关系，比如科学论文之间的引用关系或以及网页之间的链接关系等. Li 等人[15]将 RTM 模型用于 API 推荐问题，该模型在训练文档主题分布的过程中同时考虑 Mashups 与 APIs 之间的组合关系. 图 5 为 RTM 的模型图. $\boldsymbol{\theta}^{(m)}$与 $\boldsymbol{\theta}^{(a)}$分别表示

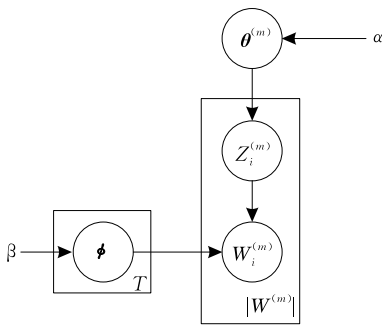

图 4 LDA 模型

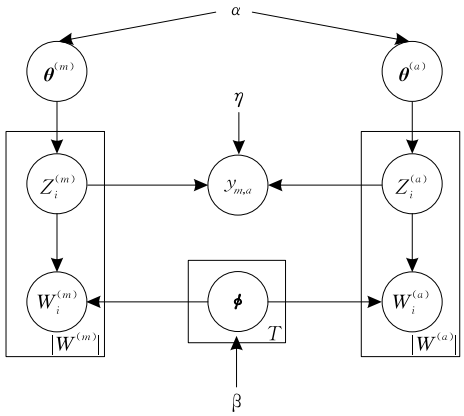

图 5 RTM 模型

Mashup 文档与 API 文档的主题分布向量. $y_{m,a}$ 是一个通过观察得到的 Mashup 与 API 文档之间的链接关系变量. 对于每一对 Mashup 与 API 文档都会有这样一个变量 $y_{m,a}$，此变量由它们的文档主题分布向量与平滑参数 η 决定. RTM 模型在训练的过程中不仅考虑 Mashups 与 APIs 的文档信息，同时在抽样文档主题时还考虑了与当前抽样文档有链接(或组合)关系文档的主题分布情况. RTM 模型的详细训练过程如算法 1 所示.

算法 1.　RTM 模型训练过程.

输入：APIs 的标签、Mashups 和 APIs 描述文档以及它们之间的组合关系

输出：$\boldsymbol{\theta}^{(m)}$，$\boldsymbol{\theta}^{(a)}$，$\boldsymbol{\phi}$，$z^{(m)}$ 和 $z^{(a)}$ 等隐含参数

1. 对于 Mashup 或 API 描述文档中的每一个隐含主题，根据 Dirichlet 分布 $Dirichlet(\beta)$ 得到该主题上的一个词多项式分布向量 $\boldsymbol{\phi}\sim Dirichlet(\beta)$.

2. 对于每一个 Mashup 描述文档，根据 Dirichlet 分布 $Dirichlet(\alpha)$ 得到一个主题分布概率向量 $\boldsymbol{\theta}^{(m)}\mid\alpha\sim Dirichlet(\alpha)$.

3. 对于每一个 API 描述文档，执行同步骤 2 过程.

4. 对于 Mashup 描述文档中每一个词汇 $w_i^{(m)}$：

(1) 从 Mashup 描述文档的主题分布概率向量 $\boldsymbol{\theta}^{(m)}$ 中为该词汇抽样一个主题 $z_i^{(m)}\mid\boldsymbol{\theta}^{(m)}\sim Multinomial(\boldsymbol{\theta}^{(m)})$；

(2) 从上述主题 $z_i^{(m)}$ 中抽样一个词汇 $w_i^{(m)}\mid z_i^{(m)}$，$\phi_{1:T}\sim Multinomial(\phi_{z_i^{(m)}})$.

5. 对于 API 描述文档中的每一个词汇 $w_i^{(a)}$，执行同步骤 4 过程.

6. 对于每一对 Mashup 与 API 描述文档，刻画一个它们之间的链接关系变量：$y_{m,a}\mid z^{(m)},z^{(a)}\sim\psi(\cdot\mid z^{(m)},z^{(a)},\lambda)$，其中 $z^{(m)}=\{z_1^{(m)},z_2^{(m)},\cdots,z_T^{(m)}\}$，$z^{(a)}=\{z_1^{(a)},z_2^{(a)},\cdots,z_T^{(a)}\}$.

$y_{m,a}=1$ 表示 Mashup 与 API 之间存在链接(或组合)关系. 通过函数 ψ 可以得到 Mashups 与 APIs 描述文档之间的相似性或者链接可能性大小. 函数 ψ 的计算依赖于 Mashups 与 APIs 描述文档的主题分布概率向量. 本文提出一种增强的余弦相似度计算公式来计算 Mashups 与 APIs 文档之间的相似性或者链接(或组合)可能性概率，定义如式(4)所示：

$$\psi(y_{m,a}=1)=\frac{\sum_{k=1}^{T}\frac{1}{e^{\lambda|z_k^{(m)}-z_k^{(a)}|}}z_k^{(m)}\cdot z_k^{(a)}}{\sqrt{(z_1^{(m)})^2+\cdots+(z_T^{(m)})^2}\sqrt{(z_1^{(a)})^2+\cdots+(z_T^{(a)})^2}}\tag{4}$$

该相似度计算公式引入一个惩罚项. 即对于两个参与计算的主题向量 $\boldsymbol{z}^{(m)}$ 和 $\boldsymbol{z}^{(a)}$，如果它们向量之间对应位置元素差越大，则惩罚也越大，参数 λ 表征该差值的惩罚程度，当设置 $\lambda=0$ 时，该相似度计算公式转变为普通的余弦相似度计算公式. 第 4 节实验证明了该改进相似度计算方法的有效性.

将所有 APIs 的标签、Mashups 和 APIs 文档以及它们之间的组合关系 $\boldsymbol{y}$ 作为输入，通过蒙特卡罗马尔科夫链方法进行抽样便可训练出 $\boldsymbol{\theta}^{(m)}$，$\boldsymbol{\theta}^{(a)}$，$\boldsymbol{\phi}$，$z^{(m)}$ 和 $z^{(a)}$ 等隐含参数. 在吉布斯抽样过程中，Mashup 描述文档中词汇主题的更新规则如式(5)所示[15]：

$$p(z_i^{(m)}=j\mid\boldsymbol{z}_{\neg i}^{(m)},\boldsymbol{w},\boldsymbol{y})\propto\prod_{a\in A}\exp\left(\frac{\eta}{|w^{(m)}|}\cdot z_j^{(a)}\right)\cdot p(z_i^{(m)}=j\mid\boldsymbol{w},\boldsymbol{z}_{\neg i}^{(m)})\tag{5}$$

其中

$$p(z_i^{(m)}=j\mid\boldsymbol{w},\boldsymbol{z}_{\neg i}^{(m)})\propto\frac{C_{j,\neg i}^{(w_i^{(m)})}+\beta}{C_{j,\neg i}^{(\cdot)}+W\beta}\times\frac{C_{j,\neg i}^{(w^{(m)})}+\alpha}{C_{\cdot,\neg i}^{(w^{(m)})}+T\alpha}\tag{6}$$

$\boldsymbol{w}$ 表示所有词汇的向量，$\boldsymbol{y}$ 表示所有 Mashups 与 APIs 之间的链接(或组合)关系，A 表示所有与 Mashup m 有组合关系的 APIs，$z_j^{(a)}$ 表示 API a 的描述文档中对应主题 j 的概率值. 平滑参数 η 表示 Mashup 与 API 描述文档之间链接关系因素的重要程度. 上述更新规则同样适用于 API 文档词汇主题的抽样，在这种情况下，A 表示与 API a 所有存在组合关系的 Mashups. RTM 模型的后验文档主题分布和主题下词的分布分别由式(2)和(3)计算得出.

3.2.3　PageRank 算法

PageRank 算法主要用于计算网页的重要性，在搜索引擎中应用十分广泛[23]. PageRank 通过网页之间的超链接关系来确定一个页面的重要性程度(等级). 当网页 A 有一个链接指向网页 B，就认为 B 获得了 A 对它贡献的分值，该分值的多少取决于网页 A 本身的重要程度，即网页 A 的重要性越大，网页 B 获得的贡献值就越高. 也就是说，一个网页的重要性由 3 个因素决定[24]：链接指向该网页的网页数量、链接指向该网页的网页质量以及从这些网页链出的网页数量. 网页的重要性称为 PageRank 值[23]. PageRank 建立在随机冲浪者模型[26]上，通常在一个浏览者浏览网页的过程当中，为了防止该浏览者跟随链接浏览网页时形成闭环，因此赋予该浏览者一个随机跳到一个新的起点网页的概率值 $(1-d)$(d 通常取值为 0.85). 对于一个网页 u，其 PageRank 值可由如下公式计算得到

$$PR(u)=(1-d)+d\sum_{v\in B_u}\frac{PR(v)}{|F_v|}\tag{7}$$

式中，B_u 表示链接指向 u 的网页集合；$|F_v|$ 表示从

u 链出的网页的数目. 通常,设置每个网页的初始 PageRank 值为 1,然后通过式(7)迭代计算各网页的 PageRank 值,直到每个网页的 PageRank 值趋于稳定.

3.3 基于主题模型的 Mashup 标签推荐方法

本小节详细介绍本文提出的方法. 如图 6 所示,针对 Mashup 的标签推荐流程分为 3 个步骤. 首先,对数据集进行预处理,筛选出 Mashups 描述文档、APIs 描述文档以及标签集等信息;接下来对上一步筛选出的信息分别采用两种主题模型进行训练;第 3 步进行标签的推荐. 当针对一个新的 Mashup m' 进行标签推荐时,首先用训练后的模型对 m' 的描述文档进行抽样,确定其主题分布情况,然后寻找与 m' 主题分布相似的 APIs,由于这些 APIs 可能不完全包含 m' 直接组合的 APIs,因此接着将这些相似 APIs 与 m' 直接组合的 APIs 合并为一个整体的 APIs 集合. 最后使用 PageRank 算法,将集合中 APIs 的标签推荐给 m'.

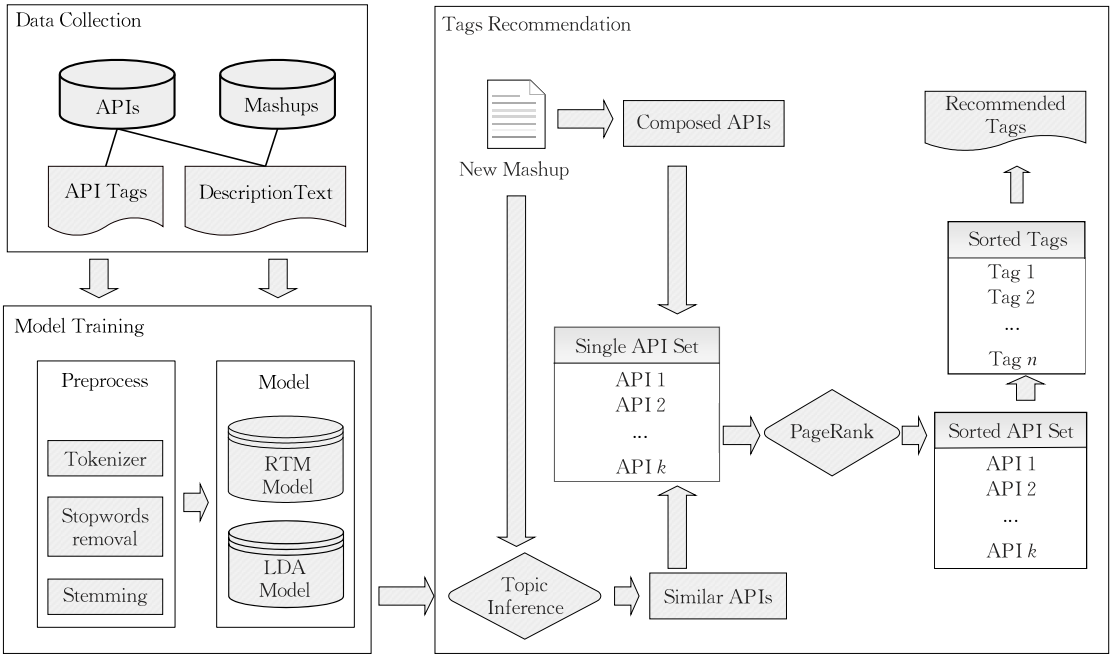

图 6 Mashup 标签推荐的执行流程

具体来说,假设通过模型训练后得到第 j 个 API 描述文档的主题分布为 $z^{(a_j)}=(z_1^{(a_j)},z_2^{(a_j)},\cdots,z_T^{(a_j)})$. 对于一个新的无标签 Mashup m'. 可以抽样得到它的主题分布向量 $\boldsymbol{Z}=(z_1^{(m')},z_2^{(m')},\cdots,z_T^{(m')})$. 然后通过式(4),获取该 Mashup 描述文档与每个 API 描述文档的相似性. 最后根据上述计算得到的相似性值对 APIs 由高到低排序. 选出 Mashup m' 的若干最相似 APIs,表示为 $A=\{a_1,a_2,a_3,a_4,\cdots\}$.

通过上述步骤,选择出了与 Mashup m' 最相似的 APIs. 基于图 3 的观察,如果 Mashup 与 API 存在直接组合关系,则说明它们的功能性描述文档相似,即它们可能会共享某些标签. 然而,上述选出的与 Mashup m' 最相似的 APIs 可能没有全部包含与 m' 直接组合的 APIs. 因此,接着将上述相似 APIs 和那些与 m' 直接组合的 APIs 合并为一个 API 集合 $A'=\{a_1,a_2,a_3,a_4,\cdots,b_1,b_2,\cdots\}$. 最后,基于一种带权重的 PageRank 算法,从 API 集合 A' 中进一步筛选出最重要的 APIs.

图 7 所示为由一个示例集合 A' 中的 APIs 构成的有向带权网络,其中网络中的顶点表示 API,一个 API A_u 指向另一个 API A_v 则表示 A_v 是一个与 A_u 主题分布相似的 API,该相似性值由式(4)计算得到. 由 A_u 指向 A_v 的边的权重由如式(8)计算得到

$$weight(A_u,A_v)=\frac{sim(A_u,A_v)}{\sum_{t\in P_u}sim(A_u,A_t)} \tag{8}$$

式中,P_u 表示由 A_u 链出的 APIs 集合;$sim(A_u,A_v)$

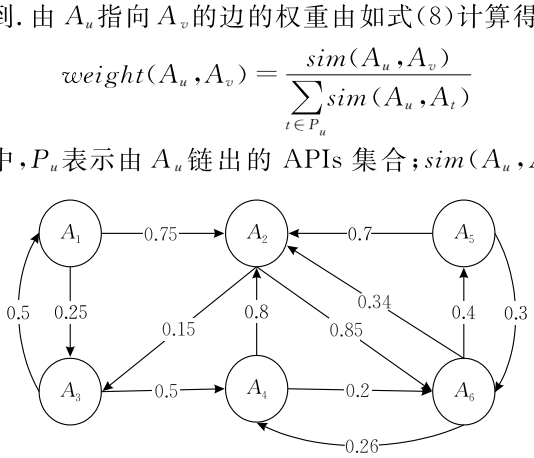

图 7 由 APIs 构成的有向带权网络

表示 A_u 与 A_v 根据式(4)计算得到的相似性值. 每条边赋予一个权重值后，每个 API 顶点的 PageRank 值可由式(9)计算：

$$PR(u)=(1-d)+d\sum_{v\in B_u}\frac{PR(v)}{|F_v|}\cdot weight(A_u,A_v) \quad (9)$$

通过式(9)，获取集合 A' 中每个 API 的 PageRank 值. 接着，将 API 依据 PageRank 值由高到低排序. 最后，选择若干 PageRank 值最高的 APIs，并将它们的标签作为待推荐的标签集合，表示为 $S=\{t_1,t_2,t_3,t_4,\cdots\}$.

通过上述步骤，已经选择出与 Mashup m' 最相关的 API 集合，并抽取它们的标签作为待推荐标签集. 但是，通过对真实 Mashup 数据集标签的统计(见第 4 节介绍)可知，待推荐标签集中的标签数目依然大于真实的 Mashup 标签数量. 因此，本文设计一种标签排序算法用于进一步过滤出最相关的标签. 该排序过程如算法 2 所示.

算法 2.　标签排序算法.

输入：未排序的待推荐标签集 $S=\{t_1,t_2,t_3,\cdots\}$，Mashup m' 的主题分布向量 $\boldsymbol{Z}=(z_1^{(m')},z_2^{(m')},\cdots,z_T^{(m')})$

输出：已排序的待推荐标签集 $S'=\{t_2,t_1,t_6,t_3,\cdots\}$

1. 将 $\boldsymbol{Z}=(z_1^{(m')},z_2^{(m')},\cdots,z_T^{(m')})$ 进行排序得到降序的序列 $\boldsymbol{Z}'=(z_3^{(m')},z_1^{(m')},z_k^{(m')},\cdots,z_T^{(m')})$(注意此时主题编号顺序发生了改变).

2. 按式(10)为集合 S 中的每个标签计算一个推荐得分：

$$Score_{t_i}=\left(1-\sin\left(\frac{\mu p_j}{p_j+1}\frac{\pi}{2}\right)\right)\cdot z_j^{(m')}\cdot t_{i,j} \quad (10)$$

3. 根据步骤 2 计算出的标签推荐得分对标签进行排序得到降序的标签集 $S'=\{t_2,t_1,t_6,t_3,\cdots\}$. 最后推荐该集合中前 M 个标签给 Mashup m'.

式(10)中，$t_{i,j}$ 表示标签 t_i 属于主题 j 的概率值；$z_j^{(m')}$ 表示集合 $\boldsymbol{Z}'$ 中属于主题 j 的元素；p_j 表示元素 $z_j^{(m')}$ 在集合 $\boldsymbol{Z}'$ 中的位置，例如元素 $z_3^{(m')}$ 在集合 $\boldsymbol{Z}'$ 中的位置为 1. 变量 μ 为平滑参数，用于协调标签推荐得分受 Mashup 主题影响的程度.

此外通过分析数据集发现有些标签被功能相似的 Mashup 频繁使用，因此认为流行的标签应该得到优先推荐. 标签的流行度由式(11)[15]计算得出

$$P(t)=\frac{N_t+c}{N_t+1} \quad (11)$$

其中，c 为控制参数，实验中取固定值 $c=0.1$. 图 8 所示为随机选取的 1000 个测试 Mashups 标签的流行度变化曲线，其中，横坐标 N_t 表示标签在所有 APIs(而不是 Mashups)标签中被使用的次数，纵坐标 $P(t)$ 表示标签流行度值. 从图中可以看出，标签被使用次数越多，其流行度越大，因此被推荐的可能性也越大. 考虑标签流行度后推荐得分由式(12)计算得出

$$Score_{t_i}=\left(1-\sin\left(\frac{\mu p_j}{p_j+1}\frac{\pi}{2}\right)\right)\cdot z_j^{(m')}\cdot t_{i,j}\cdot P(t) \quad (12)$$

通过式(4)和式(12)，本文最终推荐出的标签与 Mashup m' 既相关且比较流行. 第 4 节评估结果表明本文提出的方法能取得较好的推荐效果.

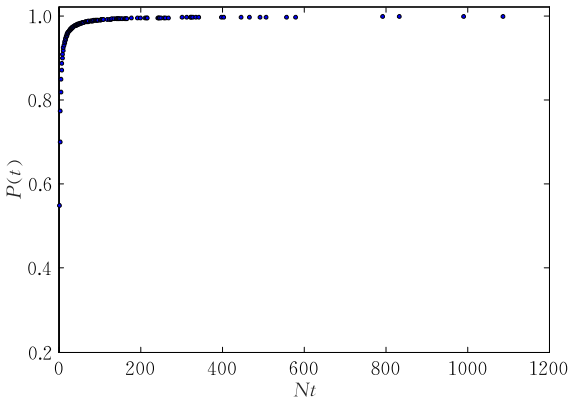

图 8　标签的流行度变化曲线

3.4　算法复杂度分析

本文提出的标签推荐算法中，主要涉及两个部分的算法：主题模型的训练过程以及通过 PageRank 算法找出最重要的 APIs. 其中主题模型 LDA 和 RTM 的训练使用的是 Gibbs 抽样方法，其训练过程中单次迭代的时间复杂度为 $O(T\cdot N)$，其中 T 表示训练的主题数目，N 表示语料库中词汇总数. 假设主题模型的迭代次数为 $Iter$，则训练 LDA 或者 RTM 模型的复杂度为 $O(T\cdot N\cdot Iter)$. 然而，LDA 和 RTM 主题模型的训练采用的是一种离线计算的方式，实际应用中，可以离线训练得到模型，然后用于在线标签推荐的过程. 因此该部分时间不在本文的考虑范围之内. 而本文采用的 PageRank 算法使用幂迭代法求解，每次迭代依次修改顶点的 PageRank 值，当 PageRank 值不再显著变化或者趋近收敛时，迭代算法结束. 由于本文 API 构成的网络顶点数目较少，因此该算法能够较快地收敛结束，其算法复杂度为 $O(k\cdot k\cdot Iterp)$，其中 k 表示网络顶点数目，$Iterp$ 表示 PageRank 算法的迭代次数.

4　实验评估

实验评估的数据来自一个提供基于互联网的 APIs 资源发布和检索的权威平台. 截止 2016 年，该

平台中 Mashup 的数量已经超过 7000，Web APIs 的数量突破 14 000 个. 本文从 ProgrammableWeb 爬取包括 Mashups 和 APIs 的描述文档及 Mashups 和 APIs 的标签信息等数据①，去掉没有标签的 Mashups 和 APIs 后，数据集包含 6693 个 Mashups 和 9122 个 APIs. 本节首先对收集的数据进行了统计分析；接着使用这些数据训练主题模型；然后使用本文设计的方法进行标签推荐；最后对实验结果进行评估并与 6 种最新水平的标签推荐方法进行分析比较.

4.1 数据集分析

图 9 所示为 6693 个 Mashups 的标签分布情况. 由图中可以看出，在真实数据中，99% 以上的 Mashups 标签数目为 1 到 6 个，因此本文在进行方法比较时只推荐 1 到 6 个标签. 本文将 6693 个 Mashups 平均分成五等份，然后采用五折交叉验证对本文提出的方法及其他方法的推荐效果进行评估，然后取五次评估结果的平均值. 每一次评估实验中，首先将测试集中 Mashups 的真实标签去掉，然后基于本文进行比较的 6 种方法为测试集中的 Mashups 分别推荐 1 到 6 个相关的标签，最后将推荐出的相关标签与 Mashups 的真实标签集进行比较，对测试集中每一个 Mashup，分别计算其 *Recall*、*Precision* 和 *F*-measure 值，最后求它们的平均值即为最终的推荐评估结果.

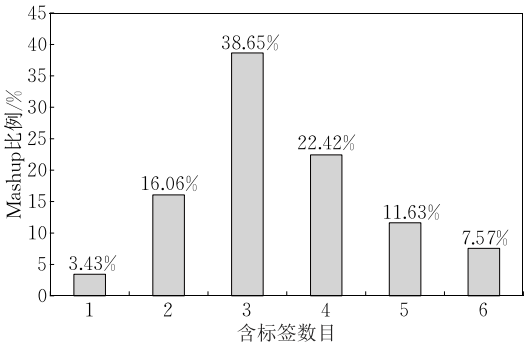

图 9 6693 个 Mashups 的标签分布情况

4.2 方法比较

本文采用 6 种方法作为基线系统进行评估和对比分析，分别介绍如下：

(1) TF[2,30]. 该方法直接从 Mashup 描述文档中抽取词频最高的关键词汇作为标签用于推荐.

(2) TF-IDF. 该方法基于描述文档的词频-逆文档频率寻找与当前 Mashup 描述文档最相似的 APIs，并将这些 APIs 的标签用于推荐.

(3) LDASA[1]. 该方法使用基本的 LDA 模型. 首先基于本文提出的增强的相似度计算方法(式(4))选择与指定 Mashup 最相似的 APIs，然后对这些相似 APIs 的标签进行排序，最后选择若干最相关的标签用于推荐.

(4) RTMSA[15]. 该方法使用 RTM 模型训练得到 Mashups 与 APIs 文档的隐含主题. 与 LDASA 方法相比，该模型在抽样主题的过程中同时考虑了 Mashups 与 APIs 文档之间的链接(或组合)关系. 该方法剩余推荐步骤同 LDASA 方法.

(5) RTMCS. 该方法使用 RTM 模型训练得到 Mashups 与 APIs 文档的隐含主题. 首先基于本文提出的增强的相似度计算方法选择与指定 Mashup 最相似的 APIs，接着将这些相似 APIs 与 Mashup 直接组合的 APIs 合并为一个整体的 APIs 集合. 然后对集合中 APIs 的标签进行排序，最后推荐若干最相关的标签. 注意该方法未使用 PageRank 算法进一步从上述合并后的 APIs 集合中寻找出最重要的 APIs.

(6) RTMPR. 本文提出的方法. 首先使用 RTM 模型训练得到 Mashups 与 APIs 文档的隐含主题. 接着基于本文提出的 PageRank 算法与排序算法，为 Mashup 推荐若干最相关的标签.

本文在训练 LDA 与 RTM 模型时设置主题数 T 为 10，迭代次数 *Iter* 为 1000，超参数 $\alpha=2.0$ 以及 $\beta=0.1$. 式(11)中控制参数取固定值 $c=0.1$. 此外，训练 RTM 模型时设置 $\eta=3$[15].

4.3 评估标准

本文采用如下评估标准：召回率、准确率以及 *F*-measure. 分别定义如下.

召回率表示推荐的相关标签占所有相关标签的比例，计算如式(13)所示：

$$Recall=\frac{|tags_r\bigcap tags_m|}{|tags_m|} \tag{13}$$

$tags_r$ 表示推荐的标签集合，$tags_m$ 表示 Mashup 真实的标签集合.

准确率表示推荐的标签集中相关标签所占的比例，计算如式(14)所示：

$$Precision=\frac{|tags_r\bigcap tags_m|}{|tags_r|} \tag{14}$$

F-measure 融合了召回率和准确率，是它们两者的调和平均值，其计算如式(15)所示：

$$F\text{-measure}=\frac{2\times Recall\times Precision}{Recall+Precision} \tag{15}$$

① http://blog.csdn.net/shimin520shimin/article/details/51209322

Recall 和 *Precision* 分别由式(13)和(14)计算得出.

4.4　实验结果

本小节首先对比分析 6 种标签推荐方法的效果.接着分析本文提出的改进的余弦相似度计算方法的性能.最后进一步分析本文所提方法中涉及到的参数对推荐结果的影响.

图 10、图 11、图 12 分别表示了本文提出的方法与 TF、TF-IDF、LDASA、RTMSA 及 RTMCS 这 5 种方法推荐性能的比较.从图中可以看出,对于本文采用的 3 种评估标准,基于主题模型的方法(包括 RTMPR、RTMCS、RTMSA 以及 LDASA)比基于关键词匹配技术的方法(包括 TF-IDF 和 TF)推荐效果好.这是由于标签通常有两个基本的特点:用户标注的随意性与标签的语义模糊性,这两种特性阻碍了推荐系统推荐出既相关且准确的标签[31].基于关键词匹配技术的标签推荐方法仅仅使用了文档或者文档之间的统计信息,这种方式可能会丢失一些比较有用的信息,比如文档的隐含主题信息等.而基于语义层次的方法充分利用了文档的隐含主题信息,在标签推荐系统中,充分利用文档的隐含主题信息对于改善推荐效果至关重要[9].这一结果同时也与前人的研究结果相一致[1-2].从图中同时可以看出,本文提出的基于主题模型的方法在推荐 1 到 6 个标签时,其召回率、准确率以及 *F*-measure 均明显优于其他 3 种方法.当推荐标签个数逐渐增加时,所有方法的召回率都不断增加,这是因为推荐的正确标签数目越来越多,同时,由于 Mashup 真实标签数目有限,可以预测召回率随着推荐标签个数的增多逐渐趋于稳定.与此相反,基本上所有方法随着标签个数的增加准确率逐渐变小,这是因为推荐了越来越多的不准确的标签.推荐 1 到 6 个标签,本文提出的方法效果平均 *F*-measure 值分别比 RTMCS、RTMSA、LDASA、TF-IDF 和 TF 的提高了 3.9%、13.6%、23.1%、78.4% 与 382.0%.

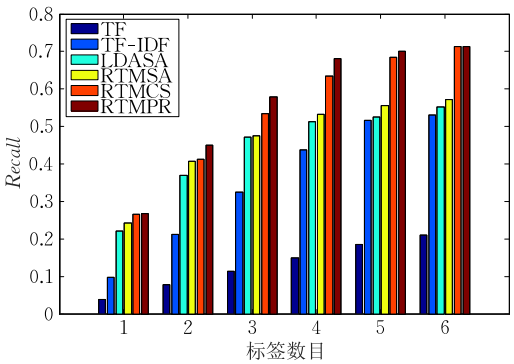

图 10　召回率比较

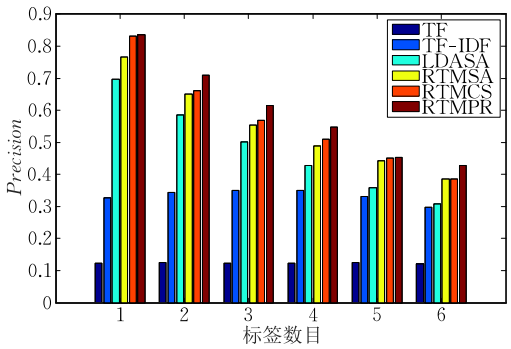

图 11　准确率比较

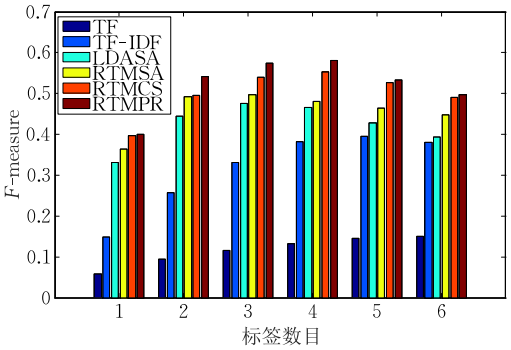

图 12　*F*-measure 值比较

为了进一步说明本文提出方法的有效性,下面将基于主题模型的方法分成 3 组比较和分析推荐效果:分别为 LDASA 与 RTMSA、RTMSA 与 RTMCS 以及 RTMCS 与 RTMPR.从图 10、图 11、图 12 可以看出,RTMSA 方法使用 3 种评估标准都比 LDASA 推荐效果好,这是因为与 LDASA 方法相比较,RTMSA 方法在抽样 Mashups 和 APIs 文档词汇主题时,同时考虑了文档之间的链接(或组合)关系.通常来说,如果两个文档之间存在链接(或组合)关系,它们则很可能具有相似的主题分布向量[13-14],对于 Mashups 与 APIs 文档而言,RTMSA 模型使得具有链接(或组合)关系的 Mashups 与 APIs 训练出的主题分布更加接近,因此会取得比 LDASA 方法更好的效果.从图中同样可以看出,使用 3 种不同的评估方法,RTMCS 方法推荐效果都比 RTMSA 好.这一结果说明如果仅仅使用第一步找出的相似 APIs 的标签进行推荐,则会忽略了 Mashup 直接组合 APIs 的标签,而在真实情况下,Mashup 与其直接组合的 APIs 也可能会共享某些标签.当然,将 Mashup 最相似的 APIs 与 Mashup 组合的 APIs 直接合并为一个整体的 APIs 集合,然后将它们的标签用于推荐也存在一定缺陷,因为该方法认为 APIs 集合中的所有 APIs 都具有相同的重要性(RTMCS).由图 10～图 12 可以看出,RTMPR 方法推荐性能比

RTMCS好,这是因为在真实世界中,Mashup的标签数目通常比较少,因此需要进一步选择集合中合适的APIs,将最重要的APIs的标签用于推荐.使用本文提出的带权重的PageRank算法能够完成这一任务.

从图10、图11、图12同时可以看出,3种基线方法中,TF-IDF方法推荐效果比TF方法要好,原因可能是部分标签只存在于Mashups标签集合中而不在APIs标签集合中,这导致TF方法无法推荐Mashups标签集中有而APIs标签集中没有的标签,因此召回率不高.6种方法中,TF方法效果最差,可能由于标签具有语义模糊性和用户标注时的随意性等特点,而简单提取关键词的方法恰恰忽略了标签的上述这些特点,从而导致召回率和准确率不高.

图13展示了本文提出的增强的文档相似度计算方法(式(4))和一般余弦相似度计算方法(式(4)中设置$\lambda=0$)之间性能的比较.图中横坐标表示推荐的标签数目,纵坐标表示F-measure值.结果表明,改进的文档相似度计算方法(图中表示为增强余弦)要优于一般的余弦相似度(图中表示为普通余弦)计算方法(式(4)中设置$\lambda=0$),即不考虑参与计算的两个向量对应元素之间值的差异性.推荐标签数目范围1到6个,本文所提方法平均F-measure提高了7.1%,这是因为改进的相似度计算方法引入了一个惩罚项,也就是说,对于两个参与计算的主题向量,如果它们对应位置元素差越大,则惩罚越大,参数λ表征该差值的惩罚程度,当设置$\lambda=0$时,该相似度计算公式变为普通的余弦相似度计算公式.实验结果证明了这一公式改进的合理性.

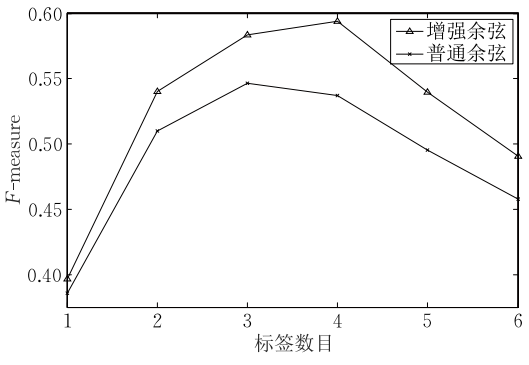

图13 相似度算法性能比较

图14展示了对标签集排序和不排序的比较结果,横坐标表示推荐的标签数目,纵坐标表示F-measure值.由图14可看出,对待推荐标签集合进行排序将显著提高推荐标签的准确度.推荐标签范围1到6个,其平均F-measure值提高了66.1%.这是因为待推荐标签集合内标签的数目通常远大于Mashup真实的标签数目(大部分为1到6个),如果简单将其作为推荐效果不佳.排序后得分高的标签优先得到推荐,即优先推荐那些与Mashup主题最相关的标签,实验结果证明标签排序算法的有效性.与此同时,与不考虑标签流行度(式(11)设置$c=1$)的推荐方法相比,考虑标签流行度在一定程度上能提高推荐标签的准确度(图15),其中纵坐标表示考虑流行度后F-measure值的提升.

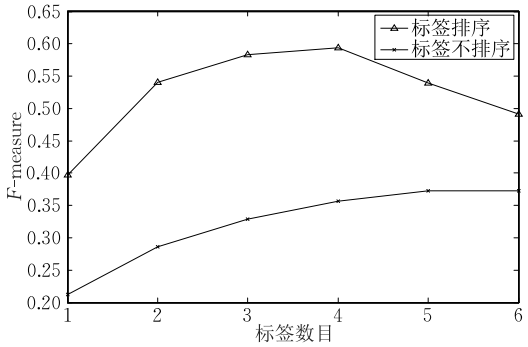

图14 标签集排序和不排序的对比

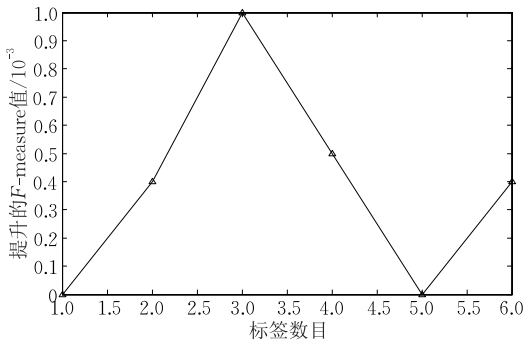

图15 考虑标签流行度的影响

综上所述,本文所提出的基于主题模型的方法明显优于传统的基于关键词匹配等的标签推荐方法.同时对标签进行排序将显著提高推荐标签的准确度,考虑优先推荐流行标签能在一定程度上提高推荐的精度.此外RTM模型比LDA模型推荐效果更好,说明考虑Mashup与API之间的历史组合关系能提高推荐标签的准确度.实验结果也表明通过PageRank算法筛选出最重要的APIs,也能进一步改善标签推荐的性能.

4.5 算法参数的影响

本文提出的增强的余弦相似度计算方法引入了平滑参数λ(式(4)),用于表示参与计算的两个向量之间对应元素差异性的惩罚权重.为了测试该参数变量的敏感度,本文设置λ的取值范围为0到5,递增区间长度为0.5.从图16中可以看出,随着λ的增加,*Recall*、*Precision*以及F-measure值随之上升

随后缓慢下降. 当设置 $\lambda=2$ 时取得的推荐效果最好. 这一变化趋势进一步表明改进的相似度计算方法的合理性. 同样为了测试式(10)中 μ 取值的变化对推荐精度的影响，实验设置 μ 取值范围为 0 到 1，递增区间长度为 0.1. 从图 17 可以看出，当 $\mu=0.3$ 时，召回率、准确率以及 F-measure 均达到峰值.

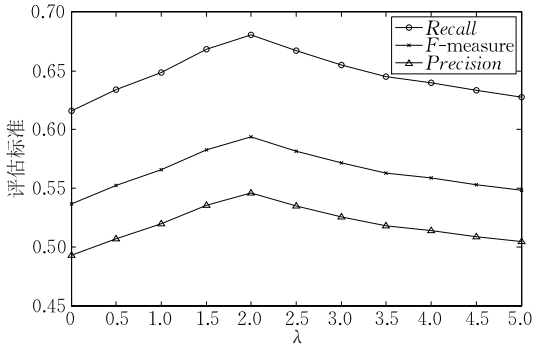

图 16　λ 的影响($tags=4$，$\mu=0.3$)

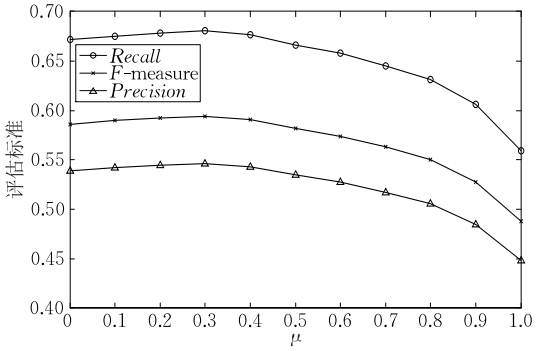

图 17　μ 的影响($tags=4$ $\lambda=2$)

4.6　标签推荐案例分析

图 18、图 19 分别表示使用本文提出的方法(RTMPR)以及基于 RTMSA 的方法为名称为“Anyvite”的 Mashup 推荐标签的详细过程. RTMPR 方法推荐标签的过程包括：根据式(4)寻找与 Mashup 最相似的 APIs 集合；将上述最相似的 APIs 与该

Mashup	名称：Anyvite		
	描述文档：anyvit straight forward servic creat onlin invit integr api googl map flickr plaxo twitter youtub		
	真实标签集：event map messag photo video		
	API 名称	API 包含的标签	与 Mashup 的相似度(递减)
与 Mashup 最相似的 APIs	Socialight	mobil map social place	0.9542672001603091
	Google Plus History	social share internet contact	0.9493900394751741
	MyNewsdesk	public relat new	0.9463777930841826
	Community Megaphone	event social	0.9373382947681377
	Napster	music stream mp3 deadpool	0.9360690002586881
	Pikeo	photo deadpool	0.9313510916666108
	API 名称	API 包含的标签	
与 Mashup 直接组合的 APIs	Flickr	photo video	
	Google Maps	map place viewer displai	
	Plaxo	offic enterpris opensoci deadpool	
	Twitter	social microblog	
	YouTube	media video	
合并后的 API 集合	Socialight，Google Plus History，MyNewsdesk，Community Megaphone，Napster，Pikeo Flickr，Google Maps，Plaxo，Twitter，YouTube		
PageRank 算法选出的最重要的 API	Socialight，Community Megaphone，Pikeo，YouTube，Flickr，Twitter		
待推荐的标签集合	mobil map social place event social photo deadpool media video photo video social microblog		
	标签	推荐得分(递减)	
标签集合中标签的推荐得分（去掉集合中重复的标签）	map	0.004139359237725237	
	photo	0.003242140711696691	
	social	0.003112135158252042	
	video	0.002696865464368347	
	event	0.0011246189806853983	
	deadpool	0.001082477395013847	
推荐标签数目	推荐标签	召回率/%	准确率/%
1	map	20	100.0
2	map photo	40	100.0
3	map photo social	40	66.6
4	map photo social video	60	75.0
5	map photo social video event	80	80.0
6	map photo social video event deadpool	80	66.6

图 18　基于 RTMPR 方法为名称为“Anyvite” 的 Mashup 推荐标签案例

Mashup	名称：Anyvite		
	描述文档：anyvit straight forward servic creat onlin invit integr api googl map flickr plaxo twitter youtub		
	真实标签集：event map messag photo video		
	API 名称	API 包含的标签	与 Mashup 的相似度(递减)
与 Mashup 最相似的 APIs	Socialight	mobil map social place	0.9542672001603091
	Google Plus History	social share internet contact	0.9493900394751741
	MyNewsdesk	public relat new	0.9463777930841826
	Community Megaphone	event social	0.9373382947681377
	Napster	music stream mp3 deadpool	0.9360690002586881
	Pikeo	photo deadpool	0.9313510916666108
待推荐的标签集合	mobil map social place social share internet contact public relat new event social music stream mp3 deadpool photo deadpool		
	标签	推荐得分(递减)	
标签集合中标签的推荐得分（去掉集合中重复的标签）	map	0.004107900439489385	
	social	0.003902603471662102	
	photo	0.003895035269152596	
	music	0.002894624395777181	
	event	0.002474687320923359	
	mobil	0.001861974022729069	
推荐标签数目	推荐标签	召回率/%	准确率/%
1	map	20	100.0
2	map social	20	50.0
3	map social photo	40	66.6
4	map social photo music	40	50.0
5	map social photo music event	60	60.0
6	map social photo music event mobil	60	50.0

图 19 基于 RTMSA 方法为名称为“Anyvite”的 Mashup 推荐标签案例

Mashup 直接组合的 APIs 合并为一个整体的 APIs 集合；接着，通过本文带权重的 PageRank 算法从合并后的 APIs 集合中进一步筛选出最重要的 APIs；然后抽取它们的标签作为待推荐标签集；最后对待推荐标签集中每个标签计算推荐得分，根据推荐得分为该 Mashup 推荐若干最相关的标签. RTMSA 方法推荐标签的过程包括：根据式(4)寻找与 Mashup 最相似的 APIs 集合；接着抽取它们的标签作为待推荐标签集；最后对待推荐标签集中每个标签计算推荐得分，根据推荐得分为该 Mashup 推荐若干最相关的标签.

从图 18 中可以看出，名称为“Anyvite”的 Mashup 真实的标签同时分布在相似的 APIs 标签集合与 Mashup 直接组合的 APIs 标签集合中，因此合并上述两个集合可以提高召回率. 而使用 RTMSA 方法，其待推荐标签集中不包含 Mashup 直接组合的 APIs 中有的标签，如“video”标签只存在于 Mashup 直接组合的 APIs 中. 这种情况下，RTMSA 方法无法推荐“video”标签给该 Mashup，因此导致推荐效果没有 RTMSA 方法好. 上述两个案例进一步说明，如果仅仅使用第一步找出的相似 APIs 的标签进行推荐，则会忽略了那些与 Mashup 直接组合 APIs 的标签，而真实情况下 Mashup 与其直接组合的 APIs 也可能会共享某些标签，因此需要合并上述两种标签集.

5 总 结

本文提出一种基于主题模型的 Mashup 标签推荐方法. 该方法由两个连续的过程组成：API 选择阶段和标签排序阶段. 首先基于 RTM 主题模型收集并通过一种带权重的 PageRank 算法找寻与 Mashup 最相关的 APIs，然后根据主题相关性对这些 APIs 的标签进行排序，最后推荐若干最相关的标签给该 Mashup. 本文分别采用 3 种评估标准，比较了 6 种标签推荐方法的性能，实验结果表明，与基于关键词匹配技术的方法相比较，基于主题模型的推荐方法具有明显的优势. 合并 Mashup 最相似的 APIs 与 Mashup 直接组合的 APIs 能提高最终的推荐标签的准确度，同时，使用本文提出的 PageRank 算法筛选出最重要的 APIs 将进一步提高标签推荐标签的准确度. 本文还设计实验比较了本文提出的增强的相似度计算方法与普通相似度计算方法的性能，结果表明，在计算两个向量的相似度时，对向量中对应

元素值的差异性引入惩罚项将显著地改善标签推荐的效果.最后,本文对相关的平滑参数(λ 与 μ)进行了敏感度测试实验,实验结果表明推荐标签的准确度对参数的变化比较敏感,因此进一步证明了本文提出方法的合理性.

未来工作将考虑融合多种社会网络关系进一步改进标签推荐的性能,如标签与 APIs 或者 Mashups 之间的标注关系以及 APIs 与 Mashups 之间的组合关系等.此外,Mashup 数量也在不断增长,未来在大数据背景下,模型的训练无疑需要更多的时间.因此考虑在开源分布式框架(如 MapReduce)上实现本文的算法也是未来感兴趣的方向.

致　谢　在此,对本文工作给予帮助与建议的老师和同学以及提出宝贵评审意见的审稿专家表示衷心的感谢!

参　考　文　献

[1] Krestel R, Fankhauser P, Nejdl W. Latent dirichlet allocation for tag recommendation//Proceedings of the 3rd ACM Conference on Recommender Systems. New York, USA, 2009: 61-68

[2] Si X H, Sun M S. Tag-LDA for scalable real-time tag recommendation. The Journal of Information & Computational Science, 2009, 6(1): 23-31

[3] Gawinecki M, Cabri G, Paprzycki M, Ganzha M. WSColab: Structured collaborative tagging for web service matchmaking //Proceedings of the International Conference on Web Information Systems and Technologies. Valencia, Spain, 2010: 70-77

[4] Fang L, Wang L J, Li M, Zhao J F. Towards automatic tagging for web services//Proceedings of the 19th International Conference on Web Services. Honolulu, USA, 2012: 528- 535

[5] Aznag M, et al. Multilabel learning for automatic web services tagging. International Journal of Advanced Computer Science & Applications, 2014, 5(8): 4910-4914

[6] Lin M, Cheung D W. Automatic tagging web services using machine learning techniques//Proceedings of the IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT). Washington, USA, 2014: 258-265

[7] Chen L, Hu L, Wu J, et al. WTcluster: Utilizing tags for web service clustering//Kappel G, Maamar Z, Motahari-Nezhad H R eds. Service-Oriented Computing. Berlin Heidelberg: Springer, 2011: 204-218

[8] Chen L, Wang Y, Yu Q, et al. WT-LDA: User tagging augmented LDA for web service clustering//Basu S, Pautasso C, Zhang L, Fu X eds. Service-Oriented Computing. Berlin Heidelberg: Springer, 2013: 162-176

[9] Wang H, Shi X, Yeung D Y. Relational stacked denoising autoencoder for tag recommendation//Proceedings of the 29th AAAI Conference on Artificial Intelligence. Austin, USA, 2015: 3052-3058

[10] Wang H, Chen B, Li W. Collaborative topic regression with social regularization for tag recommendation//Proceedings of the 23rd International Conference on Artificial Intelligence. Beijin, China, 2013: 2719-2725

[11] Azmeh Z, Falleri J R, Huchard M, Tibermacine C. Automatic web service tagging using machine learning and WordNet synsets//Proceedings of the 6th Web Information Systems and Technologies. Valencia, Spain, 2010: 46-59

[12] Fernandez A, Hayes C, Loutas N, et al. Closing the service discovery gap by collaborative tagging and clustering techniques//Proceedings of the 7th International Semantic Web Conference. Karlsruhe, Germany, 2008: 115-128

[13] Li W J, Yeung D Y, Zhang Z. Probabilistic relational PCA// Proceedings of the 23rd Advances in Neural Information Processing Systems. Hyatt Regency, Vancouver, Canada, 2009: 1123-1131

[14] Li W J, Zhang Z, Yeung D Y. Latent wishart processes for relational kernel learning//Proceedings of the 12th International Conference on Artificial Intelligence and Statistics. Clearwater Beach, Florida, USA, 2009: 336-343

[15] Li C, Zhang R, Huai J P, Sun H L. A novel approach for api recommendation in mashup development//Proceedings of the International Conference on Web Services. Anchorage, USA, 2014: 251-258

[16] Wang M, Ni B, Hua X-S, Chua T-S. Assistive tagging: A survey of multimedia tagging with human-computer joint exploration. ACM Computing Surveys, 2012, 44(4): 25

[17] Liu Z, Chen X, Sun M. A simple word trigger method for social tag suggestion//Proceedings of the Association for Computational Linguistics. Stroudsburg, USA, 2011: 1577-1588

[18] Belém F M, Martins E F, Almeida J M, Gonçalves M A. Personalized and object-centered tag recommendation methods for Web 2.0 applications. Information Processing & Management, 2014, 50(4): 524-553

[19] Pujari M, Kanawati R. Tag recommendation by link prediction based on supervised machine learning//Proceedings of the 6th International AAAI Conference on Weblogs and Social Media. Dublin, Ireland, 2012: 547-550

[20] Menezes G V, Almeida J M, Belém F et al. Demand-driven tag recommendation//Balcázar J L, Bonchi F, Gionis A, Sebag M eds. Machine Learning and Knowledge Discovery in Databases. Berlin Heidelberg: Springer, 2010: 402-417

[21] Zhao W, Guan Z, Liu Z. Ranking on heterogeneous manifolds for tag recommendation in social tagging services. Neurocomputing, 2015, 148: 521-534

[22] Bao Y, Fang H, Zhang J. TopicMF: Simultaneously exploiting ratings and reviews for recommendation//Proceedings of the 28th AAAI Conference on Artificial Intelligence. Québec City, Canada, 2014: 2-8

[23] Brin S, Page L. Reprint of: the anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems, 1998, 30(1): 107-117

[24] Shao J, Li B, Liu H. PageRank improved algorithm-adjustment damping factor. Mathematica Applicata, 2008: S1
[25] Xing W, Ghorbani A. Weighted pagerank algorithm// Proceedings of the Communication Networks and Services Research. N. B., Canada, 2004: 305-314
[26] Tyagi N, Sharma S. Comparative study of various page ranking algorithms in Web Structure Mining. International Journal of Innovative Technology and Exploring Engineering, 2012, 1(1): 2278-3075
[27] Bleid D M, Ng A Y, Jordan M I. Latent dirichlet allocation. Journal of Machine Learning Research, 2003, 3(1): 993-1022
[28] Heinrich G. Parameter estimation for text analysis. vsonix. GmbH and University of Leipzig, Germany: Technical Report CS679, 2004
[29] Chang Z, Blei D M. Relational topic models for document networks//Proceeding of the 12th International Conference on Artificial Intelligence and Statistics. Florida, USA, 2009: 81-88
[30] Wang J, Hong L J, Davison B D. Tag recommendation using keywords and association rules//Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Bled, Slovenia, 2009: 261-274
[31] Font F, Serrà J, Serra X. Analysis of the impact of a tag recommendation system in a real-world folksonomy. ACM Transactions on Intelligent Systems and Technology, 2015, 7(1): 6

LIU Jian-Xun, born in 1970, Ph. D., professor, Ph. D. supervisor. His research interests include service computing and cloud computing, theory and application of workflow management, etc.

SHI Min, born in 1991, M. S. candidate. His research interests include information retrieval and service computing.

ZHOU Dong, born in 1979, associate professor, M. S. supervisor. His research interests include information retrieval, natural language processing and machine learning, etc.

TANG Ming-Dong, born in 1978, professor, M. S. supervisor. His research interests include service computing and cloud computing.

ZHANG Ting-Ting, born in 1991, M. S. candidate. Her research interests include service computing and cloud computing.

Background

This paper studies the problem of automatic tag recommendation for Mashups. Tags have been extensively utilized to annotate Mashups, which can bring benefits from several aspects, such as facilitating the better management, classification and retrieval of Mashups in large repository. In the past, people favor manual tags creation when registering a new Mashup. However, this approach demands user intervention, which is extremely time-consuming and probes to errors. Therefore, automatic tag recommendation systems are required urgently to release people from such time-consuming and tedious task, as well as regulate the vocabulary of tags people could use.

In the past, majority of tag recommendation approaches have been proposed, including co-occurrence based methods, clustering based methods and topic model based methods, etc. In some situations, it is crucial to leverage the latent topics of items for the recommendation accuracy improvement especially when the tag-item matrix is extremely sparse. Although topic model based methods such as LDA have been wildly utilized to address the above problem, they still suffer from lacking of rich textual content. Therefore, auxiliary information such as social networks (composition relationships between Web services) should be fully used to further promote the recommendation accuracy.

To address the above issue, we propose a method for Mashup tag recommendation based on a topic model. The model simultaneously takes the description documents for Mashups and Web APIs as well as the composition relationships between them into account. Based on the model, our approach first selects the most similar APIs of a new Mashup. Subsequently, those chosen similar APIs and member APIs of this Mashup are combined into a single APIs set. Then we select several most important APIs from this APIs set based on a weighted PageRank algorithm. Finally, tags of the important APIs are recommended to this Mashup. Moreover, we design a ranking algorithm for tags to be recommended, which assigns higher recommendation score to those tags whose topics are most similar to Mashup.

The group's research interests are currently focused on the theories and methods of service computing, service management, the discovery and composition of services, etc. we have devoted ourselves to the new methods of service recommendation last several years. The long term goal is to improve the user experience and performance of service computing. The research was supported by the National Natural Science Foundation of China under Grant Nos. 61572187, 61300129, 61272063, 61572186, the project Sponsored by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, China under Grant No. [2013] 1792, the project supported by Scientific Research Fund of Hunan Provincial Education Department under Grant No. 16K030, the project supported by the Hunan Provincial Innovation Foundation for Postgraduate under Grant No. CX2016B573.