软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2023,34(6):2727−2748 [doi: 10.13328/j.cnki.jos.006508]

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

基于 TWE-NMF 主题模型的 Mashup 服务聚类方法*

陆佳炜[1], 赵 伟[2], 张元鸣[2], 梁倩卉[3], 肖 刚[1]

[1](中国计量大学 机械电子工程学院, 浙江 杭州 310018)

[2](浙江工业大学 计算机科学与技术学院, 浙江 杭州 310023)

[3](School of Computer Science and Engineering, Nanyang Technological University, Singapore 637457, Singapore)

通信作者: 肖刚, E-mail: xg@zjut.edu.cn

摘 要: 随着互联网和面向服务技术的发展, 一种新型的 Web 应用——Mashup 服务, 开始在互联网上流行并快速增长. 如何在众多 Mashup 服务中找到高质量的服务, 已经成为一个大家关注的热点问题. 寻找功能相似的服务并进行聚类, 能有效提升服务发现的精度与效率. 目前国内外主流方法为挖掘 Mashup 服务中隐含的功能信息, 进一步采用特定聚类算法如 K-means 等进行聚类. 然而 Mashup 服务文档通常为短文本, 基于传统的挖掘算法如 LDA 无法有效处理短文本, 导致聚类效果并不理想. 针对这一问题, 提出一种基于非负矩阵分解的 TWE-NMF (non-negative matrix factorization combining tags and word embedding) 模型对 Mashup 服务进行主题建模. 所提方法首先对 Mashup 服务规范化处理, 其次采用一种基于改进的 Gibbs 采样的狄利克雷过程混合模型, 自动估算主题的数量, 随后将词嵌入和服务标签等信息与非负矩阵分解相结合, 求解 Mashup 服务主题特征, 并通过谱聚类算法将服务聚类. 最后, 对所提方法的性能进行了综合评价, 实验结果表明, 与现有的服务聚类方法相比, 所提方法在准确率、召回率、*F*-measure、纯度和熵等评价指标方面都有显著提高.

关键词: Mashup 服务; 非负矩阵分解; 主题模型; 词嵌入; 服务聚类

中图法分类号: TP311

中文引用格式: 陆佳炜, 赵伟, 张元鸣, 梁倩卉, 肖刚. 基于TWE-NMF主题模型的Mashup服务聚类方法. 软件学报, 2023, 34(6): 2727–2748. http://www.jos.org.cn/1000-9825/6508.htm

英文引用格式: Lu JW, Zhao W, Zhang YM, Liang QH, Xiao G. TWE-NMF Topic Model-based Approach for Mashup Service Clustering. Ruan Jian Xue Bao/Journal of Software, 2023, 34(6): 2727–2748 (in Chinese). http://www.jos.org.cn/1000-9825/6508.htm

TWE-NMF Topic Model-based Approach for Mashup Service Clustering

LU Jia-Wei[1], ZHAO Wei[2], ZHANG Yuan-Ming[2], LIANG Qian-Hui[3], XIAO Gang[1]

[1](School of Mechanical and Electrical Engineering, China Jiliang University, Hangzhou 310018, China)

[2](College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)

[3](School of Computer Science and Engineering, Nanyang Technological University, Singapore 637457, Singapore)

Abstract: With the development of the Internet and service-oriented technology, a new type of Web application—Mashup service, began to become popular on the Internet and grow rapidly. How to find high-quality services among large number of Mashup services has become a focus of attention. It has been shown that finding and clustering services with similar functions can effectively improve the accuracy and efficiency of service discovery. At present, current methods mainly focus on mining the hidden functional information in the Mashup service, and use specific clustering algorithms such as K-means for clustering. However, Mashup service documents are usually short texts. Traditional mining algorithms such as LDA are difficult to represent short texts and find satisfied clustering effects from them. In order to solve this problem, this study proposes a non-negative matrix factorization combining tags and word embedding (TWE-NMF)

* 基金项目: 国家自然科学基金 (61976193); 浙江省自然科学基金 (LY19F020034); 浙江省重点研发计划 (2021C03136)
收稿时间: 2020-11-02; 修改时间: 2021-01-29, 2021-08-10; 采用时间: 2021-10-12; jos 在线出版时间: 2022-12-08
CNKI 网络首发时间: 2022-12-10

model to discover topics for the Mashup services. This method firstly normalizes the Mashup service, then uses a Dirichlet process multinomial mixture model based on improved Gibbs sampling to automatically estimate the number of topics. Next, it combines the word embedding and service tag information with non-negative matrix factorization to calculate Mashup topic features. Moreover, a spectral clustering algorithm is used to perform Mashup service clustering. Finally, the performance of the method is comprehensively evaluated. Compared with the existing service clustering method, the experimental results show that the proposed method has a significant improvement in the evaluation indicators such as precision, recall, *F*-measure, purity, and entropy.

Key words: Mashup service; non-negative matrix factorization (NMF); topic model; word embedding; service clustering

1 引 言

随着云计算的发展和服务计算“服务化”的思想驱动, 越来越多的公司将数据、资源或者相关业务通过 Web 服务的形式发布到互联网上, 以提高信息的利用率和自身竞争力. 然而传统基于 SOAP 协议的 Web 服务, 存在技术体系复杂、扩展性差等问题, 难以适应现实生活中复杂多变的应用场景[1]. 为克服传统服务带来的问题, 近年来, 互联网上涌现出一种轻量级的信息服务组合模式——Mashup 技术, 可以混搭多种不同 Web API, 开发出多种全新的 Web 应用, 以缓解传统服务难以适应复杂多变应用环境的问题[1−3].

然而, 随着互联网上 Mashup 服务数量以及服务功能种类不断增多, 越来越多的服务呈现功能属性差异难以界定的特性, 从数量庞大的服务集合中精确地定位满足用户特定业务需求的服务日益变得困难[4]. 以 ProgrammableWeb 网站为例, 到 2020 年 9 月为止, ProgrammableWeb 已经发布了近 8 000 条的 Mashup 服务以及 23 000 多条 Web API 服务.

不少研究表明, 通过聚类技术预先对 Mashup 服务进行聚类, 将功能相似的服务聚类到一起, 可以有效缓解上述问题[1,2,4−9]. 目前现有的方法, 主要采用潜在狄利克雷分配 (latent Dirichlet allocation, LDA)[10]或者其扩展模型[11−14], 对 Mashup 服务建立主题模型, 通过文档的主题特征向量来对服务进行聚类. 另外也有研究者通过 TF-IDF, Doc2Vec 等模型和工具对 Mashup 服务进行建模与聚类[15,16]. 虽然上述工作对 Mashup 服务聚类提出改进方案, 但仍存在以下缺陷.

(1) Mashup 服务描述文档通常比较简短、特征稀疏、信息量少, LDA 等模型在处理短文本上效果远远不如长文本, 导致目前大部分主题模型很难对缺乏训练语料的短文本进行很好地建模[17,18]. 另一方面, 由于短文本内单词基本都是出现一次, 缺少高频词信息, 对于 TF-IDF 等模型而言则很难计算出单词的语义权重[19,20].

(2) Mashup 服务描述文档与普通短文本相比仍存在一定的差异性, Mashup 服务有标签信息以及 API 组成信息, 标签信息以及 API 对应的描述信息可以作为先验信息进行辅助聚类. 但大多数研究者仅考虑服务描述文档本身蕴含的信息[1,6,7], 未考虑到描述文档与服务标签之间的联系, 从而在模型生成过程中丢失了部分潜在的语义信息.

(3) LDA 等主题模型通常需要指定主题个数[10−13], 然而服务的主题个数很难直接确定.

(4) 目前多数服务聚类算法都是将 K-means 作为最后主题特征值的聚类算法[5,9,16], 但是 K-means 算法由于受聚类中心点随机性以及无法发现非凸形状簇的影响, 可能导致聚类质量不理想.

为了能克服上述服务聚类中的缺陷, 我们提出一种基于 TWE-NMF (non-negative matrix factorization combining tags and word embedding) 主题模型的 Mashup 服务聚类方法. 该方法利用非负矩阵分解 (non-negative matrix factorization, NMF) 对 Mashup 服务进行主题挖掘, 引入改进的 Gibbs 采样方法来自动确定主题数量, 融合优化的词嵌入和单词语义权重计算方法来缓解短文本带来的稀疏性问题, 最后采用谱聚类算法对 Mashup 服务的主题特征进行聚类, 从而找到更优的解集. 概括地说, 本文的主要贡献如下.

(1) 我们将狄利克雷过程混合模型 (Dirichlet process multinomial mixture, DPMM) 与 NMF 求解主题特征相结合, 通过 DPMM 模型自动估算主题的个数, 从而克服传统 NMF 模型需要人工预先指定主题数的缺陷.

(2) 我们将服务标签和词嵌入信息引入到 NMF 模型中, 提出了一种 TWE-NMF 主题模型来处理 Mashup 服务

主题特征, 通过在 NMF 中分解 SPPMI (shifted positive pointwise mutual information) 矩阵的方法求解词嵌入信息, 并将服务标签和文本上下文信息相结合进一步精确计算单词的语义权重, 以有效缓解传统主题模型在短文本上表现效果不佳的问题, 并提高了 Mashup 服务主题建模的准确性.

(3) 我们将得到的主题特征通过谱聚类算法进行聚类, 并使用真实的 Mashup 服务数据集对我们提出的方法进行评估, 实验结果验证了所提方法的有效性, 与现有的服务聚类算法相比该方法能进一步改善聚类的效果.

本文第 2 节介绍了目前相关的研究工作. 第 3 节详细阐述了所提主题模型和聚类方法. 第 4 节以 ProgrammableWeb 上爬取的数据为例进行实验分析与评估. 第 5 节对全文进行总结与展望.

2 相关工作

服务聚类技术在服务发现中发挥着重要作用, 通过服务聚类, 将功能上类似的服务分组到相同簇中, 能有效提高服务发现的效率和准确性. 考虑到服务描述文档是 Mashup 服务聚类的主要信息来源, 目前研究都是将 Mashup 服务描述文档作为切入点, 通过对服务描述文档的分析和处理, 找出服务特征信息后实现服务聚类.

一些研究人员通过向量空间模型[15,16,21], 将 Mashup 服务描述文档表示成向量, 进而对 Mashup 服务进行聚类. 文献 [15] 利用基于词频信息的 TF-IDF 方法来提取 Mashup 服务描述文档中的特征信息, 将每个 Mashup 服务描述文档转化为 TF-IDF 向量, 再通过 K-means 算法对这些 TF-IDF 向量进行聚类. 但是, 如果使用 TF-IDF 权重向量信息进行服务功能的表征, 仅仅利用了文本表层的词频信息, 未考虑到文本潜在的语义联系.

文献 [16] 采用 Doc2vec 文本工具直接对 Mashup 服务描述文档进行建模, 利用神经网络模型进行语料的训练, 为每个单词形成一个多维向量, 将单词之间的关系映射到文档上, 并进一步将文档投影到向量空间, 随后使用聚类算法对得到的 Mashup 文档向量进行聚类, 相对于 TF-IDF 模型, 使用 Doc2Vec 能更好地挖掘 Mashup 服务的潜在语义信息. 然而使用 Doc2vec 直接训练, 需要依赖完善可靠的语料库, 才能取得较好的训练结果.

由于向量空间模型依据的是词频信息, 取决于共有词汇的数量, 会存在向量维度过高、语义稀疏等问题[1]. 为缓解上述问题, 大量研究人员开始利用主题模型处理服务聚类. 主题模型是如今最流行的语言模型之一, 它在文档和词之间引入主题维度, 将文档-词映射变为文档-主题和主题-词的映射, 通过主题层挖掘文档隐藏信息, 并将高维向量映射到低维, 以降低计算复杂度. 如文献 [2,4,6−9,22] 通过 LDA 或其扩展模型方法 (BTM, GPU-DMM, G-LDA 等) 来提取服务潜在语义信息, 再进一步对 Mashup 服务进行聚类.

虽然通过主题模型在一定程度上提高了服务聚类精度, 但是 Mashup 服务描述文档通常是短文本, LDA 等传统主题模型对短文本处理效果远不如长文本. 为解决短文本数据稀疏性问题, 文献 [1] 利用 Word2Vec 工具, 对 Wiki 语料库进行预训练, 根据词向量计算相似度高的单词, 对 Mashup 服务描述进行扩充. 然而, 利用外部语料库扩充描述可能存在语料库更新不及时, 合适性差等问题, 引入的信息容易存在噪声数据, 对最后结果产生负影响. 文献 [11] 提出了 BTM (Biterm topic model) 模型, BTM 通过在短文本集合中构建词对的方式, 缓解了稀疏性问题, 相对于 LDA 对短文本有更好的处理能力. 文献 [12] 给出了一种 GPU-DMM (general Pólya Urn Dirichlet multinomial mixture) 主题模型, 利用从大语料学习到的词嵌入信息和 GPU 过程提高短文本的主题模型效果. 文献 [17,23] 假设每个短文档只是一个较长的伪文档片段, 主题与伪文档相关联, 通过生成较长的伪文档缓解了稀疏性问题. 但是伪文档可能由许多主题无关的短文本组成, 这可能使得主题推断的效果较差[18].

NMF 模型对稀疏数据有较强的处理能力, 不少研究者采用 NMF 进行主题发现[24−27]. 文献 [27] 实验表明, NMF 在处理短文本结果优于 LDA 模型, 同时 LDA 等概率模型结果受先验参数的影响, 需要花费大量时间去调节先验参数至合适值, 而 NMF 相对更为简单, 不需要事先指定先验参数. 本文采用 NMF 对 Mashup 服务进行主题挖掘, 然而传统的 NMF 依旧存在以下问题: (1) 与 LDA 等传统主题模型一样, 主题个数难以自动确认, 需要反复地对主题数进行调整以寻找其最佳值. (2) 仅使用文档-词频信息, 忽略了单词之间的语义信息. (3) Mashup 服务描述通常为短文本, 单词在文档中往往出现一次, 使用文档-词频信息后, 导致单词语义权重区分度不高.

针对问题 (1), 部分研究者通过贝叶斯非参数模型[7,14,28−30]对主题进行自动分析, 例如文献 [7] 采用 HDP

(hierarchical Dirichlet process) 的方法对 Mashup 进行主题建模, 但是 HDP 作为 LDA 的扩展模型, 并不是为短文本设计的, 同时仅使用 HDP 主题模型也会忽略单词之间的语义关系. 文献 [28] 采用一种 GSDPMM (collapsed Gibbs sampling algorithm for the Dirichlet process multinomial mixture model) 的方法来估计主题个数, 其核心是通过改进的 Gibbs 采样方法求解 DPMM 模型, 相对于 HDP 模型能更好地处理短文本. 我们在 NMF 模型中引入此方法的求解方式, 以缓解 NMF 中难以确定主题数的问题, 从而更好地挖掘 Mashup 服务的主题信息.

针对问题 (2), 许多研究[6,22,31]通过主题模型和词嵌入信息进行主题发现. 词嵌入是自然语言处理中一组语言建模和特性学习技术的总称, 它可以同时将词汇的句法和语义信息学习为连续向量, 捕获单词语义关系. 但是在 LDA 等常见主题模型中, 词嵌入一般都是神经网络模型, 在模型上很难进行有机的统一[26]. 目前多数研究工作都是通过预先训练外部语料库的方式, 将词嵌入与主题模型进行结合. 文献 [6] 通过 Word2Vec 工具对 Wiki 百科语料库进行预训练, 得到词向量, 计算单词之间的相似度关系, 之后将 LDA 与词嵌入信息结合, 提出一种 WE-LDA 主题模型用于 Mashup 服务的主题发现, 文献 [22] 采用 Gaussian-LDA 模型用于服务主题建模, 文献 [31] 采用 GPU-DMM 的方法对 Mashup 服务进行主题挖掘. 然而上述方法多数是通过神经网络对外部语料库进行预训练, 很难保证外部语料库在 Mashup 服务上的合适性. 经过对大量研究工作进行对比, 我们发现 Levy 等人[32]在 NIPS 的论文中已经证明基于负采样的 Skip-gram 模型求解词嵌入信息, 相当于隐式分解单词的 SPPMI 矩阵. 相对于神经网络模型求解词嵌入信息, 采用分解 SPPMI 矩阵求解词嵌入信息可以和 NMF 在模型上进行有机的统一, 同时不需要预训练外部语料库. 因此, 在 NMF 求解 Mashup 服务主题的过程中, 我们利用分解 SPPMI 矩阵的方式, 进而引入词嵌入信息缓解短文本稀疏性问题.

传统的主题模型仅使用服务描述作为建模, 忽略了不少有用的先验信息. 不少研究发现[4,8,9], 标签信息作为先验信息能有效缓解数据的稀疏性问题, 并对主题发现和数据聚类及推荐有良好的促进作用. 文献 [8] 使用 Mashup 服务中的 API 构成和 API 标签信息, 并进一步结合 LDA 对 Mashup 服务进行主题建模. 文献 [9] 中提出了一种基于 LDA 的双层主题模型, 首先基于服务标签信息和 API 构成来组建 Mashup 网络层, 计算出 Mashup 之间的关系, 随后将 Mashup 之间的关系作为先验信息, 用于 LDA 以生成主题特征, 进而提高建模的准确性. 基于上述论文中借鉴标签信息的思想, 针对问题 (3), 本文将服务标签和文本上下文信息相结合提出了一种改进的 TF-IDF 单词权重方法, 用于进一步精确计算单词的语义权重, 以代替原始 NMF 模型中的文档-词频信息.

此外, 在面向主题的服务聚类过程中, 通常是将服务聚类到包含其概率最大的主题类中, 或者进一步采用聚类算法, 计算服务之间的相似度并进行聚类. 考虑到 Mashup 服务可能受到多个主题的共同影响, 直接将服务聚类到其主题概率最大的类中并不合适, 因此采用特定聚类算法计算服务主题特征的相似度更为合适[1]. 但是 K-means、FCM 等经典的聚类算法只能发现球状簇, 无法发现非凸形状的簇, 容易陷入局部最优, 导致聚类结果下降[33]. 谱聚类是一种基于谱图理论的无监督聚类算法, 其本质是利用谱松弛方法将聚类问题转换为图的最优划分问题[34], 与 K-means 等传统聚类算法相比, 能发现任意形状的簇, 不容易陷入局部最优. 在面向主题的服务聚类过程中, 我们利用谱聚类算法对 Mashup 主题特征进行聚类, 从而进一步提高服务的聚类精度.

3 方 法

本文所提方法的整体框架如图 1 所示, 此框架包含以下 3 部分内容.

(1) 将爬取到的真实 Mashup 服务文档进行数据预处理, 通过 DPMM 模型自动计算主题数量 K.

(2) 提取服务描述中的名词和服务标签信息, 通过 TCSW (tags and context semantic weight) 方法, 计算单词的语义权重信息, 代替 NMF 主题模型中文档-词频矩阵. 将 Mashup 服务描述文档长度作为滑动窗口, 得到单词共现信息, 进一步计算单词的上下文 SPPMI 矩阵信息. 接下来, 将计算得到的单词语义权重和 SPPMI 矩阵引入 NMF 中, 提出 TWE-NMF 模型求解 Mashup 主题信息.

(3) 将 Mashup 服务的主题信息作为特征值, 通过谱聚类方法进行聚类评估.

为了方便阅读, 本节将文中涉及的一些常用符号及其含义归纳在表 1 中.

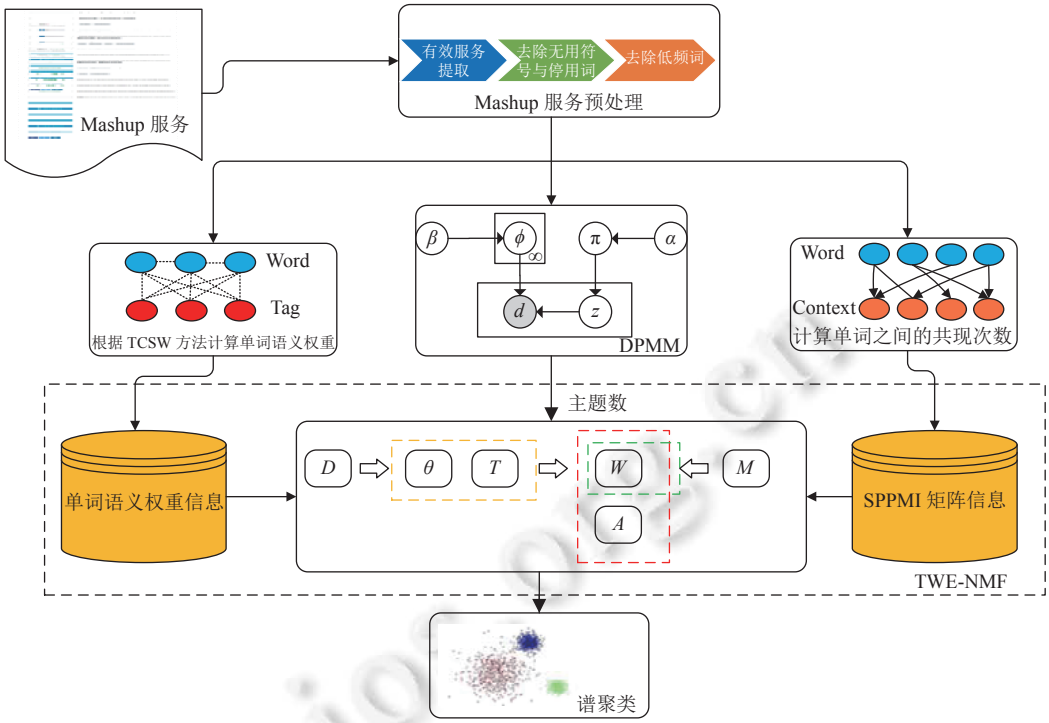


图 1 方法总体框架

表 1 常用符号表

符号	含义
V	语料库中单词的数量
N	语料库中文档数量
K	主题数
E	嵌入空间维度
$D \in R^{N\times V}$	文档-单词关系矩阵
$T \in R^{V\times K}$	单词-主题矩阵
$\theta \in R^{D\times K}$	文档-主题矩阵
$M \in R^{V\times V}$	单词上下文信息矩阵
$W \in R^{V\times E}$	词嵌入矩阵
$S \in R^{E\times E}$	缩放因子
$A \in R^{K\times E}$	主题嵌入矩阵
N_d	文档d中单词的数量
m_z	主题z中文档的数量
N_d^w	文档d中单词w出现的次数
n_z^w	主题z中单词w出现的次数

3.1 Mashup 主题数确认

主题建模是一种文本信息特征抽取技术, 它能够将文本中潜在的隐含主题信息挖掘出来. 然而传统的 LDA, NMF 等模型都需要预先指定主题的个数, 无法通过模型自动确定主题的数量, 需要反复进行实验才能确定主题的最佳值. 本文通过 DPMM 模型, 一种贝叶斯非参数模型, 来解决传统主题模型中主题数 K 需要人工预先指定的问

题. 在 DPMM 模型中, 主题的数量 K 不需要作为输入参数, 可以从数据中推断主题的数量, 并允许主题的数量随着观察到的新数据的增加而增长. 但由于传统的方法求解 DPMM 收敛速度较慢, 本文采用 GSDPMM, 一种经过改进的 Gibbs 采样方法[28]求解该模型, 来对 Mashup 服务数据集的主题数进行估计. 首先, DPMM 模型如图 2 所示.

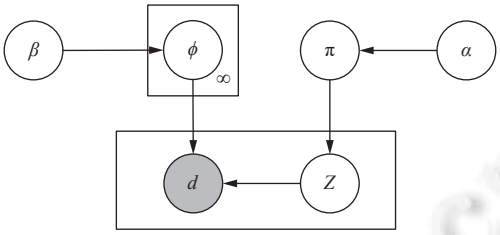


图 2 DPMM 模型

图 2 中, α 、 β 为超参数, π 为主题的先验概率分布, ϕ 为单词在主题上的先验概率分布, d 为当前文档, 阴影圆圈表示可观察到的变量, z 表示主题. DPMM 生成过程如以下公式所示:

$$\pi|\alpha \sim GEN(\alpha) \tag{1}$$

$$z_d|\pi \sim Mult(\pi),\ d=1,2,\ldots,N \tag{2}$$

$$\phi_k|\beta \sim Dir(\beta),\ k=1,2,\ldots,\infty \tag{3}$$

$$d|z_d,\{\phi_K\}_{k=1}^{K} \sim p(d|\phi_{z_d}) \tag{4}$$

相对于传统 DPMM 模型求解方式, 我们通过 Gibbs 采样来求解 DPMM 模型, 可以得到如下求解公式:

$$p(z_d=z|Z,N,\alpha,\beta) \propto \frac{m_{z,\neg d}}{N-1+\alpha} \frac{\prod_{w\in d}\prod_{j=1}^{N_d^w}(n_{z,\neg d}^w+\beta+j-1)}{\prod_{i=1}^{N_d}(n_{z,\neg d}+V\times\beta+i-1)} \tag{5}$$

$$p(z_d=K+1|Z,N,\alpha,\beta) \propto \frac{\alpha}{N-1+\alpha} \frac{\prod_{w\in d}\prod_{j=1}^{N_d^w}(\beta+j-1)}{\prod_{i=1}^{N_d}(V\times\beta+i-1)} \tag{6}$$

公式 (5) 表示文档 d 选择当前已存在的主题 z 的概率, 公式 (6) 表示文档 d 选择新主题的概率, N_d^w 表示单词 w 在文档 d 中出现的次数, $\neg d$ 表示不统计文档 d 信息的统计结果, $m_{z,\neg d}$ 表示除去当前文档 d 信息后每个主题中文档的数量, $n_{z,\neg d}^w$ 表示未统计文档 d 信息下单词 w 在主题 z 中的数量, $n_{z,\neg d}$ 表示未统计文档 d 信息下主题 z 中单词的数量. 通过公式 (5) 与公式 (6) 的计算结果使得 DPMM 可以在迭代中动态调整主题数到合适值, 并且拥有更快的收敛性, 使模型在短文本处理上能更好地挖掘主题数量. 从算法时间复杂度的角度来看, DPMM 的求解主题值主要在每次迭代中, 使用公式 (5) 计算文档 d 选择现有 K 个主题的概率和公式 (6) 计算文档 d 选择新主题的概率, 由于公式 (5) 和公式 (6) 都和文档长度呈线性关系, 因此执行一次 DPMM 过程的时间复杂度为 O(KNL), L 表示文档长度.

3.2 词嵌入信息引入

词嵌入模型寻求单词的连续表示, 即具有共同意义的词在潜在空间中拥有大致相似的嵌入形式, 已被证明是捕获单词语义关系的有效工具[24]. Word2Vec 为目前使用最广的词嵌入工具之一, 主要基于 CBOW 和 Skip-gram 两种模型进行训练[21]. 词嵌入将语义属性相似的单词在连续向量空间中投影到同一区域, 可以提高主题模型的聚类性能[26]. 本文利用分解 SPPMI 矩阵[32]的方法求解词嵌入信息, 以便和 NMF 在模型上能进行更好的统一.

SPPMI 矩阵可以通过点互信息 (pointwise mutual information, PMI) 进行计算, PMI 被广泛用于计算单词间相似度的关系. 当两个单词在文本中共现概率越大时, 单词间的相关性就越强, PMI 计算公式如下所示:

$$PMI(x,y)=\log\frac{P(x,y)}{P(x)P(y)} \tag{7}$$

根据单词 w_j 和其上下文单词 w_c 在语料库中的实际共现次数, 可以估算出两者之间的 PMI 值:

$$PMI(w_j,w_c)=\log\frac{\#(w_j,w_c)\cdot E}{\#(w_j)\cdot\#(w_c)} \tag{8}$$

其中, $\#(w_j,w_c)$ 表示单词 w_j 和上下文单词 w_c 在语料库中的实际共现次数, E 为单词和上下文单词对共现的总次数. $\#\left(w_j\right)=\sum_{w_c\in V}\#(w_j,w_c)$, $\#(w_c)=\sum_{w_j\in V}\#(w_j,w_c)$, 进一步可得出 SPPMI 矩阵的计算方式为:

$$SPPMI(w_j,w_c)=\max(PMI(w_j,w_c)-\log\kappa,\ 0) \tag{9}$$

其中, κ 为负采样系数. 经过预处理后的 Mashup 服务描述文档较短, 我们将整个服务描述文档作为滑动窗口的长度, 求解单词共现情况. 通过分解公式 (9) 中 SPPMI 矩阵, 可以将词嵌入信息引入到 NMF 求解主题信息中, 从而缓解短文本稀疏性问题. 从算法时间复杂度的角度来看, SPPMI 信息的求解主要为词频信息的统计, 时间复杂度为 $O(NL^2)$.

3.3 单词语义权重计算

传统的 NMF 主题模型采用文档-词频信息或者 TF-IDF 权重值作为文档-单词矩阵信息求解主题特征, 但是 Mashup 服务描述文档通常较短, 关键单词在描述中很难再次出现. 即多数关键性单词的 TF 值等于 1, 对文档的区分作用非常小, 因此传统的词频信息和 TF-IDF 模型无法很好地对 Mashup 服务描述文档进行建模. 我们在对大量的 Mashup 服务描述文档分析后发现, 服务标签在一定程度上总结了服务的功能特点, 单词与服务标签相似度越高, 说明该单词在描述文档中的语义比重相较于其他单词要高. 若服务描述文档中的单词与上下文之间关系紧密, 更有可能是反映功能特征的单词. 此外, 服务描述中的名词信息, 相对于形容词、副词等其他形式的单词, 对主题的影响更加突出, 因此可以相对调整名词性单词的语义权重, 以更好挖掘主题信息. 本文在传统 TF-IDF 模型的基础上, 针对 TF-IDF 对短文本稀疏性问题处理上的不足和无法有效地获取潜在的语义信息等问题, 提出一种结合服务标签和文本上下文语义信息计算单词语义权重的方法 (tags and context semantic weight, TCSW), 进一步提高了名词性单词的语义权重值. 具体方法如下.

(1) 使用 Python NLTK 对服务描述内容进行词性标注, 将名词进行词形还原, 去重后放入名词集合 *Nset*, 然后提取服务标签信息, 保存到文档对应的标签集合 *Tag* 中.

(2) 对于文档中的每一个单词 w_x, 通过公式 (10) 计算 w_x 和其上下文相关单词的平均相似度, 得到 w_x 文本上下文语义权重信息 $WeightContext(w_x)$, 通过公式 (11) 计算 w_x 和服务标签的最大相似度, 得到 w_x 的服务标签语义权重信息 $WeightTag(w_x)$. $sim(w_x,w_y)$ 为单词 w_x 和 w_y 的相似度关系, 通过 WordNet 工具计算.

(3) 对于文档中的单词 w_x, 如果该单词在名词集合 *Nset* 中, 则通过公式 (12) 重新计算其语义权重信息, 否则文档中该单词的权重值为其 TF-IDF 值. 公式 (12) 主要通过放大单词的 TF-IDF 权重值实现, 若名词单词与上下文以及标签联系越紧密, 则分母越小, 得到的语义权重值也越大. 此处综合考虑了服务描述文档中单词之间的关联性以及单词和标签的关系, ω 设定为 0.5.

$$WeightContext(w_x)=\sum_{w_x,w_y\in d}\frac{sim(w_x,w_y)}{N_d-1},\ (x\neq y) \tag{10}$$

$$WeightTag(w_x,Tag_d)=\max_{t\in Tag_d}\{sim(w_x,t)\} \tag{11}$$

$$SemWeight(w_x)=\frac{TF\text{-}IDF(w_x)}{1-(WeightContext(w_x)\times\omega+WeightTag(w_x)\times(1-\omega))},\ (0<\omega<1) \tag{12}$$

算法 1. TCWS 语义权重计算.

输入: 文档集 *Docs*, 名词集 *Nset*, 文档对应的服务标签集 *Tag*;
输出: 文档-单词权重矩阵 *D*.

1. For *d* in *Docs*:
2. For w_x in *d*:
3. Calculate *WeightContext*(w_x) according to Eq. (10);

4. Calculate $WeightTag(w_x, Tag_d)$ according to Eq. (11);
5. For w_x in d:
6. If w_x in $Nset$:
7. Calculate $SemWeight(w_x)$ according to Eq. (12);
8. $D[d][w_x] = SemWeight(w_x)$;
9. Else:
10. $D[d][w_x] = TF\text{-}IDF(w_x)$;
11. Return D

从算法时间复杂度的角度来看, TCSW 方法主要由上下文语义权重信息和服务标签的最大相似度两个部分的计算组成, 时间复杂度为 $O(NL^2 + tNL)$, t 为服务标签数量.

3.4 TWE-NMF 主题建模

基于第 3.1–3.3 节, 可以得到服务文档的主题个数, 文档的单词语义权重信息, 单词的上下文 SPPMI 矩阵信息. 我们通过 NMF 将上述信息进一步融合, 提出了一种 TWE-NMF 模型求解主题信息. TWE-NMF 模型框架如图 3 所示, 主要由文档与主题间关系, 文本上下文信息与词嵌入间的关系及主题与词嵌入间的关系 3 个部分组成.

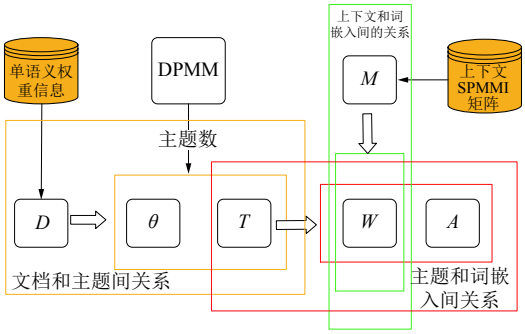


图 3 TWE-NMF 主题模型

3.4.1 文档-主题关系计算

给定全局文档-单词关系矩阵 D, 通过 NMF 将其分解为文档-主题矩阵 θ 和主题-单词矩阵 T 的乘积, NMF 的非负性保证了分解为文档-主题分布和主题-单词分布的可解释性. 分解 D 的函数表示为:

$$J = \| D - \theta T^{\top} \|_F^2 \text{ subject to : } \theta \geqslant 0 \text{ and } T \geqslant 0, \theta \in R^{N \times K}, T \in R^{V \times K} \tag{13}$$

相比于传统的 NMF 模型中的文档-词频关系矩阵, 在本文提出的模型中, 通过使用第 3.3 节中 TCSW 方法计算得到的单词语义权重信息来进行代替, 以有效挖掘潜在的语义信息. 同时使用第 3.1 节中 DPMM 模型自动估计合理主题数 K, 缓解 NMF 中主题个数 K 难以确认的问题.

3.4.2 上下文-词嵌入关系计算

由第 3.2 节中的公式 (9) 可以计算得到单词的上下文 SPPMI 矩阵 M, 分解 M 矩阵可以在 TWE-NMF 中引入词嵌入信息, 分解 M 的公式如下所示:

$$J = \| M - WSW^{\top} \|_F^2 \tag{14}$$

其中, S 是一个额外的对称因子, 用于 M 的近似求解, W 为单词的词嵌入矩阵. 相关研究表示[35], 上述形式的对称 NMF 求解, 比其基本形式即 S 作为单位矩阵的形式, 能提供更好的近似效果.

3.4.3 主题-词嵌入关系计算

利用 Mashup 服务文档和单词间的关系, 可以发现主题信息, 通过文档内单词上下文的共现信息, 可以学习到词嵌入信息. 但是这两个部分并不相互孤立, 语义相关的单词通常属于相似的主题, 在嵌入空间中也很接近. 因此

本文假设单词嵌入与它们的主题相关, 关系公式如下所示:

$$J = \| T - WA^{\top} \|_F^2 \tag{15}$$

我们在第 4 节对该假设进行了证明. 其中, 公式 (15) 将主题-单词矩阵 T 分解为主题嵌入矩阵 A 和词嵌入矩阵 W 的乘积, 将词嵌入与主题信息相联系起来, 进一步提高了主题建模的准确性.

3.4.4 目标函数求解

通过公式 (15) 可以将公式 (13) 和公式 (14) 进一步联系起来, 得到 TWE-NMF 主题模型目标函数:

$$J = \lambda_d \| D - \theta T^{\top} \|_F^2 + \lambda_w \| M - WSW^{\top} \|_F^2 + \lambda_t \| T - WA^{\top} \|_F^2 \text{ s.t. } \theta \geqslant 0 \text{ and } T \geqslant 0 \tag{16}$$

为了方便求解该目标函数, 参考文献 [36,37] 中解法, 将公式 (16) 重写为以下公式:

$$\begin{aligned} J(\theta, T, W, S, A) =& \lambda_d Tr(DD^{\top} - 2DT\theta^{\top} + \theta T^{\top} T\theta^{\top}) \\ &+ \lambda_w Tr(MM^{\top} - 2MWSW^{\top} + WSW^{\top} WSW^{\top}) + \lambda_t Tr(TT^{\top} - 2TAW^{\top} + WA^{\top} AW^{\top}) \end{aligned} \tag{17}$$

其中, Tr 表示矩阵求迹, λ_d, λ_w 和 λ_t 为不同部分的权重系数, 用于调整各部分计算的误差对结果的影响. 根据正则化约束得到以下目标函数:

$$L = J(\theta, T, W, S, A) + Tr(\alpha\theta^{\top}) + Tr(\beta T^{\top}) + Tr(\gamma W^{\top}) + Tr(\varphi S^{\top}) + Tr(\mu A^{\top}) \tag{18}$$

其中, α, β, γ, φ, μ 为正则化参数, 为使目标函数最小化, 对公式 (18) 求偏导得到以下公式:

$$\frac{\partial L}{\partial \theta} = -\lambda_d DT + \lambda_d \theta T^{\top} T + \alpha \tag{19}$$

$$\frac{\partial L}{\partial T} = -\lambda_d D^{\top}\theta + \lambda_d T\theta^{\top}\theta - \lambda_t WA^{\top} + \lambda_t T + \beta \tag{20}$$

$$\frac{\partial L}{\partial W} = -2\lambda_w MWS + 2\lambda_w WSW^{\top} WS - \lambda_t TA + \lambda_t WA^{\top} A + \gamma \tag{21}$$

$$\frac{\partial L}{\partial S} = -\lambda_w W^{\top} MW + \lambda_w W^{\top} WSW^{\top} W + \varphi \tag{22}$$

$$\frac{\partial L}{\partial A} = -\lambda_t T^{\top} W + \lambda_w AW^{\top} W + \mu \tag{23}$$

根据 Kuhn-Tucker 条件: $\alpha \odot \theta = 0$, $\beta \odot T = 0$, $\gamma \odot W = 0$, $\phi \odot S = 0$, $\mu \odot A = 0$, $\odot$ 表示阿达马乘积, 即矩阵对应位置的乘积. 利用阿达马乘积, 令上述公式偏导为 0, 进一步得到以下等式方程:

$$-(DT) \odot \theta + (\theta T^{\top} T) \odot \theta + \alpha \odot \theta = 0 \tag{24}$$

$$-(\lambda_d D^{\top}\theta + \lambda_t WA^{\top}) \odot T + (\lambda_d T\theta^{\top}\theta + \lambda_t T) \odot T + \beta \odot T = 0 \tag{25}$$

$$-2(\lambda_w MWS + \lambda_t TA) \odot W + (\lambda_t WA^{\top} A + 2\lambda_w WSW^{\top} WS) \odot W + \gamma \odot W = 0 \tag{26}$$

$$-(\lambda_w W^{\top} MW) \odot S + (\lambda_w W^{\top} WSW^{\top} W) \odot S + \varphi \odot S = 0 \tag{27}$$

$$-(T^{\top} W) \odot A + (AW^{\top} W) \odot A + \mu \odot A = 0 \tag{28}$$

根据乘法更新法则:

$$\theta \leftarrow \theta \odot \frac{DT}{\theta T^{\top} T} \tag{29}$$

$$T \leftarrow T \odot \frac{\lambda_d D^{\top}\theta + \lambda_t WA^{\top}}{\lambda_d T\theta^{\top}\theta + \lambda_t T} \tag{30}$$

$$W \leftarrow W \odot \frac{2\lambda_w MWS + \lambda_t TA}{\lambda_t WA^{\top} A + 2\lambda_w WSW^{\top} WS} \tag{31}$$

$$S \leftarrow S \odot \frac{W^{\top} MW}{W^{\top} WSW^{\top} W} \tag{32}$$

$$A \leftarrow A \odot \frac{T^{\top} W}{AW^{\top} W} \tag{33}$$

通过公式 (29)–公式 (33) 可求解 Mashup 服务文档-主题矩阵 θ 和主题-单词矩阵 T, 词嵌入矩阵 W, 主题嵌入矩阵 A. TWE-NMF 算法流程如算法 2.

算法 2. TWE-NMF.

输入: Mashup 服务语料库 *Docs*, 服务标签集 *Tag*, 迭代次数 *I*, 权重参数 λ_d, λ_w, λ_t;
输出: 文档-主题矩阵 θ, 词嵌入矩阵 *W*, 单词-主题矩阵 T, 主题嵌入矩阵 *A*.

1. Set *Num* = {}, *Nset* = {}, *Co* = {} // *Num* 为字典统计单词数量, *Nset* 为名词集统计名词, *Co* 为统计单词对的共现次数
2. For *d* in *Docs*:
3. For *w* in *d*:
4. if *w* is Noun: *Nset*.add(*w*) //如果是名词性单词加入名词集合
5. *Num*[*w*] ++ //计算单词出现的次数
6. For *d* in *Docs*:
7. For *w* in *d*:
8. if *Num*[*w*] < *threshold*: delete *w* //删除低频词
9. L = Len(*d*); // 计算删除单词后的文档长度
10. For *w* in *d*:
11. if(iscount(*w*)) continue; //已经统计过的单词跳过
12. For *i* in L:
13. if(w_i != *w*) *Co*[w_i][*w*]++, *Co*[*w*][w_i]++;
14. Calculate *M* according Eq. (8) and Eq. (9) //根据公式 (8) 和公式 (9) 计算 SPPMI 矩阵 *M*
15. Calculate *D* according to Algorithm1 //根据算法 1 计算文档-单词关系矩阵 *D*
16. Calculate *K* according to GSDPMM //根据 GSDPMM 方法求解主题个数
17. 初始化: θ, T, W, S, A
18. For *t* in (1, *I*):
19. Update θ according to Eq. (29); //根据公式 (29) 计算 θ
20. Update T according to Eq. (30); //根据公式 (30) 计算 T
21. Update W according to Eq. (31); //根据公式 (31) 计算 W
22. Update S according to Eq. (32); //根据公式 (32) 计算 S
23. Update A according to Eq. (33); //根据公式 (33) 计算 A
24. Return θ, T, W, A

从算法时间复杂度的角度来看, TWE-NMF 的时间复杂度由 4 部分组成, 主题数 *K*, SPMMI 矩阵 *M*, 文档-单词关系矩阵 *D* 以及最后的参数求解, 其中 SPMMI 矩阵 *M* 和文档-单词关系矩阵 *D* 两步骤可以同时进行, 因此前 3 部分的整体复杂度为 $O((K+t)NL+NL^2)$, 最后参数求解部分的复杂度由公式 (29)–公式 (33) 构成, 由于 *D* 和 *M* 是稀疏矩阵, 公式 (29) 的时间复杂度为 $O(NKV)$, 公式 (30) 的复杂度为 $O(NVK+VKE)$, 公式 (31) 的复杂度为 $O(V^2E+VKE+E^2V)$, 公式 (32) 的时间复杂度为 $O(V^2E+E^3+E^2V)$, 公式 (33) 的时间复杂度为 $O(VKE)$, 因此参数更新部分的时间复杂度为 $O(I(V^2E+VKE+E^2V+E^3+NKV))$, TWE-NMF 算法的整体时间复杂度为 $O(I(V^2E+VKE+E^2V+E^3+NKV)+(K+t)NL+NL^2)$.

3.5 基于服务主题特征的谱聚类

在 TWE-NMF 主题建模后, 对获得的 Mashup 服务的“服务文档-主题”特征向量聚类, 我们结合谱聚类算法进行面向服务主题的聚类, 将不同的 Mashup 服务划分到不同的类中. 整个聚类流程大致分为 3 步.

(1) 计算相似度矩阵 *SI*, 服务主题特征之间的相似度可以通过公式 (34) 高斯核函数计算. 公式中 θ_i 表示 Mashup 服务 *i* 的主题特征, δ 为尺度参数.

$$SI_{ij} = \exp\left(-\frac{\| \theta_i - \theta_j \|_2^2}{2\delta^2}\right) \tag{34}$$

(2) 由公式 (35) 将矩阵 SI 的每一列的元素相加, 并将每一列作为元素添加到度矩阵 G 对角线上, 随后通过 G 计算 Laplacian 矩阵 $L = G - S$, 并通过公式 (36) 对 L 进行标准化处理.

$$G_{ij} = \sum\nolimits_j SI_{ij} \tag{35}$$

$$L = G^{-\frac{1}{2}} L G^{-\frac{1}{2}} \tag{36}$$

(3) 根据公式 (37) 计算 L 的特征值, Tr 表示矩阵求迹, 将特征值从小到大排序, 取前 k 个特征值, 并计算前 k 个特征值的特征向量, 将 k 列向量合为一个矩阵, 得到服务文档特征向量矩阵 F, 并根据特征向量矩阵 F 将服务文档进行聚类划分.

$$\arg\min_F Tr\left(F^{\top} G^{-\frac{1}{2}} L G^{-\frac{1}{2}} F\right) \text{ s.t. } F^{\top} F = I \tag{37}$$

4 实验分析与评估

实验所采用的数据从 ProgrammableWeb 平台爬取, 其中 Mashup 服务数量为 6 217 条, Web API 数量为 11 930 条 (数据下载地址: https://github.com/viivan/Mashup-and-Web-Api-data1). 实验代码主要采用 Python 编写, 实验环境使用 64 位 Windows 10 操作系统, 内存容量为 16 GB.

在对 ProgrammableWeb 平台上爬取到的 Mashup 服务信息进行整理分析后, 发现一些 Mashup 服务语料信息较差, 有些服务描述文档存在内容过于简短, 无任何有效描述信息、服务重复注册, 分类标签信息不准确等问题. 优质的语料信息能够取得更好的结果, 为提高方法的准确性, 在方法的初始阶段, 需要对爬取的 Mashup 服务进行预处理操作. 具体的操作步骤如下.

(1) 针对每条 Mashup 服务信息, 提取出服务名称、服务描述、API 组合信息、类别信息以及服务标签信息进行整理, 去除重复注册的服务, 剔除关键信息为空的无效服务, 对缺少类别信息的服务进行人工标定.

(2) 对于每条服务的描述内容, 根据 NLTK 库中的停用词表去除停用词, 并删除数字, 符号等无用信息.

(3) 在实验中发现去除低频词可以有效地提高试验结果, 降低计算复杂度, 在预处理数据时, 本文为单词次数设定了阈值, 如果单词在全部文档中出现的次数小于阈值, 则认为该单词为干扰词, 统计时去除该单词.

由于 ProgrammableWeb 存在自带类别信息不准确的问题, 在预处理后的 Mashup 服务中, 我们选取服务数量较多的 12 个类别进行人工重新分类, 最终确定使用 1 346 条 Mashup 服务进行聚类实验 (数据下载地址: https://github.com/viivan/Mashup-data2). 实验数据集分布如表 2 所示.

表 2 实验数据分布

类别	数量	类别	数量	类别	数量
Search	253	Mapping	159	Real Estate	85
Travel	152	Video	106	eCommerce	82
Photos	103	Telephony	93	Games	71
Weather	88	Music	87	Messaging	67

4.1 聚类实验结果分析

为证明实验有效性, 本文采用以下基准方法进行对比试验, 如需进一步了解算法的效果, 读者可从 https://github.com/viivan/Mashup-clustering-algorithms-TWE-NMF 上下载这 8 种聚类方法的实现代码.

- T+Q: 通过 TF-IDF 将每个 Mashup 服务描述文档表示成向量形式, 进行 QT 聚类[38].
- LDA+K: 通过 LDA 主题模型对 Mashup 服务文档进行主题建模得到主题特征[9]. 在此基础上, 利用 K-means 算法对主题向量进行聚类.
- LDA+API+K: 通过 Word2Vec 对 API 描述文档进行预训练, 得到词向量. 对于 Mashup 服务描述文档中每个

单词, 在训练好的词向量模型中找出其前 3 个相似词[1]. 将这些相似词合并到 Mashup 服务描述文档中, 通过 LDA 主题模型建模, 最后采用 K-means 进行聚类.

● LDA+Wiki+K: 通过 Word2Vec 对 Wiki 语料库进行预训练, 得到词向量, 对于 Mashup 服务描述文档中每个单词, 在训练好的词向量模型中找出其前 3 个相似词[1]. 将扩充后的服务描述文档通过 LDA 建模后使用 K-means 算法聚类.

● BTM+K: 采用针对短文本改进的 BTM 主题模型对 Mashup 服务主题建模[11], 随后使用 K-means 聚类.

● GPU-DMM+K: 采用结合词嵌入信息的 GPU-DMM 主题模型[12]对 Mashup 进行 K-means 聚类.

● CLM+SC: 通过 CLM 主题模型[24]对 Mashup 服务主题建模, 采用谱聚类的方法对结果进行聚类.

● TWE-NMF+SC: 本文提出的方法, 综合结合词嵌入和服务标签等信息, 对 Mashup 服务进行主题建模, 随后对结果采用谱聚类.

实验参数设置如下.

LDA, BTM, GPU-DMM 等模型主题数 K 设置为真实类数量, LDA 中模型 α=0.1, β=0.01; BTM 模型中参数 α=50/K, β=0.1, GPU-DMM 中 α=50/K, β=0.1; GSDPMM 中参数 α=0.01, β= 0.005, TEW-NMF 中参数设置 λ_d=0.05, λ_w=0.05, λ_t=50, 迭代次数 I=50, ω=0.5, 词嵌入维度为 100.

为更好地分析实验结果, 本文引入准确率 (*Precision*), 召回率 (*Recall*), *F*-measure 指标, 纯度 (*Purity*) 和熵 (*Entropy*) 这 5 种指标来分析实验的聚类结果, 计算公式如下所示:

$$Precision(RM_i)=\frac{|SM_i\cap RM_i|}{|SM_i|} \tag{38}$$

$$Recall(RM_i)=\frac{|SM_i\cap RM_i|}{|RM_i|} \tag{39}$$

$$F\text{-measure}=\frac{2\times Recall\times Precision}{Recall+Precision} \tag{40}$$

$$Purity(RM_i)=\frac{\max(num_{ij})}{|RM_i|},\ (1\leqslant i\leqslant Q,\ 1\leqslant j\leqslant P) \tag{41}$$

$$Purity(RM)=\sum_{i=1}^{Q}\frac{|RM_i|}{|SM|}Purity(SM_i) \tag{42}$$

$$Entropy(RM_i)=-\sum_{j=1}^{P}\frac{num_{ij}}{|RM_i|}\log\left(\frac{num_{ij}}{|RM_i|}\right) \tag{43}$$

$$Entropy(RM)=\sum_{i=1}^{Q}\frac{|RM_i|}{|RM|}Entropy(RM_i) \tag{44}$$

其中, SM_i 表示第 i 类标准分类结果, RM_i 表示第 i 类实验聚类结果, 聚类整体的准确率、召回率和 *F*-measure 由所有类的平均值表示, 精准率与召回率和 *F*-measure 越高, 则证明聚类的精准性越好. num_{ij} 表示 SM_i 和 RM_j 的交集, 纯度越高, 熵越低则说明聚类效果越好.

图 4–图 8 展示了这 8 种方法在 4–12 类 (表 2) 服务数据规模下准确率、召回率、*F*-measure、纯度与熵的表现情况. 其中, TWE-NMF+SC 方法使用 GSDPMM 自动估算主题数, 而其他 7 种基准方法均在实验前预先指定好 4、8、12 的主题数.

分析图 4–图 8 中的实验结果, 可以得出如下分析结论.

(1) 与其他方法相比, 基于 TF-IDF 的方法, 结果最为不理想, 其主要原因是因为 TF-IDF 模型仅仅使用词频信息作为依据, 无法挖掘服务之间的潜在语义和功能关系, 也没考虑到单词之间的关系, 同时 Mashup 服务描述通常为短文本, TF-IDF 模型很难区分出关键性单词权重.

(2) LDA、BTM、GPU-DMM 等主题模型, 通过主题层能更好挖掘服务之间的潜在关系, 所以相对于 TF-IDF, 这类主题模型效果普遍较好. GPU-DMM 和 BTM 优于 LDA, 主要是因为 GPU-DMM 通过预训练的词嵌入信息,

BTM 则通过词对的方式, 来获取单词间关系, 缓解了稀疏性问题, 能更好地处理短文本. LDA+API、LDA+Wiki 效果不如 LDA, 主要是因为前两者在通过外部语料库对 Mashup 服务扩充服务描述时, 引入了噪声信息. 另外在实验结果中 LDA+API 略优于 LDA+Wiki, 我们认为主要是因为 API 描述文档从语义上相对更接近 Mashup 服务, 噪声信息更少.

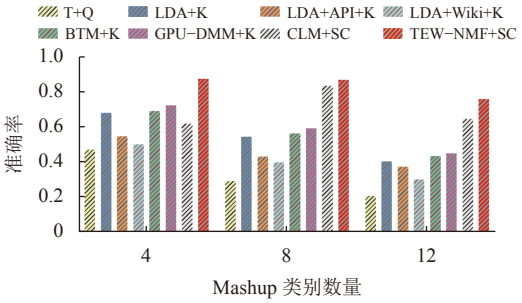


图 4 准确率

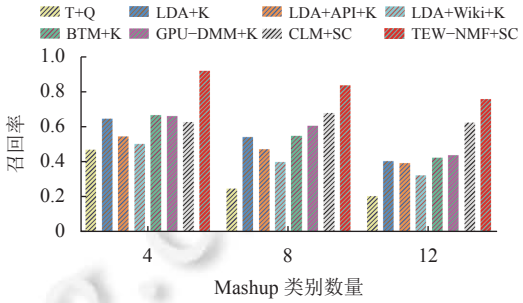


图 5 召回率

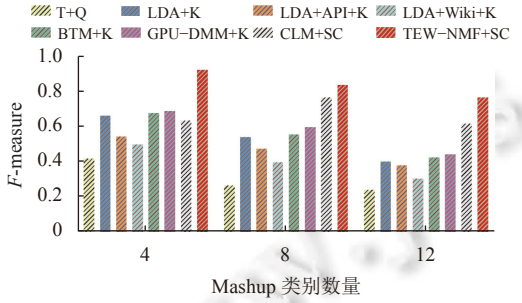


图 6 *F*-measure

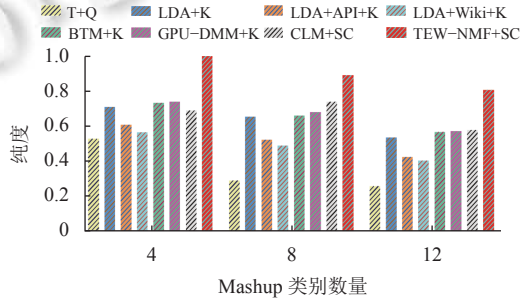


图 7 纯度

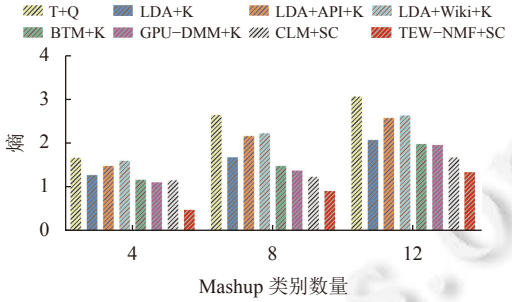


图 8 熵

(3) 仔细观察 CLM+SC 方法, CLM+SC 在 8 类、12 类时效果优于前几种方法, 在 4 类时效果却不理想, 通过分析实验数据后我们认为, 由于在 4 类时服务数量较少, 可用信息不足, 通过分解得到的词嵌入信息不理想, 同时 CLM 采用 TF-IDF 值代替原始文档-单词的词频矩阵, 由于 Mashup 服务描述较短, 无法较好的区分单词权重, 所以在 4 类时效果不佳.

(4) 对比 TEW-NMF+SC 和其他方法, 效果明显提升, 因为 TWE-NMF 同时引入了词嵌入信息和服务标签信息, 捕获了单词间的语义关系, 加大了关键单词的权重信息, 缓解了短文本带来的稀疏性问题, 能有一个较优的建模结果, 同时采用谱聚类的方式也能得到更好的聚类效果. 另外, TWE-NMF+SC 方法使用 GSDPMM 自动估算主题数 K, 在估算过程中, K 的取值虽然会有小范围的波动, 但总体和实际的主题数非常接近, 实验效果也证明了

DPMM 模型自动估计主题后能得到理想的聚类结果. 而其他基准方法需要预先指定主题数, 在现实情况中, 如果主题数未知, 容易出现主题数指定过大或过小的问题, 进而影响聚类效果.

4.2 建模可视化结果分析

为证明主题建模结果有效性, 在上述分析的基础上, 本文利用 t-SNE 工具[39], 对表 2 中 12 类服务在 LDA、BTM、GPU-DMM、CLM 和 TWE-NMF 主题建模后, 将得到的文档-主题向量进行可视化处理, 展示结果如图 9 所示.

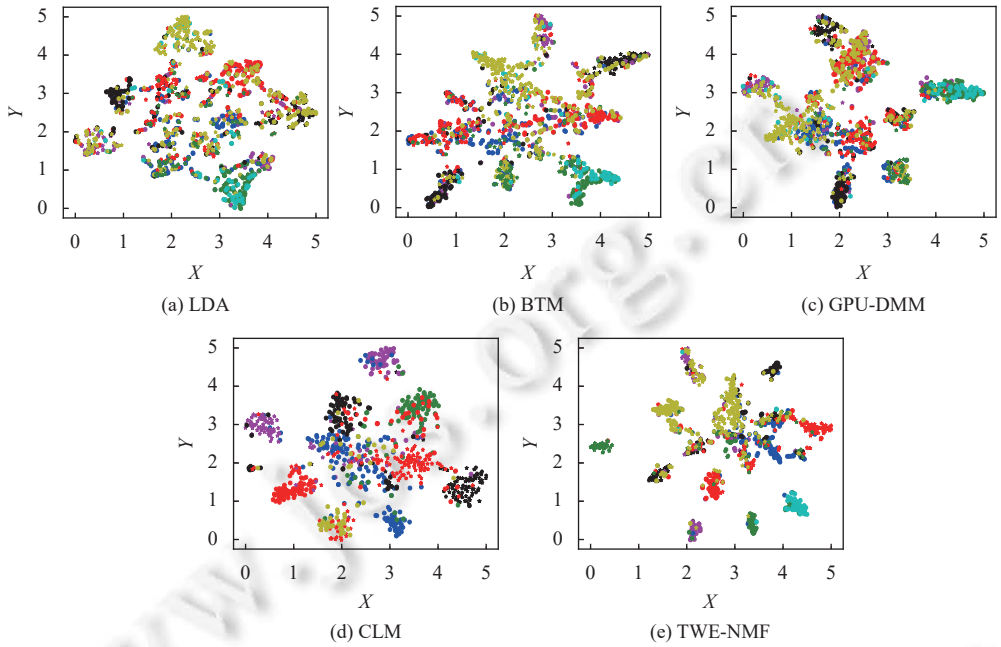


图 9 不同主题模型建模后的可视化结果

从图 9 可知, LDA 主题模型得到的特征向量可视化后分布效果最差, 类之间重叠情况严重, 同一个类分布分散, 类之间区分度不高. 相比 LDA 模型, BTM 和 GPU-DMM 对短文本处理能力更强, 因此模型得到的分布效果优于 LDA, 在可视化后类之间的区别较为明显, 但是类的重叠情况还是相对比较严重. CLM 通过在 NMF 中引入词嵌入信息模型求解主题特征, 缓解了短文本中的稀疏性问题, 从而可以得到较好的分布, 但是类之间的区分效果不如 TWE-NMF. TWE-NMF 不仅引入优化后的词嵌入信息, 还通过 TCSW 方法重新计算单词权重, 有效地缓解了短文本中信息不足的问题, 得到的服务主题特征在可视化后分布最优, 虽然类之间也存在交集, 但是从图 9(e) 中明显可以看出相同的类基本都在一起, 类别间的区分效果对于前 4 种方法明显得到了提升. 读者可运行本节程序 (https://github.com/viivan/Mashup-visualization-TWE-NMF) 来进行更深入了解.

4.3 词嵌入、单词权重及谱聚类对聚类结果的影响

为证明词嵌入、单词权重方法和谱聚类的有效性, 以 NMF 为基础, 结合不同的改进点, 我们准备了 8 种比较方法, 在 10 类服务情况下进行了针对性的实验, 其中表 3 展示了 8 种方法之间的对比情况.

表 3 各方法比较

方法	NMF+K	NMF+SC	WE-NMF+K	WE-NMF+SC	T-NMF+K	T-NMF+SC	TWE-NMF+K	TWE-NMF+SC
词嵌入	—	—	√	√	—	—	√	√
单词权重	—	—	—	—	√	√	√	√
K-means	√	—	√	—	√	—	√	—
谱聚类	—	√	—	√	—	√	—	√

● NMF+K: 通过传统的非负矩阵分解方法对 Mashup 进行主题建模, 并采用 K-means 方法对建模结果进行聚类.

● NMF+SC: 通过传统的非负矩阵分解方法对 Mashup 进行主题建模, 并采用谱聚类方法对建模结果进行聚类.

● WE-NMF+K: 在非负矩阵分解中引入词嵌入信息后对 Mashup 服务进行主题建模, 采用 K-means 方法对结果进行聚类.

● WE-NMF+SC: 在非负矩阵分解中引入词嵌入信息后对 Mashup 服务进行主题建模, 采用谱聚类方法对结果进行聚类.

● T-NMF+K: 通过 TCSW 方法重新计算单词权重信息, 代替传统非负矩阵中的文档-词频信息, 对 Mashup 服务进行主题建模, 采用 K-means 方法对结果进行聚类.

● T-NMF+SC: 通过 TCSW 方法重新计算单词权重信息, 代替传统非负矩阵中的文档-词频信息, 对 Mashup 服务进行主题建模, 采用谱聚类方法对结果进行聚类.

● TWE-NMF+K: 在非负矩阵分解中综合词嵌入信息和 TCSW 方法计算的单词权重信息对 Mashup 服务进行建模, 采用 K-means 对最后结果聚类.

● TWE-NMF+SC: 本文提出的方法, 在非负矩阵分解中综合词嵌入信息和 TCSW 方法计算的单词权重信息对 Mashup 服务进行主题建模, 采用谱聚类的方式对最后结果聚类.

NMF+K、WE-NMF+K、T-NMF+K 和 TWE-NMF+K 采用了 K-means 进行聚类, NMF+SC、WE-NMF+SC、T-NMF+ SC 和 TWE-NMF+ SC 采用了谱聚类. 从图 10 中可以看出, 在 K-means 与谱聚类效果的实验对比中, NNF+SC, WE-NMF+SC, T-NMF+SC, TWE-NMF+SC 相对于 NMF+K, WE-NMF+K, T-NMF+K, TWE-NMF+K 在纯度上提高了 5.78%、16.02%、8.40%、16.84% , *F*-measure 指标提高了 10.42%、19.35%、6.21%、12.30%, 这说明采用谱聚类的聚类结果明显得到了提高. 在进一步结合文献 [40,41] 的实验结果和分析结论后, 我们认为谱聚类相对于 K-means 有更好的聚类效果, 其主要原因有以下 3 点: (1) 由于 K-means 初始中心是随机选取的, 采用随机点作为初始聚类中心, 很容易陷入局部最优. 谱聚类则是通过计算得到的特征值作为聚类依据, 相比采取随机点作为初始聚类点, 特征值作为依据更为合理, 减少了异常聚类中心点对结果带来的影响. (2) K-means 采用欧式距离作为度量相似度的方式, 导致其只能发现球状类型的, 对于非凸形状的簇, 且易于陷入局部最优而不能发现数据集的真实分布, 谱聚类则能发现任意形状的簇且收敛于全局最优解. (3) Mashup 服务特征信息较少且相对稀疏, 而谱聚类在构建 Laplacian 特征时进行了降维操作, 对稀疏数据的处理能力更强, 因此有相对更好的结果.

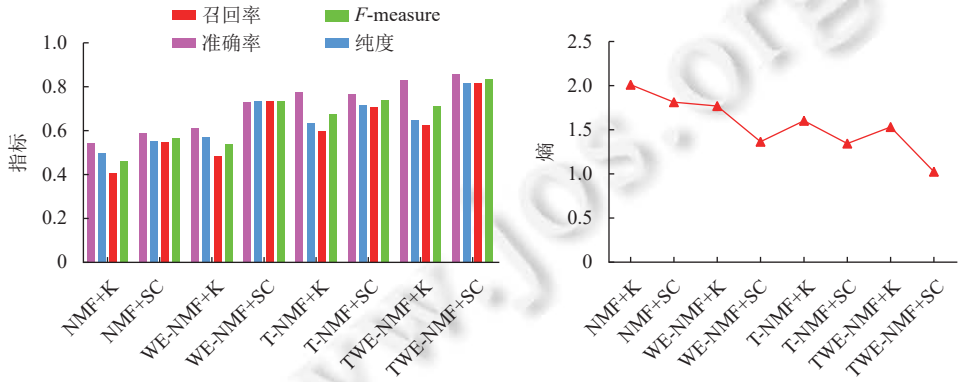


图 10 词嵌入、单词权重及谱聚类对聚类结果的影响

同样地, WE-NMF 的方法在 K-means (WE-NMF+K) 和谱聚类 (WE-NMF+SC) 下优于 NMF (NMF+K、NMF+SC) 的方法, 主要是 WE-NMF 通过词嵌入信息弥补了传统 NMF 中缺少单词语义关系的缺点. 此外, T-NMF 通过 TCSW 方法提高了关键单词的权重信息, 所以在不同聚类方法下都优于 NMF 方法. 而 TWE-NMF 综合了

WE-NMF 和 T-NMF 两者的优点, 图 10 中的结果也进一步证明本文所提方法的有效性 (本节代码可从 https://github.com/viivan/Impact-factor-on-Mashup-clustering-TWE-NMF 下载).

4.4 主题关键单词分析

使用 TWE-NMF 对 Mashup 服务主题建模后, 我们进一步对每个主题相关单词进行分析. 本实验抽取部分主题并对每个主题的 TOP10 关键词进行可视化处理, 表 4 为相关主题的 TOP10 关键词. 从表 4 可以轻易地看出这 5 类主题分别和表 2 中的 weather、travel、music、eCommerce、Telephony 这 5 大类有关.

表 4 相关主题 TOP10 关键词

Topic1	Topic2	Topic3	Topic4	Topic5
travel	price	music	phone	weather
guide	comparison	artist	call	forecast
destination	product	song	mobile	condition
information	deal	discover	number	information
blog	compare	listen	use	city
world	shop	play	app	local
trip	amazon	new	cell	provide
traveler	best	video	access	get
map	item	lyric	lose	update
community	shopping	genre	email	sport

进一步对单词在主题上的分布进行可视化展示. 观察单词的分布结果, 从图 11 中可以看出, 同一主题下, 每个主题的大多数关键单词都聚集在一起, 即语义相关的词通常属于相似的主题, 在嵌入空间中也很接近, 这说明第 3.4 节中公式 (15) 的假设是合理的 (实现详见 https://github.com/viivan/Mashup-Keyword-visualization).

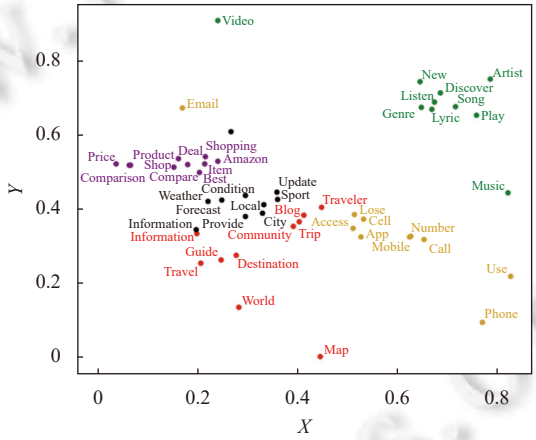


图 11 单词可视化结果

4.5 低频词结果分析

本节实验对低频词在最终实验的效果影响进行探究, 后文表 5 为不同阈值下, 删除出现次数小于阈值的单词的实验结果. 当阈值为 1 时, 则表示不删除单词. 从图 12 中可以看出, 去除低频词后, 可以有效地提升最后的聚类效果. 在不删除低频词的情况下, 类重叠的情况较为严重, 主要原因是许多无效单词干扰了正常的语义信息, 删除低频词可以减少无效单词的干扰. 由于 Mashup 服务文档描述本来就十分简短, 低频词阈值的设定是一种经验值, 需要反复进行实验调优, 同时其设定依据也和文档数量有关, 删除过多的单词又会导致语义的丢失. 本部分实验内容可以参考 https://github.com/viivan/Mashup-low-frequency-words-visualization 提供的相关程序.

4.6 主题数对实验结果的影响

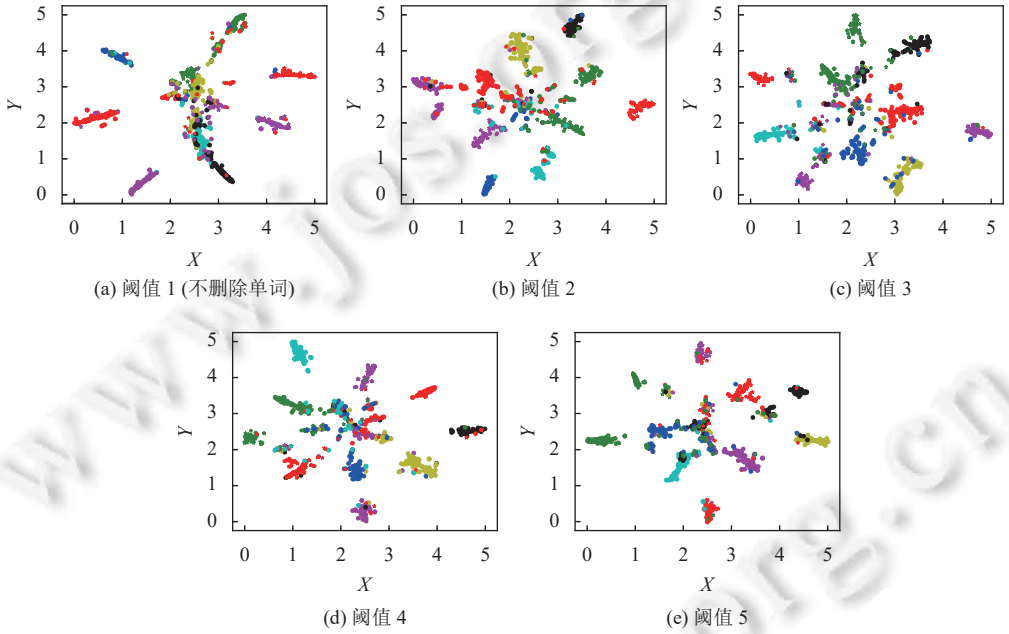
为测试不同主题数对实验结果的影响, 本节实验将主题数设为 3 个范围: 5 (小于真实主题数量), 15–45 (略大

于真实主题数量), 100–200 (远大于真实主题数量), 我们对各类主题模型, 包括 TWE-NMF 均预先指定好主题数, 来观察各项聚类指标在不同主题数下的聚类结果. 实验所用主题模型如下 (实验数据代码详见 https://github.com/viivan/Mashup-topic-number-test).

- LDA+SC: 通过 LDA 主题模型对服务建模, 采用谱聚类方法聚类.
- NMF+SC: 通过传统非负矩阵分解对 Mashup 服务主题建模, 采用谱聚类方法聚类.
- TWE-NMF+SC: 本文所提方法.

表 5　去除低频词实验结果

阈值	准确率	召回率	*F*-measure	纯度
1	0.661 23	0.575 44	0.614 53	0.630 08
2	0.860 70	0.844 19	0.852 37	0.846 39
3	0.861 17	0.797 29	0.828 00	0.798 79
4	0.864 06	0.817 84	0.840 31	0.817 02
5	0.858 17	0.816 07	0.836 59	0.815 06



(a) 阈值 1 (不删除单词)　(b) 阈值 2　(c) 阈值 3

(d) 阈值 4　(e) 阈值 5

图 12　不同阈值下 TWE-NMF 建模可视化结果

从图 13 分析可知, 主题数为 5 时, 由于设定的主题数小于真实服务类数, 许多不相关的服务被认为是同一个主题, 导致聚类效果并不佳. 当主题数为 15 时, LDA+SC, NMF+SC 和 TWE-NMF+SC 效果都得到了明显提升, 3 种方法均在主题数为 15 时取得了较优的结果, 说明最佳主题数在 15 左右. 当主题数从 15 到 200 变化时, NMF+SC 效果变差最为明显, 尤其在大主题数的情况下, 已经无法得到很好的结果. TWE-NMF+SC 表现最优, 各类指标均能有较好的结果, 尤其在主题数不是远大于真实主题数的情况下, 受到影响较小. 在较大的主题数下 3 种方法聚类效果都有下降, 对实验结果分析后, 我们认为结果变差的原因是产生过多无意义的主题, 导致主题特征无法很好区分文档. 从图 13 中主题数影响分析可知, 相对于 LDA 和 NMF, TWE-NMF 对主题数设定的容错率更高, 如果结合 DPMM 模型寻找更优的主题数, 可使结果进一步得到提高, 这从一定程度证明了 TWE-NMF 能产生较高质量的主题分布.

4.7　实验参数对结果的影响

为了探究本文提出方法参数的可靠性, 我们对 GSDPMM 和 TWE-NMF 中的参数进行实验评估, 读者可从

https://github.com/viivan/Mashup-weight-parameter-test-TWE-NMF 下载本节实验代码以了解实验效果.

4.7.1 GSDPMM 中 α 和 β 对主题数的影响

主题数的确认是通过 GSDPMM 的方法自动确认, GSDPMM 模型受到 α, β 参数的影响, 本节进一步分析参数 α, β 对主题数生成的影响.

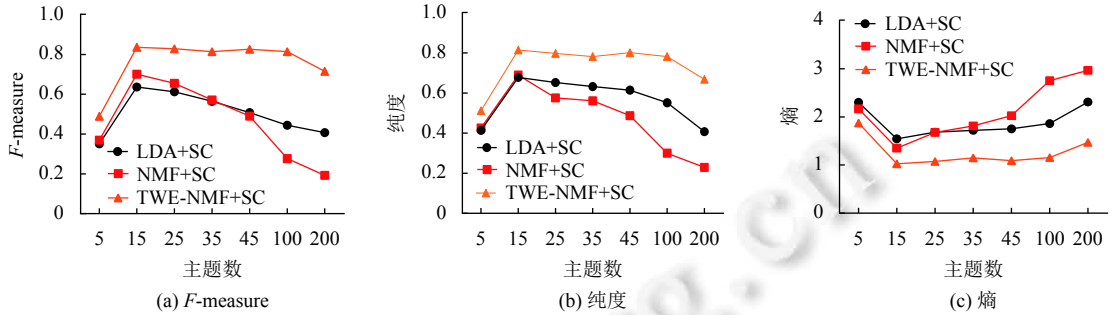


图 13 主题数对结果的影响

图 14 展示了不同 α, β 值下生成主题数的数量, 从图中可知 β 不变时 α 越大, 产生的主题数总体呈现上升的趋势, 由公式 (6) 可知 α 的大小影响新主题产生的概率, α 越大越容易产生新的主题; 当 β 值过小或者 β 值过大时, 由图中可以看到产生的主题数小于正常值. 结合第 4.6 节中的实验结果可知, 主题数在 5 左右时得到的主题数小于 Mashup 服务数据集的真实主题数, 最终聚类结果将会严重下降. 因此 β 值不宜设置的过大或过小. 结合图 13 的结果, 在 β 值设定合理的情况下, 通过 GSDPMM 方法生成主题个数在我们的 TWE-NMF 模型中均能有较好的结果.

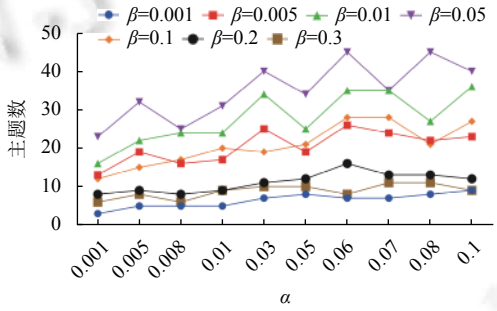
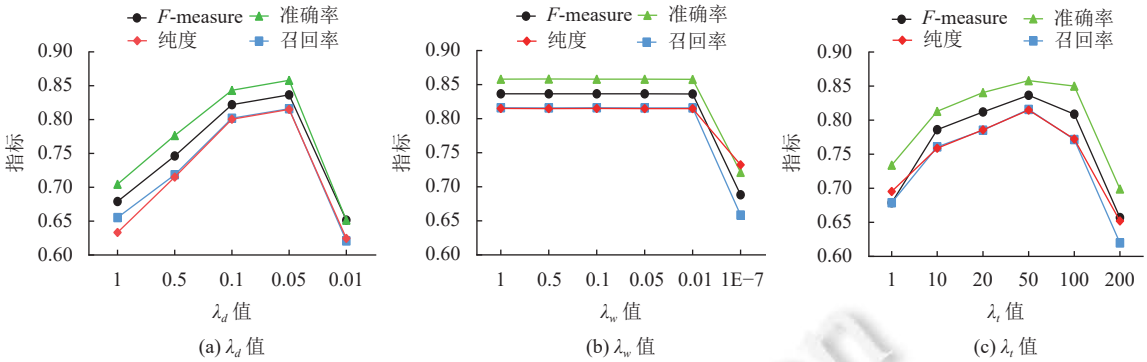


图 14 GSDPMM 参数对主题数影响

4.7.2 TWE-NMF 中权重系数对结果的影响

在第 3.4 节中我们使用公式 (16) 设置了 3 个权重系数来对目标函数调优, 本实验通过调整 3 个权重系数值进一步分析权重系数对实验结果的影响.

由图 15(a) 可知, λ_d 较小时, 可以得到较好的结果. 结合第 3.4 节, λ_d 所关联部分可由全局文档-单词关系矩阵 D 得出, 因此可以适当缩小这部分的误差权重系数. 但是当权重系数过小时, 由于过度忽略该部分的误差影响, 又会导致结果变差. 同理, λ_w 所关联部分是由已知的 SPPMI 矩阵 M 获得, 根据图 15(b) 的结果, λ_w 的权重系数保持在一个较小值对实验并不会产生太大影响, 但是当 λ_w 过小时, 因为过度忽略词嵌入矩阵部分的误差, 会进一步导致效果大幅度降低. λ_t 所关联部分是由主题-单词矩阵 T 所得. D 和 M 是可知的, T 是分解所得, 因此相对于 D 和 M 所在部分, λ_t 部分所带来的误差会对结果产生较大的影响. 参考图 15(c) 可知, λ_t 较大时效果越好, 主要是 λ_t 部分引入了未知的主题嵌入矩阵, 同时单词-主题矩阵也是由全局文档-单词关系矩阵分解得到, 都是未知的矩阵, 因此需要重点考虑这一部分的误差. 但是需要注意的是, 当 λ_t 权重过大时又会过度放大该部分的误差, 进而降低最后的实验效果.



(a) λ_d 值　(b) λ_w 值　(c) λ_t 值

图 15　TWE-NMF 参数对实验结果影响

4.7.3　TWE-NMF 中迭代次数对结果的影响

TWE-NMF 算法 (详见第 3.4.4 节) 通过不断迭代的方式对参数进行更新求解, 我们对迭代次数在实验中的影响进行了研究. 从图 16 中可以看出, 迭代 30 次之前, 由于算法还未完全收敛, 实验结果受到误差信息的影响较大, 导致聚类结果较差. 在迭代 40 次后可以看出聚类结果较为稳定, 算法基本已经收敛.

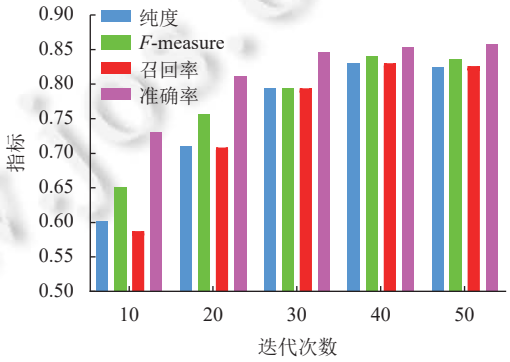
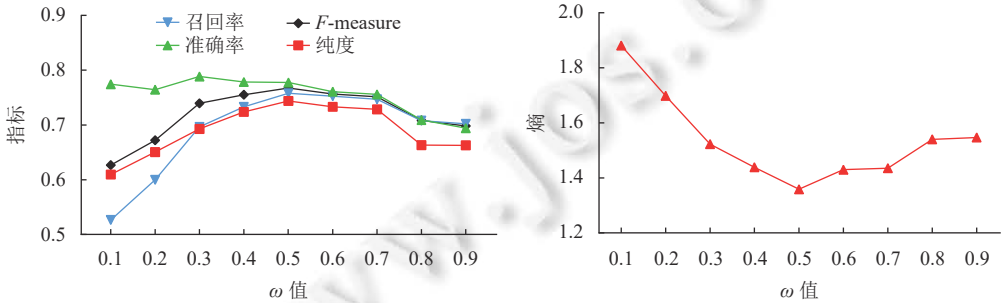


图 16　迭代次数对实验结果的影响

4.7.4　TCSW 中 ω 值对实验结果的影响

为了探究 TCSW 中超参数对实验结果的影响, 本节对 TCSW 中的超参数 ω 值行进行探究. 基于表 2 的数据集, 图 17 为 ω 值在不同情况下的实验结果.



(a) 召回率、F-measure、准确率、纯度　(b) 熵

图 17　超参数 ω 对实验结果的影响

从图 17 可以看出, 当 ω 值小于 0.5 时, 随着 ω 值的缩小实验结果也变得更差. 结合公式 (12) 进行分析, ω 值偏小代表语义权重的计算以服务标签的相似度为主. 虽然服务标签在一定程度总结了服务的功能特点, 但是仍会

存在功能特征描述不准确、不全面等问题, 过于依赖服务标签也会导致实验结果不佳. 当 ω 值大于 0.5 时, ω 值越大, 则语义权重的计算越偏向于单词上下文相似度计算. 然而 Mashup 服务描述文档通常较短, 单词的数量有限, 很有可能会忽略一些潜在的可表示服务核心功能特征的单词. 例如某服务的名词集合中有 position、location、photo 这 3 个名词, 服务标签为 photo, 最终计算出的核心单词偏向 position, 也会对后续的聚类造成影响. 从图 17 可以看出 ω 的值在 0.4–0.6 之间可以获得较好的实验效果, 因此本文将 ω 的权重设为 0.5, 综合考虑计算过程中上下文相关单词的平均相似度和服务标签的相似度, 以便获得更好的计算结果.

5 总结与展望

为了能够有效提升 Mashup 服务聚类的精度, 本文提出一种基于 TWE-NMF 主题模型的 Mashup 聚类方法. 该方法首先对 Mashup 服务进行规范化处理, 包括提取服务描述, 服务标签, API 组成等特征信息, 去除标点符号, 停用词, 低频词等干扰信息. 然后在 NMF 中引入 GSDPMM 模型, 自动确定主题数 K, 随后采用融合词嵌入和 TSCW 方法的 TWE-NMF 模型, 求解文档-主题矩阵, 最后基于谱聚类的算法将服务聚类. 为保证实验的可靠性, 我们以 ProgrammableWeb 爬取的真实数据进行实验. 实验结果表明, 我们提出方法与现有的方法相比, 能有效提高聚类的精度.

在下一阶段, 本文的研究工作主要包括 3 个方面: 1) 在计算单词语义权重时融入动词、名词的组合信息进行优化; 2) 优化谱聚类算法使其能自适应确定聚类的数量. 3) 将本文所提 Mashup 服务聚类方法与 API 推荐工作相结合, 提高 Web API 推荐精度.

References:

[1] Cao BQ, Xiao QX, Zhang XP, Liu JX. An API service recommendation method via combining self-organization map-based functionality clustering and deep factorization machine-based quality prediction. Chinese Journal of Computers, 2019, 42(6): 1367–1383 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2019.01367]

[2] Xia BF, Fan YS, Tan W, Huang KM, Zhang J, Wu C. Category-aware API clustering and distributed recommendation for automatic mashup creation. IEEE Trans. on Services Computing, 2015, 8(5): 674–687. [doi: 10.1109/TSC.2014.2379251]

[3] Li HC, Liu JX, Cao BQ, Shi M. Topic-adaptive Web API recommendation method via integrating multidimensional information. Ruan Jian Xue Bao/Journal of Software, 2018, 29(11): 3374−3387 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5482.htm [doi: 10.13328/j.cnki.jos.005482]

[4] Shi M, Liu JX, Zhou D, Cao BQ, Wen YP. Multi-relational topic model-based approach for Web services clustering. Chinese Journal of Computers, 2019, 42(4): 820–836 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2019.00820]

[5] Jiang B, Ye LY, Pan WF, Wang JL. Service clustering based on the functional semantics of requirements. Chinese Journal of Computers, 2018, 41(6): 1255–1266 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2018.01255]

[6] Shi M, Liu JX, Zhou D, Tang MD, Cao BQ. WE-LDA: A word embeddings augmented LDA model for Web services clustering. In: Proc. of the 2017 IEEE Int'l Conf. on Web Services (ICWS). Honolulu: IEEE, 2017. 9–16. [doi: 10.1109/ICWS.2017.9]

[7] Xiao QX, Cao BQ, Zhang XP, Liu JX, Hu R, Li B. Web services clustering based on HDP and SOM neural network. In: Proc. of the 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing. Guangzhou: IEEE, 2018. 397–404. [doi: 10.1109/SmartWorld.2018.00097]

[8] Shi M, Liu JX, Cao BQ, Wen YP, Zhang XP. A prior knowledge based approach to improving accuracy of Web services clustering. In: Proc. of the 2018 IEEE Int'l Conf. on Services Computing (SCC). San Francisco: IEEE, 2018. 1–8. [doi: 10.1109/SCC.2018.00008]

[9] Cao BQ, Liu XQ, Li B, Liu JX, Tang MD, Zhang TT, Shi M. Mashup service clustering based on an integration of service content and network via exploiting a two-level topic model. In: Proc. of the 2016 IEEE Int'l Conf. on Web Services (ICWS). San Francisco: IEEE, 2016. 212–219. [doi: 10.1109/ICWS.2016.35]

[10] Blei DM, Ng AY, Jordan M. Latent Dirichlet allocation. Journal of Machine Learning Research, 2001, 3: 601–608.

[11] Yan XH, Guo JF, Lan YY, Cheng XQ. A biterm topic model for short texts. In: Proc. of the 22nd Int'l Conf. on World Wide Web. Rio de Janeiro: ACM, 2013. 1445–1456. [doi: 10.1145/2488388.2488514]

[12] Li CL, Wang HR, Zhang ZQ, Sun AX, Ma ZY. Topic modeling for short texts with auxiliary word embeddings. In: Proc. of the 39th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. Pisa: ACM, 2016. 165–174. [doi: 10.1145/2911451.2911499]

[13] Das R, Zaheer M, Dyer C. Gaussian LDA for topic models with word embeddings. In: Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th Int'l Joint Conf. on Natural Language Processing. Beijing: ACL, 2015. 795–804.

[14] Teh YW, Jordan MI, Beal MJ, Blei DM. Hierarchical Dirichlet processes. Journal of the American Statistical Association, 2006, 101(476): 1566–1581. [doi: 10.1198/016214506000000302]

[15] Gao W, Chen L, Wu J, Gao HH. Manifold-learning based API recommendation for mashup creation. In: Proc. of the 2015 IEEE Int'l Conf. on Web Services. New York: IEEE, 2015. 432–439. [doi: 10.1109/ICWS.2015.64]

[16] Zhang XP, Liu JX, Cao BQ, Xiao QX, Wen YP. Web service recommendation via combining Doc2Vec-based functionality clustering and DeepFM-based score prediction. In: Proc. of the 2018 IEEE Int'l Conf. on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom). Melbourne: IEEE, 2018. 509–516. [doi: 10.1109/BDCloud.2018.00082]

[17] Quan XJ, Kit C, Ge Y, Pan SJ. Short and sparse text topic modeling via self-aggregation. In: Proc. of the 24th Int'l Conf. on Artificial Intelligence. Buenos: AAAI Press, 2015. 2270–2276.

[18] Li XM, Zhang JJ, Ouyang JH. Dirichlet multinomial mixture with variational manifold regularization: Topic modeling over short texts. In: Proc. of the 2019 AAAI Conf. on Artificial Intelligence, 2019, 33(1): 7884–7891. [doi: 10.1609/aaai.v33i01.33017884]

[19] Xu JM, Xu B, Wang P, Zheng SC, Tian GH, Zhao J, Xu B. Self-taught convolutional neural networks for short text clustering. Neural Networks, 2017, 88: 22–31. [doi: 10.1016/j.neunet.2016.12.008]

[20] Xu JM, Wang P, Tian GH, Xu B, Zhao J, Wang FY, Hao HW. Short text clustering via convolutional neural networks. In: Proc. of the 1st Workshop on Vector Space Modeling for Natural Language Processing. Denver: The Association for Computational Linguistics, 2015. 62–69.

[21] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv:1301.3781, 2013.

[22] Tian G, Zhao ST, Wang J, Zhao ZQ, Liu JJ, Guo LT. Semantic sparse service discovery using word embedding and Gaussian LDA. IEEE Access, 2019, 7: 88231–88242. [doi: 10.1109/ACCESS.2019.2926559]

[23] Zuo Y, Wu JJ, Zhang H, Lin H, Wang F, Xu K, Xiong H. Topic modeling of short texts: A pseudo-document view. In: Proc. of the 22nd ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. San Francisco: ACM, 2016. 2105–2114. [doi: 10.1145/2939672.2939880]

[24] Xun GX, Li YL, Gao J, Zhang AD. Collaboratively improving topic discovery and word embeddings by coordinating global and local contexts. In: Proc. of the 23rd ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Halifax: ACM, 2017. 535–543. [doi: 10.1145/3097983.3098009]

[25] Suh S, Choo J, Lee J, Reddy CK. Local topic discovery via boosted ensemble of nonnegative matrix factorization. In: Proc. of the 26th Int'l Joint Conf. on Artificial Intelligence. Melbourne: AAAI Press, 2017. 4944–4948.

[26] Shi T, Kang K, Choo J, Reddy CK. Short-text topic modeling via non-negative matrix factorization enriched with local word-context correlations. In: Proc. of the 2018 World Wide Web Conf. Lyon: ACM, 2018. 1105–1114. [doi: 10.1145/3178876.3186009]

[27] Chen Y, Zhang H, Liu R, Ye ZW, Lin JY. Experimental explorations on short text topic mining between LDA and NMF based schemes. Knowledge-based Systems, 2019, 163: 1–13. [doi: 10.1016/j.knosys.2018.08.011]

[28] Yin JH, Wang JY. A model-based approach for text clustering with outlier detection. In: Proc. of the 32nd IEEE Int'l Conf. on Data Engineering (ICDE). Helsinki: IEEE, 2016. 625–636. [doi: 10.1109/ICDE.2016.7498276]

[29] Guo JJ, Gong ZG. A nonparametric model for event discovery in the geospatial-temporal space. In: Proc. of the 25th ACM Int'l on Conf. on Information and Knowledge Management. Indianapolis: ACM, 2016. 499–508. [doi: 10.1145/2983323.2983790]

[30] Du N, Farajtabar M, Ahmed A, Smola AJ, Song L. Dirichlet-hawkes processes with applications to clustering continuous-time document streams. In: Proc. of the 21st ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Sydney: ACM, 2015. 219–228. [doi: 10.1145/2783258.2783411]

[31] Chen T, Liu JX, Cao BQ, Peng ZL, Wen YP, Li R. Web service recommendation based on word embedding and topic model. In: Proc. of 2018 IEEE Int'l Conf. on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom). Melbourne: IEEE, 2018. 903–910. [doi: 10.1109/BDCloud.2018.00133]

[32] Levy O, Goldberg Y. Neural word embedding as implicit matrix factorization. In: Proc. of the 27th Int'l Conf. on Neural Information Processing Systems. Cambridge: MIT Press, 2014. 2177–2185.

[33] Xie JY, Ding LJ. The true self-adaptive spectral clustering algorithms. Acta Electronica Sinica, 2019, 47(5): 1000–1008 (in Chinese with English abstract). [doi: 10.3969/j.issn.0372-2112.2019.05.004]

[34] Cai XY, Dai GZ, Yang LB, Zhang GQ. A self-adaptive spectral clustering algorithm. In: Proc. of the 27th Chinese Control Conf. Kunming: IEEE, 2008. 551–553. [doi: 10.1109/CHICC.2008.4605517]

[35] Li T, Ding C. The relationships among various nonnegative matrix factorization methods for clustering. In: Proc. of the 6th Int'l Conf. on Data Mining (ICDM'2006). Hong Kong: IEEE, 2006. 362–371. [doi: 10.1109/ICDM.2006.160]

[36] Salah A, Ailem M, Nadif M. Word co-occurrence regularized non-negative matrix tri-factorization for text data co-clustering. In: Proc. of the 32nd AAAI Conf. on Artificial Intelligence and the 30th Innovative Applications of Artificial Intelligence Conf. and the 8th AAAI Symp. on Educational Advances in Artificial Intelligence. New Orleans: AAAI Press, 2018. 489.

[37] Ailem M, Salah A, Nadif M. Non-negative matrix factorization meets word embedding. In: Proc. of the 40th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. Shinjuku: ACM, 2017. 1081–1084. [doi: 10.1145/3077136.3080727]

[38] Heyer LJ, Kruglyak S, Yooseph S. Exploring expression data: Identification and analysis of coexpressed genes. Genome Research, 1999, 9(11): 1106–1115. [doi: 10.1101/gr.9.11.1106]

[39] van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of Machine Learning Research, 2008, 9: 2579–2605.

[40] Cai D, He X, Han J. Document clustering using locality preserving indexing. IEEE Trans. on Knowledge and Data Engineering, 2005, 17(12): 1624–1637. [doi: 10.1109/TKDE.2005.198]

[41] Chang EC, Huang SC, Wu HH. Using K-means method and spectral clustering technique in an outfitter's value analysis. Quality & Quantity, 2010, 44(4): 807–815. [doi: 10.1007/s11135-009-9240-0]

附中文参考文献:

[1] 曹步清, 肖巧翔, 张祥平, 刘建勋. 融合SOM功能聚类与DeepFM质量预测的API服务推荐方法. 计算机学报, 2019, 42(6): 1367–1383. [doi: 10.11897/SP.J.1016.2019.01367]

[3] 李鸿超, 刘建勋, 曹步清, 石敏. 融合多维信息的主题自适应Web API推荐方法. 软件学报, 2018, 29(11): 3374−3387. http://www.jos.org.cn/1000-9825/5482.htm [doi: 10.13328/j.cnki.jos.005482]

[4] 石敏, 刘建勋, 周栋, 曹步清, 文一凭. 基于多重关系主题模型的Web服务聚类方法. 计算机学报, 2019, 42(4): 820–836. [doi: 10.11897/SP.J.1016.2019.00820]

[5] 姜波, 叶灵耀, 潘伟丰, 汪家磊. 基于需求功能语义的服务聚类方法. 计算机学报, 2018, 41(6): 1255–1266. [doi: 10.11897/SP.J.1016.2018.01255]

[33] 谢娟英, 丁丽娟. 完全自适应的谱聚类算法. 电子学报, 2019, 47(5): 1000–1008. [doi: 10.3969/j.issn.0372-2112.2019.05.004]

陆佳炜(1981－), 男, 副教授, CCF 专业会员, 主要研究领域为服务计算, 软件架构, 大数据可视化.

梁倩卉(1977－), 女, 博士, 讲师, 主要研究领域为数据科学, 人工智能.

赵伟(1996－), 男, 硕士, 主要研究领域为服务计算, 数据挖掘.

肖刚(1965－), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为智能制造, 云制造.

张元鸣(1977－), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为服务计算, 大数据分析, 并行计算.