计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(3):9−17 [doi: 10.15888/j.cnki.csa.006241]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于多粒度特征和混合算法的文档推荐系统①

邬登峰[1,2], 白　琳[2], 王　涛[3], 李　慧[2], 许舒人[2]

[1](中国科学院大学, 北京 100049)
[2](中国科学院 软件研究所 软件工程技术研究开发中心, 北京 100190)
[3](北京智识企业管理咨询有限公司, 北京 100101)

摘　要: 文库系统对信息的传播利用有着重要的作用, 但在文库系统中出现信息过载问题后, 数据的利用率会大大降低. 针对该问题提出了一种基于多粒度特征和混合算法的文档推荐系统, 系统在短语和词语两个粒度上对用户兴趣及文档特征进行建模, 综合基于内容推荐算法及协同过滤算法, 为用户生成兴趣列表. 系统测试数据表明, 系统在准确率、召回率、覆盖率、新颖度等指标上均有较为优异的表现, 其为用户推荐的文档较符合用户实际偏好, 有助于提升文库系统的数据利用率, 改善用户体验.

关键词: 用户兴趣模型; 文档特征; 基于内容推荐; 协同过滤; 推荐系统

引用格式: 邬登峰,白琳,王涛,李慧,许舒人.基于多粒度特征和混合算法的文档推荐系统.计算机系统应用,2018,27(3):9–17. http://www.c-s-a.org.cn/1003-3254/6241.html

Document Recommendation System Based on Multi-Granularity Features and Hybrid Algorithms

WU Deng-Feng[1,2], BAI Lin[2], WANG Tao[3], LI Hui[2], XU Shu-Ren[2]

[1](University of Chinese Academy of Sciences, Beijing 100049, China)
[2](Technology Center of Software Engineering, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)
[3](Beijing KM Consulting Co. Ltd., Beijing 100101, China)

Abstract: Document System plays an important role in information dissemination and utilization. However, with the emergence of information overload, the utilization rate of data would greatly decrease. To solve this problem, a document recommendation system based on multi-granularity features and Hybrid Algorithms is proposed. User interest and document feature models are established on both phrase and term granularities. Then, the system generates recommendation lists for users based on the combination of content-based and collaborative-filtering algorithms. The tests based on authentic data demonstrate that the document recommendation system has a better performance on precision, recall rate, coverage rate and novelty. The recommendation lists are more in line with users' interests. This helps to increase the utilization rate of data and improves user experience with better performance.

Key words: user interest model; document feature; content-based recommendation; collaborative-filtering; recommendation system

针对文库系统中文档数据量较大, 且数据量日趋增多, 会导致大量文档无法有效展现, 用户难以精准地获取到所需文档, 引发数据利用率降低的问题. 因此迫切地需要一种机器辅助功能帮助用户做一些信息筛选的工作. 传统的基于文档分类、基于用户搜索的方法, 在文档数据量达到较大规模时, 筛选能力有限, 且无法为用户发掘潜在的可能感兴趣的内容, 可以考虑引入推荐系统. 文库系统中的推荐, 本质即是文本内容的推

① 基金项目: 北京市科技计划项目 (D171100003417002)
收稿时间: 2017-06-12; 修改时间: 2017-06-27; 采用时间: 2017-07-08; csa 在线出版时间: 2018-01-25

荐, 因此可以参考新闻推荐领域的方法, 如文献[1]中的协同过滤的推荐方法, 通过计算用户行为的相似度, 为目标用户生成推荐列表, 但是协同过滤方法的冷启动问题较为严重. 文献[2]中的基于内容的推荐方法, 通过匹配用户兴趣与文本特征的相似度从而产生推荐结果, 但是这种方法只能为用户推荐与其历史兴趣相似的内容, 因此在新颖度上存在不足, 无法发掘用户潜在兴趣. 此外, 文档推荐系统的一个重要基础是精准的用户兴趣及文档特征模型, 业界一般基于文本内容抽取特征词语后采用空间向量模型来表示, 但是直接采用词语特征向量模型来计算, 存在着向量维度过高、数据稀疏的问题.

针对上述问题, 本文提出了一种基于多粒度特征和混合算法的文档推荐系统, 系统综合了用户的显式反馈和隐式反馈为用户兴趣建模, 且在用户兴趣模型和文档特征模型中分别设计了词语和短语两个粒度, 这种建模方法既保证了模型的精准度, 也避免了模型的过拟合问题. 系统还综合了时间窗口法和遗忘函数法动态更新用户兴趣模型, 确保模型的精准可靠. 在推荐方法上, 系统采用了基于内容推荐和协同过滤推荐相结合的混合推荐方法, 弥补了单一算法的不足.

1 相关技术

1.1 用户兴趣建模

推荐系统为用户提供个性化服务的过程中, 服务质量高度依赖于系统掌握的用户兴趣的准确程度, 因此确定用户兴趣模型并针对用户兴趣的变化及时更新模型也是提高推荐系统服务质量的一个重要方面.

基于向量空间模型的方法是业界常用的方法, 该方法通过抽取所采集用户兴趣数据的特征项并计算相应权值构成表示用户兴趣模型的向量[3]. 这种方法在一定程度上能够准确刻画用户兴趣, 且简单易实现, 但也存在着一些问题, 如某些标签缺乏语义明确性, 无法体现用户个性化的兴趣偏好, 且此方法中兴趣标签是离散的, 在大型应用系统中可能存在着数据稀疏的问题.

此外, 区别于文档特征模型一次建立始终有效的特点, 用户兴趣模型还面临着动态漂移的问题, 针对该问题的主流解决方案有时间窗口法和遗忘函数法. 时间窗口法[4]认为用户只对最近访问的概念感兴趣, 利用滑动时间窗滤除过时的兴趣, 时间窗口法简单易实现, 且能兼顾用户兴趣的累计计算以及历史兴趣的淘汰策略, 但是其应用效果非常依赖于时间窗口大小的取值, 而且存在用户兴趣突变的问题. 遗忘函数法[5]假设用户兴趣的变迁是一种渐进的过程, 即用户兴趣随着时间的流逝逐渐减弱, 遗忘的速度是先快后慢, 对于用户长时间没有更新的特征, 认为其不能再代表用户的当前兴趣, 可以通过遗忘函数让其不断“衰老”来达到过滤的目的[6], 如 Maloof 和 Michalski 等采用了一种遗忘函数处理用户兴趣特征[7]. 遗忘函数法符合人类记忆衰减特性, 但是该方法需要辅助设计淘汰策略, 用以淘汰无效特征, 避免大量无效特征积累影响系统效果.

1.2 推荐算法

推荐系统[8,9]是一种重要的信息过滤机制, 可以有效地解决信息过载的问题. 通过挖掘用户和信息之间的关联关系, 从而帮助用户从大量的信息中获取到他们可能会感兴趣的内容.

推荐算法是整个推荐系统最核心的部分, 在很大程度上决定了推荐效果. 业界主流算法主要有基于内容的推荐 (Content-Based Recommendation)、协同过滤 (Collaborative Filtering) 和混合推荐方法 (Hybrid Approach)[10]等.

基于内容的方法是信息检索领域的重要研究内容, 是指通过比较资源和用户兴趣模型的相似程度向用户推荐信息的方式, 该方法较多的应用在可计算的文本领域, 如浏览页面的推荐、新闻推荐等[11–14], 这种推荐方式简单有效, 不需要领域知识, 也有着比较成熟的分类学习方法能够为其提供支持, 如数据挖掘、聚类分析等, 但是也存在着一些缺点, 如很难推荐较为新颖的结果, 对新用户的推荐处理较为困难等. 业界常用的用于比对内容相似度的算法有: 余弦相似度算法, Jaccard 系数值计算法等.

协同过滤的方法是推荐系统中常用的算法之一, 其基本思想是计算用户间或项目间的相似度, 然后根据该相似度, 预测目标用户对目标项目的偏好程度而产生推荐结果[15,16]. 协同过滤算法不需要考虑被推荐项目的具体内容[17], 可以为用户提供较为新颖的推荐结果. 目前, 主流的协同过滤算法分为两类: 基于用户的协同过滤算法[18]和基于项目的协同过滤算法[19]. 基于用户的协同过滤算法根据用户对项目的偏好, 计算用户之间的相似度, 找出目标用户的最近邻居集合, 基于近邻用户的偏好为目标用户生成推荐集[20,21], 该算法能够有效地利用相似用户的反馈信息, 为目标用户产生

推荐集, 但是当用户和项目间的关联数据较少时, 无法精准计算出目标用户的相似用户群体, 此时的冷启动问题较为严重[22,23]. 基于项目的协同过滤算法根据用户操作过的项目, 对项目之间的相似度进行预测, 这在一定程度上减少了冷启动问题对推荐系统质量的影响, 但是这种方法生成的推荐集覆盖率低, 且无法为用户提供较为新颖的推荐[24].

2 系统关键技术

2.1 多粒度用户兴趣及文档特征

针对传统方法中用户兴趣及文档特征建模时存在的问题. 本系统中, 基于空间向量模型做了进一步的优化, 精细划分特征模型, 将用户兴趣及文档特征模型分为短语和词语两个粒度. 短语粒度上采用了空间向量模型来刻画用户及文档的特征, 计算用户兴趣或文档中涉及到某特征标签的次数来刻画模型, 次数越多则表明该用户或文档与该主题的相关度越高, 最终以带有权重值的特征标签向量对用户兴趣及文档特征建模, 这种方法简便易实现, 但在应用到文档推荐系统中时, 也存在一些缺点, 由于特征短语来源于海量文档内容, 短语分布离散, 过于稀疏, 无法有效地进行相似度匹配计算, 因此, 在本系统中设计了词语粒度上的特征, 相较于短语粒度的空间向量模型而言, 改善了特征标签稀疏的问题, 也提高了系统的覆盖率. 为建立上述两个粒度的特征模型, 在实现中, 需要首先从文本数据中提取出能够代表文本特征的关键短语以及关键词语, 本文基于开源中文处理项目 HanLP 提供的 API (Application Programming Interface) 来实现, HanLP 提供了标准分词、NLP 分词、索引分词、最短路径分词、词典分词等多种分词方式, 在关键短语提取方面提供了基于互信息和左右信息熵的短语提取识别解决方案, 在工程实现时, 直接调用相关接口即可, 简单易实现.

对用户兴趣建模时, 为确保用户兴趣模型的精准可靠, 系统综合了多项数据来源分析用户的兴趣特征, 包括用户的显式反馈和隐式反馈, 本文中用户显式反馈指用户在浏览文档时点击“喜欢”或“不喜欢”按钮, 显式表达偏好的行为, 这种通过显式反馈得到的用户兴趣是比较准确客观的, 但是也存在灵活性差, 对用户工作侵入性强等缺点. 用户隐式反馈是指用户使用文库系统时, 系统后台记录的用户日志, 这种隐式反馈对用户是透明的, 不会干扰用户的正常工作, 但是基于隐式反馈得到的用户兴趣准确性不够高, 存在一定的偏差. 因此, 本文中将用户显式反馈与隐式反馈相结合, 以达到更好的用户兴趣建模效果.

系统基于用户的每条显式反馈日志、搜索日志、浏览日志及下载日志, 分析该操作关联的文档标题, 从中提取出关键短语、关键词语, 得到刻画用户兴趣的特征短语空间向量及特征词语列表[25], 综合所有日志数据, 做累加操作, 得到最终的用户兴趣模型及其评分值. 系统在累加的时候对不同来源设置了不同的权重值, 权重值是基于对业务系统的分析及实际使用中数据的统计学习得到的. 用户兴趣特征短语权重评分计算公式如下:

$$Weight_{phrase} = \sum_{\forall d \in D} W_d * N \tag{1}$$

式中, D 表示所有不同来源的集合, W_d 表示各个不同来源预设的权重值, N 表示该短语出现的频次.

最终, 由这些特征短语及其权重评分值、操作日期构成用户兴趣特征 T_i, 公式如下:

$$T_i = (Phrase_i, Weight_i, Time_i) \tag{2}$$

式中, $Phrase_i$ 表示特征短语, $Weight_i$ 表示权重评分值, $Time_i$ 表示操作日期.

用户的兴趣模型由若干个独立的兴趣特征组成, 公式如下:

$$U_p = \{T_1, T_2, \cdots, T_n\} \tag{3}$$

此即为短语粒度上的用户兴趣特征空间向量.

最后, 系统将短语粒度上的用户兴趣特征空间向量作为中间结果保存到数据库中, 以备后续计算使用.

词语粒度上, 用户兴趣模型由关键词语列表组成, 公式如下:

$$U_w = \{W_1, W_2, \cdots, W_n\} \tag{4}$$

式中, W_i 表示用户兴趣特征词列表中的单个关键词语.

词语粒度的用户兴趣特征, 是基于短语兴趣特征计算而来的, 对短语特征取出全部的 *Phrase* 数据, 进行分词处理, 去重后得到一个词语列表, 即为词语粒度的用户兴趣特征, 本文中调用了 HanLP 提供的标准分词接口实现.

文档特征的建模, 类似于用户兴趣建模, 分别处理文档路径、文档标题、文档摘要以及附件内容, 结合不同来源的权重值对多个关键短语列表进行累加合并, 得到短语粒度的文档特征模型, 其中文档路径为目标文档的分级路径, 包含了其分类信息. 对合并后的特征短语空间向量做进一步的分词、去重处理, 即得到词

语粒度的文档特征.

针对用户兴趣动态漂移的问题, 系统中综合滑动窗口法和遗忘函数法更新用户兴趣模型, 既能将用户无效兴趣特征淘汰, 也能动态更新用户有效兴趣特征的权重值, 这是符合自然状态下人类兴趣漂移状态的.

时间窗口法, 基于对实际业务系统运行数据的统计分析, 设置时间窗口大小 K=15, 单位是: 日. 即对最近 K 日内用户兴趣进行累积计算. 具体计算过程如下: 系统初始化时, 分析近 K 日内所有日志数据, 得到用户的初始兴趣模型. 后续每日更新的时候, 则分析当日活跃用户日志数据, 进行如下操作:

(1) 将新出现的特征短语加入用户兴趣特征池中;

(2) 将当日涉及到的, 而用户兴趣特征池中已有的兴趣特征短语的时间戳更新为当前日期, 权重值更新为旧权重值与当日权重值之和;

(3) 将时间戳超出 K 日窗口大小的兴趣特征淘汰, 这样即可保证用户兴趣特征池中所有兴趣特征都是在 K 日窗口内的数据.

遗忘函数法, 对时间窗口法维护的特征池中每个特征短语的权重值进行衰减, 计算公式如下:

$$Weight_i = Weight_i * S \tag{5}$$

其中 S 为遗忘因子, 由参考文献[6]中提到的方法确定.

2.2 混合推荐算法

通过相关技术一节的分析可以发现, 基于内容推荐算法和协同过滤算法分别有擅长的领域及适用性较差的领域, 因此, 在本系统中, 综合了两种算法形成一个混合的推荐系统, 这样既可以发挥各自的优点, 也能弥补单一算法的不足[26].

由于协同过滤算法分为基于用户的协同过滤和基于项目的协同过滤两种, 考虑本系统的应用场景中与用户对个人兴趣传承的需求相比, 更需要关注相同兴趣群体中的热点事件, 因此, 采用了基于用户的协同过滤算法, 且从计算量和存储需求考虑, 本系统中用户数量远远少于文档数量, 采用基于用户的协同过滤算法也是较为合适的.

系统中的基于协同过滤算法, 首先计算目标用户的相似用户群, 将相似用户群中用户感兴趣的文档推荐给目标用户, 在这个过程中需要和目标用户的历史数据比对, 过滤已操作过的文档, 避免重复推荐, 算法的流程图如图 1 所示.

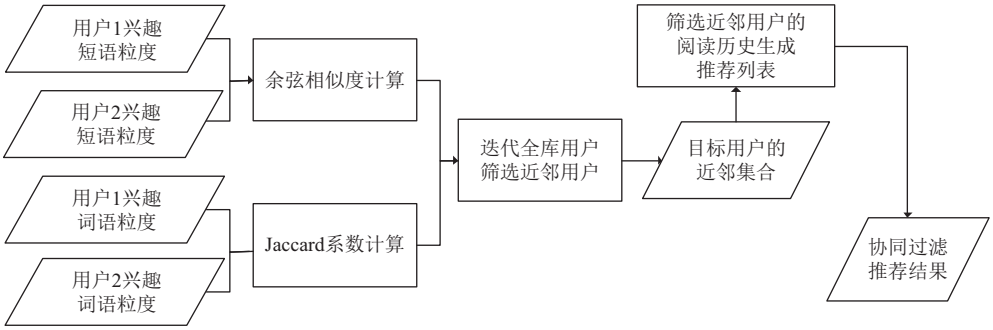

图 1 协同过滤推荐模块流程图

首先, 基于用户兴趣模型计算用户之间的相似度, 确定目标用户的相似用户群, 这里分别处理短语和词语两个粒度上的特征.

短语粒度上, 采用余弦相似度的算法来计算两个不同用户之间的相似度, 即取出两个用户的短语特征向量, 计算两者的余弦夹角值, 公式如下:

$$Score_{pcol} = \frac{\sum_{i=1}^{N} U_{w,i} U_{v,i}}{\sqrt{\sum_{i=1}^{N} U^2_{w,i}} \sqrt{\sum_{i=1}^{N} U^2_{v,i}}} \tag{6}$$

式中 U_w, U_v 分别表示两个不同用户的短语粒度上的兴趣特征向量.

词语粒度上, 采用 Jaccard 系数值来计算用户间的相似度, Jaccard 系数是两个集合交集与并集的元素数目之比, 用于测量两个集合在共同项目上的重叠度. 其匹配两个对象之间的相似度时, 仅关注特征是否存在, 而不关注该特征的权重值. 此处, 在两个不同用户的特征词语集合之间进行计算, 公式如下:

$$Score_{wcol} = \frac{|N(w) \cap N(v)|}{|N(w) \cup N(v)|} \tag{7}$$

式中 $N(w)$, $N(v)$ 分别表示两个不同用户的特征词语集合.

经过短语、词语两个粒度上的相似度计算, 系统分别得到了 $Score_{pcol}$, $Score_{wcol}$ 两个值, 将二者相加作为最终的用户间相似度, 公式如下:

$$Score_{tcol} = Score_{pcol} + Score_{wcol} \tag{8}$$

算法中将目标用户兴趣模型分别与其他用户的兴趣模型进行匹配计算, 按照相似度数值排序, 取前 M 个用户作为最近邻居集合. 这里 M 值可以结合业务系统实验确定. 最后, 从这 M 个近邻用户的历史记录中取若干篇文档作为协同过滤算法的推荐结果, 选取算法如下.

以选取 T 篇文档为例, 系统在配置文件中设置近邻个数为 M 个, 因此我们从每个近邻用户的阅读历史中取篇文档进行推送, 此处我们按照时间顺序由最近操作的文档开始向前取, 同时在取文档的时候需要与目标用户的阅读历史列表进行比对, 过滤掉目标用户已经操作过的文档, 避免重复推荐.

基于内容的推荐算法, 直接匹配用户兴趣模型和文档特征模型的相似度, 为目标用户生成推荐列表, 计算内容相似度的算法流程图如图 2 所示.

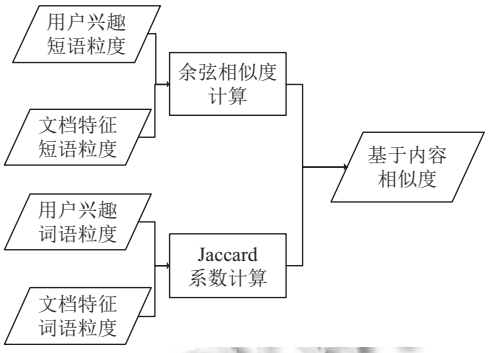

图 2 基于内容推荐模块流程图

与协同过滤部分类似, 这里分别处理短语和词语两个粒度上的特征.

短语粒度上, 基于余弦相似度算法, 计算短语粒度上用户兴趣和文档特征两个特征向量之间的余弦夹角值作为二者相似度的度量, 公式如下:

$$Score_{pcon} = \frac{\sum_{i=1}^{N} U_{p,i} D_{p,i}}{\sqrt{\sum_{i=1}^{N} U^2{}_{p,i}} \sqrt{\sum_{i=1}^{N} U^2{}_{p,i}}} \tag{9}$$

式中, U_p 表示短语粒度上的用户兴趣特征向量, D_p 表示短语粒度上的文档特征向量.

词语粒度上, 基于 Jaccard 系数值计算用户与文档的相似度, 公式如下:

$$Score_{wcon} = \frac{|N(U) \cap N(D)|}{|N(U) \cup N(D)|} \tag{10}$$

式中, $N(U)$ 表示用户特征词语集合, $N(D)$ 表示文档特征词语集合.

经过短语、词语两个粒度上用户和文档的相似度计算, 系统分别得到了 $Score_{pcon}$ 和 $Score_{wcon}$, 将二者相加作为最终的基于内容相似度匹配值, 公式如下:

$$Score_{tcon} = Score_{pcon} + Score_{wcon} \tag{11}$$

算法中将目标用户的兴趣模型分别与推荐库中的每一篇文档特征进行匹配计算, 基于相似度值 $Score_{tcon}$ 排序, 取前 Z 篇文档作为基于内容推荐的结果. 这里的 Z 值可以结合业务系统进行实验确定.

由于最终给用户展示的推荐列表只有一个, 因此需要对基于内容推荐和协同过滤推荐的结果融合, 系统中采用了预留推荐位的方法, 设最终的推荐列表中包含 A 篇文档, 融合策略中, 将前 B 个推荐位预留给基于内容推荐, 后 $(A–B)$ 个推荐位预留给协同过滤推荐, 这里 A、B 值在系统实现时均设在配置文件中, 可根据系统运行数据修改 A、B 值, 调整两种推荐算法的权重, 优化系统推荐效果. 此外, 对新注册用户处理时, 由于缺乏基础数据, 无法确定该用户的兴趣模型, 因此无法使用基于内容推荐和协同过滤算法为该用户生成推荐列表, 针对这样的冷启动问题, 系统中设置了独立的冷启动策略, 基于同部门用户的下载热度给新用户推送其可能感兴趣的文档, 即将该用户的同部门用户下载过的文档按照下载频度排序, 推荐前列的 A 篇文档.

3 系统设计与实现

本系统为服务端程序, 以文库系统中的文档作为推荐库. 首先计算出文档特征、用户兴趣模型, 再基于混合推荐算法, 生成推荐结果, 返回给文库系统, 由文库系统向用户展现. 本着高效、低耦合、分层、模块化的原则, 将系统划分为 5 个模块, 包括: 用户兴趣特征提取模块、文档特征提取模块、基于内容推荐模块、协同过滤推荐模块和推荐结果融合模块. 组织结构如图 3 所示.

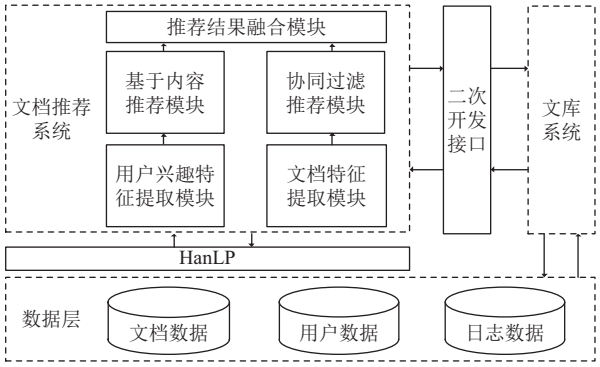

图 3 基于多粒度特征和混合算法的文档推荐系统结构图

系统通过调用文库系统提供的 API 获得文档数据、用户数据、日志数据等, 这些数据都由文库系统维护, 文库系统通过浏览器与用户交互. 基于 HanLP 提供的 API 进行分词、关键短语提取、关键词语提取. 综合对文档数据、用户数据、日志数据的分析, 为文档特征、用户兴趣建模. 从而进行基于内容的推荐、协同过滤推荐并将推荐结果融合. 推荐结果会再次通过文库系统提供的 API 存回数据库中, 供文库系统使用. 本系统也提供若干 API 供文库系统二次开发使用.

3.1 文档特征提取模块

本模块中实现了文档特征建模的算法, 分析文档的主要信息, 包括文档路径、文档标题、文档摘要、附件内容等, 基于 HanLP 提供的 API 进行分词, 提取关键短语、关键词语, 建立文档的特征模型, 模块的类图如图 4 所示.

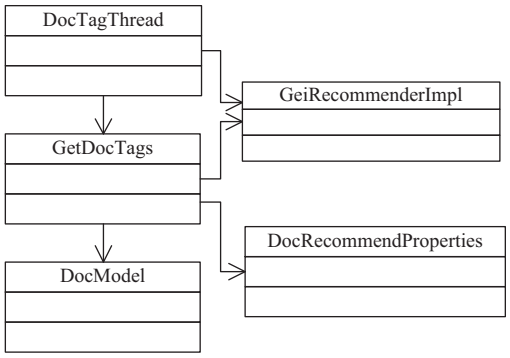

图 4 文档特征提取模块类图

为提升计算效率, 模块实现时采用了多线程的解决方案. 其中, DocTagThread 为文档特征提取的线程, 通过调用 GetDocTags 类中提供的方法, 分别从文档路径、文档标题、文档摘要、附件内容中获取到文档特征, 这些文档数据则通过 GeiRecommendImpl 类中提供的 API 从文库系统中获取. 系统计算得到的文档特征也会通过 GeiRecommendImpl 类中提供的 API 存回数据库中. DocModel 类中定义了文档特征模型的数据结构. DocRecommendProperties 为加载配置文件的类.

3.2 用户兴趣特征提取模块

本模块中实现了用户兴趣建模的算法, 分析用户操作日志数据, 经过分类计算后, 融合得到用户的兴趣模型. 由于用户兴趣漂移现象的存在, 在系统日常运营时, 此模块还负责对用户兴趣模型进行更新, 模块的类图如图 5 所示.

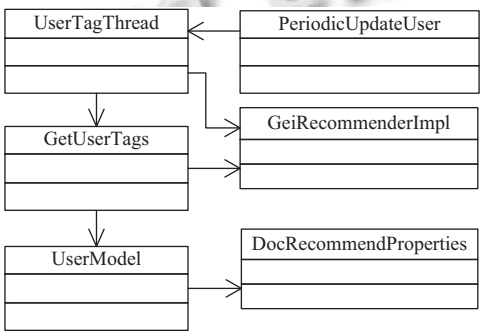

图 5 用户兴趣特征提取模块类图

为提升计算效率, 模块实现时采用了多线程的解决方案. 其中, UserTagThread 为用户特征提取的线程, 通过调用 GetUserTags 类中提供的方法, 分别从用户显式反馈、用户搜索日志、下载日志、浏览日志中提取用户的兴趣特征, 这些日志数据则通过 GeiRecommendImpl 类中提供的 API 从文库系统中获取. 系统计算得到的用户兴趣特征也会通过 GeiRecommendImpl 类中提供的 API 存回数据库中. PeriodicUpdateUser 类中提供了创建线程定时更新用户兴趣模型及推荐列表的方法. UserModel 类中的属性及方法定义了用户兴趣模型的数据结构.

由于累积 K 日窗口内的兴趣特征数量比较庞大, 且大量特征短语权重评分值较低, 使用这些特征进行相似度计算对整个推荐系统的效果提升并不是很明显, 却会增加系统的计算量, 降低系统效率. 因此, 系统中采用了以下优化策略, 选取其中权重值前 10 的特征短语组成用户短语粒度的兴趣特征向量, 而词语粒度的词语集合则由这 10 个特征短语经过分词生成. 用这样的短语特征向量和词语列表来计算相似度, 既能保证较好的推荐效果, 也降低了系统计算量, 这里的数值 10 设置在配置文件中, 可以根据系统运行效果优化调整.

基于实际业务需求考虑, 系统中对用户兴趣模型及推荐列表的更新都是采用的每日更新策略, 即夜间

系统低负载时, 对当日活跃用户更新兴趣特征模型和推荐列表.

3.3 推荐模块

推荐模块基于已建立的用户兴趣模型及文档特征模型, 使用基于内容推荐和协同过滤推荐结合的混合推荐算法为目标用户生成个性化文档推荐列表. 由于最终在文库系统中展现给用户的推荐列表只有一个, 因此还需要对推荐列表进行融合.

3.3.1 基于内容推荐模块

本模块中实现了基于内容推荐算法, 对用户兴趣模型和文档特征模型进行相似度匹配, 依据相似度数值从高到低进行筛选, 为目标用户生成推荐列表, 模块的类图如图 6 所示.

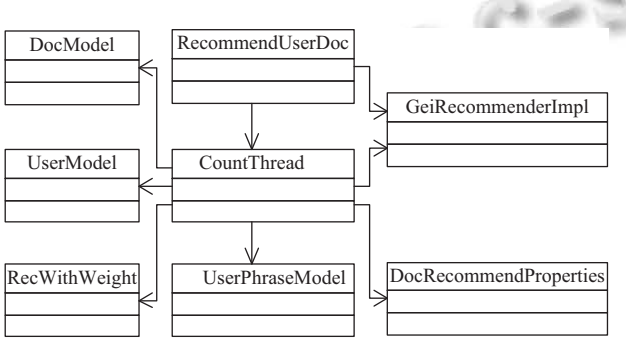

图 6 基于内容推荐模块类图

图 6 中, RecommendUserDoc 类创建了基于内容推荐计算线程 CountThread, 在该计算线程中进行相似度计算, 线程数可以由配置文件类 DocRecommend-Properties 根据配置文件设定, CountThread 计算线程中实现了余弦相似度算法与 Jaccard 系数算法, 进行词语、短语两个粒度上用户间相似度的计算, 其调用了 GeiRecommenderImpl 中提供的 API 获取存在数据库中的用户兴趣模型和文档特征模型数据, 最终的输出结果是包含文档 ID 及相似度评分值的推荐列表. 基于内容推荐结果会作为中间结果, 通过 GeiRecommenderImpl 类提供的 API 存回数据库中. RecWithWeight 类中定义了基于内容推荐列表的数据结构, 包括文档 ID 和权重.

3.3.2 协同过滤推荐模块

本模块中实现了基于用户的协同过滤推荐算法, 对不同用户的兴趣模型进行相似度匹配, 按匹配分值从高到低顺序, 计算出目标用户的相似用户群, 根据该相似用户群中用户偏好的文档为目标用户生成推荐列表, 模块的类图如图 7 所示.

图 7 中, CollaborativeFiltering 类关联其他类, 创建多线程进行相似度计算. Evaluator 类中实现了余弦相似度算法以及 Jaccard 系数算法, 进行词语、短语两个粒度上用户间相似度的计算, 其调用了 GeiRecommend-erImpl 类提供的 API 获取计算所需的基础数据. 协同过滤推荐模块的计算结果会被作为中间结果通过 GeiRecommenderImpl 类提供的 API 存回数据库中. UserPhraseModel 类中定义了短语粒度的用户兴趣模型数据结构.

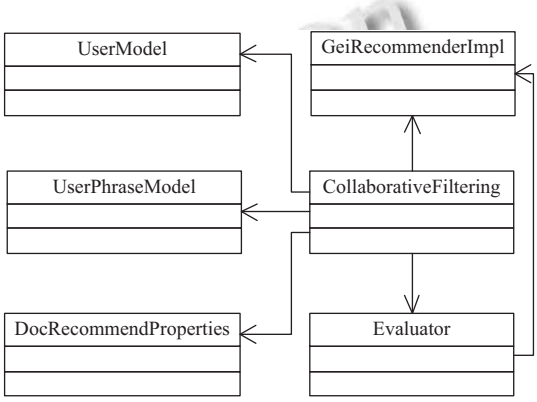

图 7 协同过滤推荐模块类图

3.3.3 推荐结果融合模块

本模块对基于内容推荐结果及协同过滤推荐结果进行融合, 并实现了面向新用户推荐文档的冷启动策略, 模块的类图如图 8 所示.

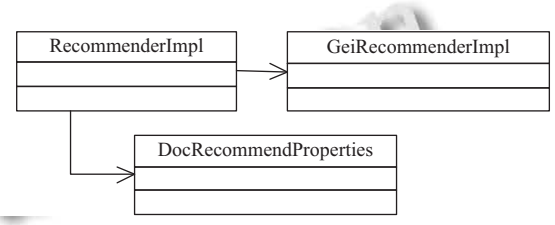

图 8 推荐结果融合模块类图

图 8 中, RecommenderImpl 类中提供的方法 getRecommendDocsByUserId() 实现了融合算法, 完成了基于内容推荐和协同过滤推荐结果的融合, 并实现了针对新用户按照同部门用户下载热度进行推荐的策略. GeiRecommenderImpl 类提供了与文库系统交互的 API.

4 实验与结果分析

本文采用了离线实验的方法评测推荐系统的效果, 基于北京市长城企业战略研究所的知识管理平台 KRP 系统进行实验, 实验数据集中包含了脱密后的 300 活跃用户数据, 17 万篇文档数据, 以及 10 日周期内的系统运行日志数据.

实验中, 以活跃用户的相邻 2 个活跃日为一个测试用例. 根据用户原有兴趣模型结合首个活跃日用户操作确定用户最新的兴趣模型, 以此为用户生成推荐列表. 将用户第二个活跃日所有操作过的且未显式标记为“不喜欢”的所有文档认定为用户感兴趣的文档, 基于此, 计算系统的准确率、召回率、覆盖率、新颖度, 并对 10 日实验周期内所有实验结果取均值, 作为系统最终评测数据. 经多次实验分析, 系统相关参数设置如表 1 所示时推荐效果较好.

表 1 实验配置

参数名称	参数值	参数名称	参数值
浏览日志权重	1	下载日志权重	2
搜索日志权重	3	文档路径权重	5
文档标题权重	2	文档摘要权重	3
附件内容权重	1	协同过滤近邻数	5
短语粒度特征个数	10	推荐列表大小	10
基于内容推荐个数	6	协同过滤推荐个数	4

关于系统的评测指标说明如下[27].

(1) 准确率

准确率描述最终的推荐列表中有多少比例是用户实际感兴趣的文档, 如下式计算:

$$Precision = \frac{\sum_{u}|R(u)\cap T(u)|}{\sum_{u}|R(u)|} \tag{12}$$

式中, $R(u)$ 表示系统为用户生成的推荐列表, $T(u)$ 表示用户行为展现的实际偏好列表.

(2) 召回率

召回率描述的是有多少用户实际感兴趣的文档包含在最终的推荐列表中, 如下式计算:

$$Recall = \frac{\sum_{u}|R(u)\cap T(u)|}{\sum_{u}|T(u)|} \tag{13}$$

式中, $R(u)$ 表示系统为用户生成的推荐列表, $T(u)$ 表示用户行为展现的实际偏好列表.

(3) 覆盖率

覆盖率反映了系统发掘长尾的能力, 覆盖率越高说明越能够将长尾中的文档推荐给用户, 如下式计算:

$$Coverage = \frac{\left|\bigcup_{u\in U}R(u)\right|}{|I|} \tag{14}$$

式中, $R(u)$ 表示系统为用户生成的推荐列表, I 表示全部文档.

(4) 新颖度

新颖度可以用被推荐的文档的平均流行度来度量, 如果推荐出的文档都很热门, 则说明推荐的新颖度较低, 否则说明推荐结果比较新颖; 计算平均流行度的时候对每个文档的流行度取对数, 这是因为文档的流行度分布满足长尾分布, 在取对数后, 流行度的平均值更加稳定. 如下式计算:

$$Novelty = \frac{\sum_{u\in U}\sum_{d\in R(u)}\log(1+P(d))}{\sum_{u\in U}|R(u)|} \tag{15}$$

式中, $R(u)$ 表示单个用户的推荐列表, d 表示单个文档, $P(d)$ 表示文档的流行度, 即该文档在所有用户感兴趣文档数据集中出现的频次.

实验中以单独的基于内容推荐列表、协同过滤推荐列表与混合推荐列表三者对比分析, 相关实验数据如表 2 所示.

表 2 不同推荐算法在实验数据集中的性能

算法	准确率(%)	召回率(%)	覆盖率(%)	新颖度
基于内容推荐	31.61	14.68	24.83	6.11
协同过滤推荐	19.53	13.33	27.69	7.36
混合推荐算法	36.25	24.29	38.73	6.97

由实验数据分析可知本系统建立的用户及文档标签较为精准可靠, 基于混合推荐算法为用户生成的推荐列表也较为契合用户实际需求. 系统的准确率、召回率、覆盖率、新颖度等指标的实验数值, 直接表明了本系统的有效性. 实际应用时, 可在更长周期内监测系统运行数据, 并结合用户调查和在线实验的测评方法, 不断优化系统参数, 提升系统综合性能.

5 结束语

本文介绍了基于多粒度特征和混合算法的文档推荐系统架构、功能模块、文档特征模型、用户兴趣模型以及推荐过程的设计与实现. 系统中综合了用户显式反馈和隐式反馈信息, 经过分词、关键短语提取、关键词语提取等操作后, 在短语粒度和词语粒度两个层面对用户兴趣建模. 系统对文档特征建模时也分别处理了短语粒度和词语粒度两个层面, 这种方法既能较为准确的刻画用户兴趣及文档特征也在一定程度上解决了数据稀疏问题, 提升了推荐系统的效果; 在此基础上, 系统使用基于内容推荐和协同过滤推荐混合的推荐算法为用户生成文档推荐列表, 此方法综合了两种推荐算法的优点, 又互相弥补了单一算法的不足. 在系统实现时, 将推荐模型相关参数都设置在配置文件

中, 增强了系统的灵活性, 也便于系统的迭代优化. 文章最后对系统的实验效果分析, 得出系统中所用的方法是有效可行的.

参考文献

1 彭菲菲, 钱旭. 基于用户关注度的个性化新闻推荐系统. 计算机应用研究, 2012, 29(3): 1005–1007.

2 Li LH, Chu W, Langford J, *et al*. A contextual-bandit approach to personalized news article recommendation. Proceedings of the 19th International Conference on World Wide Web. Raleigh, NC, USA. 2010. 661–670.

3 Vatturi PK, Geyer W, Dugan C, *et al*. Tag-based filtering for personalized bookmark recommendations. Proceedings of the 17th ACM Conference on Information and Knowledge Management. Napa Valley, CA, USA. 2008. 1395–1396.

4 Klinkenberg R. Learning drifting concepts: Example selection vs. example weighting. Intelligent Data Analysis, 2004, 8(3): 281–300.

5 Koychev I, Schwab I. Adaptation to drifting user's interests. Proceedings of ECML 2000 Workshop: Machine Learning in New Information Age. Barcelona, Spain. 2000. 39–46.

6 单蓉. 用户兴趣模型的更新与遗忘机制研究. 微型电脑应用, 2011, 27(7): 10–11.

7 Bollacker KD , Lawrence S, Giles CL. Discovering relevant scientific literature on the Web. IEEE Intelligent Systems and Their Applications, 2000, 15(2): 42–47. [doi: 10.1109/5254.850826]

8 Ricci F, Rokach L, Shapira B, *et al*. Recommender Systems Handbook. Berlin, Germany: Springer, 2011: 1–842.

9 王立才, 孟祥武, 张玉洁. 上下文感知推荐系统. 软件学报, 2012, 23(1): 1–20.

10 许海玲, 吴潇, 李晓东, 等. 互联网推荐系统比较研究. 软件学报, 2009, 20(2): 350–362.

11 van den Oord A, Dieleman S, Schrauwen B. Deep content-based music recommendation. Advances in Neural Information Processing Systems 26. Lake Tahoe, NV, USA. 2013. 2643–2651.

12 Lops P, de Gemmis M, Semeraro G, *et al*. Content-based and collaborative techniques for tag recommendation: An empirical evaluation. Journal of Intelligent Information Systems, 2013, 40(1): 41–61. [doi: 10.1007/s10844-012-0215-6]

13 Achakulvisut T, Acuna DE, Tulakan R, *et al*. Science concierge: A fast content-based recommendation system for scientific publications. PLoS One, 2016, 11(7): e0158423. [doi: 10.1371/journal.pone.0158423]

14 Philip S, Shola PB, Abari OJ. Application of content-based approach in research paper recommendation system for a digital library. International Journal of Advances Computer Science and Applications, 2014, 5(10): 37–40.

15 Jeong B, Lee J, Cho H. An iterative semi-explicit rating method for building collaborative recommender systems. Expert Systems with Applications, 2009, 36(3): 6181–6186. [doi: 10.1016/j.eswa.2008.07.085]

16 de Campos LM, Fernández-Luna JM, Huete JF, *et al*. Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. International Journal of Approximate Reasoning, 2010, 51(7): 785–799. [doi: 10.1016/j.ijar.2010.04.001]

17 杨家慧, 刘方爱. 基于巴氏系数和 Jaccard 系数的协同过滤算法. 计算机应用, 2016, 36(7): 2006–2010.

18 Konstan JA, Miller BN, Maltz D, *et al*. GroupLens: Applying collaborative filtering to Usenet news. Communications of the ACM, 1997, 40(3): 77–87. [doi: 10.1145/245108.245126]

19 Zhao ZD, Shang MS. User-based collaborative-filtering recommendation algorithms on hadoop. Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining. Phuket, Thailand. 2010. 478–481.

20 黄裕洋, 金远平. 一种综合用户和项目因素的协同过滤推荐算法. 东南大学学报 (自然科学版), 2010, 40(5): 917–921.

21 Lee SK, Cho YH, Kim SH. Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations. Information Sciences, 2010, 180(11): 2142–2155. [doi: 10.1016/j.ins.2010.02.004]

22 Lika B, Kolomvatsos K, Hadjiefthymiades S. Facing the cold start problem in recommender systems. Expert Systems with Applications, 2014, 41(4): 2065–2073. [doi: 10.1016/j.eswa.2013.09.005]

23 Fernández-Tobías I, Braunhofer M, Elahi M, *et al*. Alleviating the new user problem in collaborative filtering by exploiting personality information. User Modeling User-Adapted Interaction, 2016, 26(2-3): 221–255. [doi: 10.1007/s11257-016-9172-z]

24 邓爱林, 朱扬勇, 施伯乐. 基于项目评分预测的协同过滤推荐算法. 软件学报, 2003, 14(9): 1621–1628.

25 上海林原信息科技有限公司. HanLP: Han language processing. http://hanlp.linrunsoft.com/. [2015-04-02].

26 Lu ZQ, Dou ZC, Lian JX, *et al*. Content-based collaborative filtering for news topic recommendation. Proceedings of the 29th AAAI Conference on Artificial Intelligence and the 27th Innovative Applications of Artificial Intelligence Conference. Austin, TX, USA. 2015. 217–223.

27 项亮. 推荐系统实践. 北京: 人民邮电出版社, 2012.