基于 TextRank 的用户模板构建方法

段 准,刘功申
(上海交通大学 信息内容分析技术国家工程实验室,上海 200240)

摘 要:在基于内容的推荐系统中,初始用户模板的准确性对后面的推荐精度有很大影响。因此,在系统初始时,必须从少量用户信息中准确地提取出用户兴趣模板,尽可能减少噪声的引入。否则会在后期更新模板时产生偏移性问题,造成推荐的不准确。针对此问题,文中提出了一种基于 TextRank 算法建立初始模板的方法。首先对所拥有的少量用户感兴趣文本进行预处理并确定词义项,然后进行聚类,接下来对聚类得到的每个类别分别以义项为单位构建 TextRank 模型,并引入相似度影响因子、共现度影响因子、类权重影响因子对 TextRank 模型中的概率转移矩阵进行改进。迭代之后选取每个类中最为关键的若干义项进行综合,得到最终的初始用户模板。实验结果表明,该算法得到的初始用户模板较为精确,可以达到较好的推荐效果。

关键词:内容推荐算法;同义词词林;层次聚类;TextRank;图模型
中图分类号:TP301 **文献标识码**:A **文章编号**:1673-629X(2015)10-0001-06
doi:10.3969/j.issn.1673-629X.2015.10.001

Method of Building User Profile Based on TextRank

DUAN Zhun, LIU Gong-shen
(National Engineering Laboratory for Information Content Analysis Technology, Shanghai Jiaotong University, Shanghai 200240, China)

Abstract: In content-based recommendation system, the accuracy of the initial user profile has a great influence on the accuracy of recommendation later. Therefore, profile must be built as precise as possible on condition of having little user information when the system is in initial state. Otherwise, it will bring offset when updating the user profile later, which will cause inaccuracy of recommendation. A method of building initial user profile based on TextRank is presented in this paper. At first, the texts user interested in are preprocessed and the meaning of each word is determined. Then, clustering operation is done and TextRank models are built by using meaning of word as unit. Various influence factors are also introduced to make the TextRank transition probability matrix better. At last, the most important meanings of word are chosen from each cluster to build the final initial user profile. Experimental results show that the accuracy of recommendation is high by using this method.

Key words: content recommendation algorithm; Tongyici Cilin; hierarchical clustering; TextRank; graph model

0 引 言

随着互联网技术的迅猛发展,人们逐渐地从曾经信息匮乏的时代步入了信息过载的时代:过量的信息使得用户无法从中获取对自己有用的部分,信息使用效率降低。由于在信息过滤[1]中的良好表现,推荐系统成为解决信息过载问题的有效方法[2]。由此,应运而生了各种个性化推荐系统,包括 Amazon、YouTube、eBay 等诸多网站都部署了不同形式的推荐系统,在获取更好的用户体验的同时,产生了巨大的商业利润。

现阶段,常用的推荐系统有基于内容的推荐系统、协同过滤推荐系统[3]、混合推荐系统等[4]。其中基于内容的推荐是指通过用户选择对象的特征推断用户偏好,为用户推荐具有其他类似属性的对象。由于自然语言处理和机器学习等技术日趋成熟,该推荐算法在文本推荐领域有着广泛的应用。

对于基于内容的推荐算法,其典型流程为:收集用户爱好信息;建立用户模板;对待推荐文本集内的文本生成文本向量;计算用户向量与文本向量的相关系数,将相关系数高的文本推荐给用户;根据用户的反馈信息更新模板模版以提高模板精度[5]。然而,当系统添

收稿日期:2014-11-27 **修回日期**:2015-03-06 **网络出版时间**:2015-09-23
基金项目:国家自然科学基金资助项目(61472248,61171173)
作者简介:段 准(1991-),男,硕士,研究方向为推荐系统、自然语言处理;刘功申,副教授,研究方向为自然语言处理。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20150923.1509.068.html

加新用户时,会有冷启动的问题,此时新用户已标识的感兴趣文档数量很少,建立一个准确的用户模板相对困难但却至关重要。因为如果引入大量的用户不感兴趣的噪声词,接下来系统必然会推荐相关用户不感兴趣文档,模板在自学习更新时必然会造成偏移,影响整个系统的推荐效果。

文中提出了一种基于 TextRank 算法的初始模板建立方法。主要内容有:

(1)对拥有的少量用户文本进行预处理并且确定每个词的义项,这里使用《同义词词林》确定语义。

(2)聚类处理预处理后的文档,由于预先不知道目标信息集合内到底包含多少类别,文中采用自底向上的层次聚类方法。

(3)对聚类得到的每个类别分别以义项为单位构建 TextRank 模型,并引入相似度影响因子、共现度影响因子、类权重影响因子对 TextRank 模型中的概率转移矩阵进行修正。迭代之后得到每个类中最为关键的 N 个义项。

(4)将每类的关键义项进行综合,计算权重,得到最终的初始用户模板。

1　相关工作

TextRank[6] 算法源于著名的 PageRank[7] 算法。PageRank 是 Google 用于衡量特定网页价值的著名算法。其主要思想为:如果有大量网页链接到该网页或者有一个很重要的网页链接到该网页,则该网页价值会很高。一个页面的得分是由所有链向它的页面的重要性经过迭代得到的,最后得到一个页面的等级。当 PageRank 算法被引入到自然语言处理领域,就诞生了 TextRank,并且被广泛应用于文本关键词提取,其通过把文本分割成若干组成单元并建立图模型,利用投票机制对文本中的重要成分进行排序,可以实现无指导关键词抽取。

文中正是利用了 TextRank 算法提取出聚类后每个类别的关键义项,组成用户模板。

在自然语言处理中,消除歧义一直是一个重要方面。特别是对中文,由于存在着大量的一词多义和同义词现象,传统的通过字符串匹配不涉及上下文的方法很难确定两个词之间的关系。比如“红领巾”在指代小学生的时候很明显和“衣服”没关系,但是不为指代时就与“衣服”有关系。所以要想正确建立词语间的关系图,必须要先由上下文确定词语的词义。在建立初始模板时,所拥有的用户感兴趣文本并不多,所以确定词义的工作量并不大,却能对提高精度有较为明显的效果。

文中使用《同义词词林》这个外部资源,确定词语的词义,并且计算义项之间的相似度,由此建立 TextRank 模型。最终,以义项为单位,建立用户模板[8]。

2　算　法

文中提出的算法,是基于一种对实际情况的考察:当用户在搜取感兴趣文档时,每次往往会以集簇的形式获取感兴趣文档。比如当用户观看体育方面新闻时,对世界杯方面感兴趣,他往往会阅读不止一篇关于世界杯的文章。这样,尽管系统初始时用户标识的感兴趣文章不多,但由于其具有集簇性,利用此特性,就可以建立起一个相对准确的用户初始模板。整个算法的架构见图 1。

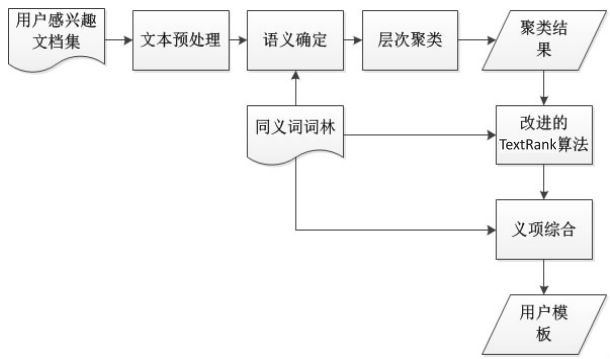

图 1　整体架构图

2.1　文本集语义确定

《同义词词林》[9] 最初是由梅家驹等于 1983 年编纂而成。哈尔滨工业大学信息检索实验室利用众多词语相关资源,完成了《同义词词林扩展版》,其收录词语近 7 万条,全部按意义进行编排,是一部同义类词典。同义词词林按照树状的层次结构把所有收录的词条组织到一起,把词汇分成大、中、小 3 类。每个小类里都有很多的词,这些词又根据词义的远近和相关性分成了若干个词群。每个段落中的词语又进一步分成了若干个行,同一行的词语要么词义相同,要么词义有很强的相关性。文中采用同义词词林确定文本中每个词的义项。

算法步骤如下:

(1)对用户感兴趣文档进行预处理,包括分词、去停用词、词性标注。

(2)因为在用户模板中,动词和名词辨识度大,价值高,所以保留其中的动词和名词,其余剔除,对于一词多义中包含多个词性的,只要含动词词性或名词词性则保留,进入下一步。

(3)确定保留词 W 的义项。

若 W 在正文中,步骤如下:

①在同义词词林中搜索 W 的所有义项,组成集合 $Q=\{s_1,s_2,\cdots,s_n,\cdots\}$,其中 s_n 为 W 的义项(如 W =“爱慕”,则 Q = { Gb09A01 = , Gb09B01 , Gb13A01 = } ,其中

Gb09A01 = 为同义词词林中的语义编码)。若 Q 中只有一个元素,则义项已确定,否则进入下一步。

②因为语义的关联不能脱离句子,此处以句子为单位,$r = R$ 为半径做一个窗口,即取 W 所在句子的前 r 个句子和后 r 个句子,挑选出窗口内所有的词,构成集合 $P = \{W_1, W_2, \cdots, W_k, \cdots\}$,其中 W_k 为 W 的邻近词。

③在同义词词林中搜索 P 中每个词的语义,每个词 W_i 对应一个语义集合 $Q_i = \{s_{i1}, s_{i2}, \cdots, s_{in}, \cdots\}$。

④计算集合 Q 中每个义项与集合 Q_i 中每个义项的相似度 $\mathrm{sim}(s_j, s_{ik})$,计算方法见文献[10]。得到 W 的义项 s_j 和集合 Q_i 的相似度定义如下:

$$\mathrm{sim}(s_j, Q_i) = \mathrm{Max}\ \mathrm{sim}(s_j, s_{ik}) \tag{1}$$

⑤对 W 的每个义项计算得分:

$$\mathrm{Score}(s_j) = \frac{\sum_{i=1}^{T} \lambda_i * \mathrm{sim}(s_j, Q)}{T} \tag{2}$$

其中,T 为集合 P 中词总数;λ_i 为修正系数,代表词 W 与 W_i 有语义关联的可能性大小。λ_i 有如下特点,在 W 所在句子中出现的所有词以及窗口中半径为 1 处的句子中出现的所有词应该具有相同的 λ 值,出现在窗口其他句子中的词随着距离的增大,与 W 有语义关联性的可能性减小。为计算方便,在这里把距离 W 大于 N 的句子视为没有语义相关性,这是因为与每个词最相关的词往往出现在本句以及相邻句子中。所以得到 λ_i 的表达式如下:

$$\lambda_i = \begin{cases} \dfrac{\sum_{j=0}^{1} \log \dfrac{N+2}{j+1}}{K * \sum_{j=0}^{N} \log \dfrac{N+2}{j+1}}, 0 \leqslant r_i \leqslant 1 \\ \dfrac{\log \dfrac{N+2}{r_i+1}}{L_i * 2 * \sum_{j=0}^{N} \log \dfrac{N+2}{j+1}}, 1 < r_i \leqslant N \end{cases} \tag{3}$$

其中,N 为前面选取窗口的半径;K 为 W 所在句子,W 左邻句子,W 右邻句子三个句子中除了 W 外的所有词汇数量;r_i 表示词 W_i 在窗口中距离中心的位置,$r_i = 0$ 表示该词汇与 W 在同一句中;L_i 表示 W_i 所在句子中的词汇数量。

⑥从 W 的语义集合 Q 中选取得分最高的义项作为 W 在此处的义项。

若 W 在标题中,确定义项方法如下:

①与 W 在正文中时操作①相同。

②对正文中的所有保留词汇进行词频统计,保留频率最高的前 M 个词,组成集合 $P = \{W_1, W_2, \cdots, W_M, \cdots\}$。

③在同义词词林中搜索 P 中每个词的语义,每个词 W_i 对应一个语义集合 $Q_i = \{s_{i1}, s_{i2}, \cdots, s_{in}, \cdots\}$。

④得到 W 的义项 s_j 和集合 Q_i 的相似度。

⑤对 W 的每个语义计算得分:

$$\mathrm{Score}(s_j) = \frac{\sum_{i=1}^{M} \mathrm{sim}(s_j, Q_i)}{M} \tag{4}$$

⑥取得分最高的作为标题中 W 的义项。

2.2　用户已标识文本聚类

在确定过文本中每个保留词的义项后,需要对这些文本进行聚类处理。在聚类算法中,最常见的两类聚类技术为划分方法和层次方法。对于划分方法而言,聚类之前需要事先指定 K 值和初始划分;对于层次聚类则不需要,它是在预先不知道目标信息集合内到底包含多少类别的情况下,将所有的信息组成不同的类。所以根据文中的应用场景,此处使用自底向上的层次聚类方法。由于层次聚类算法较为经典,具体步骤不再进行赘述。文中对该算法的略微改动之处在于:

(1)每篇文章以义项为单位,不以关键词为单位,用一个向量表示。

(2)由于当某个义项出现在标题中时显然比出现在正文中更有价值,为了聚类更加准确,这里采用改进的 tf/idf[11] 公式计算向量中每个义项的权重,见文献[12],对出现在标题中义项的贡献度进行调整。

聚类处理后,用户表示的感兴趣文本被分成多个小文本集,文本集中每个文本的保留词词义已经确定。

2.3　用改进 TextRank 算法提取关键义项

经过前面的聚类,现在需要对聚类结果中的每一类别单独进行处理,最终目的是挑选出每个类别中最具价值的那些义项,这些义项可以很好地代表这一类别,这样最后融合的用户模板就能很好地反映用户兴趣之所在。文中使用了一种改进的 TextRank 算法,提取义项。

TextRank 源于 PageRank 算法,其主要思想就是一个网页如果有越多的网页链接到它,其价值越高,如果指向它的网页越有价值,那么这个网页给待评价页面的投票就越有价值。TextRank 算法就是 PageRank 在自然语言处理方面的应用,用于判断词语的价值。

文中需要提取一个类别中的关键义项。显然,在一个类别中最具有概括性的义项会和该类别中很多其他义项有相关关系。并且只有很多有价值的义项和该义项相关,该义项对于该类别才有价值。基于这个思想,文中使用 TextRank 算法提取义项。TextRank 的迭代模型在理论上支持带权运算,但是传统的 TextRank 模型是基于影响力均分的,显然,这是不合理的[13]。如果图模型中一个相连的词语越重要,或者图中的一个链接对于最后的评估越有价值,那么对应的词理应

分取更多的价值。文中用各种影响力因子对此情况进行改进。对每个类别同样地采用下面的操作。

2.3.1 核心义项提取图模型构造

(1)为了降低维数以及减少噪声,首先对每篇文章中的义项通过前面使用过的改进 tf/idf 公式进行排序,剔除排名最后的 K 个义项。

(2)将该类别中每篇文章剩下的义项进行综合,取并集,得到集合 $Q=\{s_1,s_2,\cdots,s_i,\cdots\}$。其中,$s_i$ 为该类别中出现过的义项。

(3)以集合 Q 中每个元素作为节点做图。和前面一样,计算每个节点之间的义项相似度,如果相似度大于阈值,则在节点之间建立一条边[14]。这样就得到了一个无向图模型 $G=(V,E)$。其中,$V=\{s_1,s_2,\cdots,s_i,\cdots\}$,若 $\mathrm{sim}(s_i,s_j)>T$,则 $<s_i,s_j>\in E$。其中 T 为相似度阈值。

2.3.2 建立评分转移概率矩阵

在该模型中,评分转移概率矩阵可以反映一个义项的价值传递给相邻每个节点的概率大小。该矩阵记为 $\boldsymbol{S}$。

$$\boldsymbol{S}=\begin{bmatrix} p_{11} & p_{21} & \cdots & p_{n1} \\ p_{12} & p_{22} & \cdots & p_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ p_{1n} & p_{2n} & \cdots & p_{nn} \end{bmatrix} \tag{5}$$

其中,p_{ij} 即为义项 i 的价值转移给义项 j 的概率。所以矩阵每一列的和为 1。修正后的矩阵记为 $\boldsymbol{G}$,见等式(6)。

$$\boldsymbol{G}=\alpha\boldsymbol{S}+(1-\alpha)\frac{1}{n}\boldsymbol{U} \tag{6}$$

其中,α 为阻尼系数,此处取 0.8;n 为图中节点的总数,这里即图中所有义项的总;$\boldsymbol{U}$ 是一个全 1 矩阵。现在要寻找 $\boldsymbol{G}$ 的特征向量,即满足 $\boldsymbol{q}=\boldsymbol{G}\boldsymbol{q}$ 的向量。求法为任意选取一个 $\boldsymbol{q}^{\text{current}}$ 开始,迭代计算见等式(7),当 $\boldsymbol{q}^{\text{next}}$ 和 $\boldsymbol{q}^{\text{current}}$ 足够接近时结束迭代。所求向量 $\boldsymbol{q}$ 的每一维的权重就代表该义项的得分多少。

$$\boldsymbol{q}^{\text{next}}=\boldsymbol{G}\boldsymbol{q}^{\text{current}} \tag{7}$$

由此可见,首先要计算出矩阵 $\boldsymbol{S}$ 中每个概率 p_{ij} 的大小。传统方法见等式(8)。

$$p_{ij}=\begin{cases} 0, & <s_i,s_j>\notin E \\ \dfrac{1}{M}, & <s_i,s_j>\in E \end{cases} \tag{8}$$

其中,M 为与节点 s_i 相连的所有节点的总数。但是这种均分策略并没有考虑到每个节点的特殊性。首先,是基于义项间的相关性建立起的边,显然,如果义项之间相关性越大,那么 p_{ij} 就应该越大。其次,在两个义项 A 和 B 之间相关性没有那么大的情况下,如果义项 B 在该类别的很多文章都有很高的权重,那么就说明 B 很可能是足以表达该类别主题的义项,于是为了能够提取出这种概括性义项,人为地希望 A 尽可能多地给 B 投票,那么概率 p_{AB} 就应该对应地取大一些。最后,同样假设 A 和 B 在语义上相似度并不大,但是在类文档中 A 和 B 经常一起出现,那么就说明在该类特定情况下,A 和 B 极有可能是相关的,那么如果 A 是关键义项的话,B 也应该被选出,也就是说如果 A 和 B 共现性很大的话,p_{AB} 的值也应该增大。

综上,文中将转移概率 p_{ij} 拆分为三部分,见式(9)。

$$p_{ij}=aQ_{ij}+bR_{ij}+cK_{ij} \tag{9}$$

其中,p_{ij} 为节点 s_i 价值传递给 s_j 概率的大小;Q_{ij} 为相似度影响概率;R_{ij} 为类权重影响概率;K_{ij} 为共现性影响概率;$a+b+c=1$ 且 $0\leqslant Q_{ij},R_{ij},K_{ij}\leqslant 1$。

现在分别计算每个部分。

(1)相似度影响概率。

$$Q_{ij}=\frac{\mathrm{sim}(s_i,s_j)}{\sum\limits_{<s_i,s_k>\in E}\mathrm{sim}(s_i,s_k)} \tag{10}$$

其中,$\mathrm{sim}(s_i,s_j)$ 为 s_i 和 s_j 的相似度,计算方法和前面确定语义时方法相同。

(2)类权重影响概率。

此处要为每个义项计算出一个针对类别的权重,该权重可以从出现频率和分布上大体反映某些义项对于类别的重要程度。考虑到篇章中标题,正文中出现义项代表性的差别,采用类似 TF * PDF[15] 的算法。每个义项 s_i 的类权重 w_i 如下:

$$w_i=|F_{i\_\text{body}}|\exp\left(\frac{n_{i\_\text{body}}}{N}\right)+\beta|F_{i\_\text{title}}|\exp\left(\frac{n_{i\_\text{title}}}{N}\right) \tag{11}$$

$$|F_{i\_\text{body}}|=\frac{F_{i\_\text{body}}}{\sqrt{\sum\limits_{k=1}^{K}F_{k\_\text{body}}^2}} \tag{12}$$

$$|F_{i\_\text{title}}|=\frac{F_{i\_\text{title}}}{\sqrt{\sum\limits_{l=1}^{L}F_{l\_\text{title}}^2}} \tag{13}$$

其中,$\beta>1$;N 为文档的总数;F_{i_body} 为义项 s_i 在 N 篇文档的正文中出现的频率;n_{i_body} 为在 N 篇文档的正文中出现过义项 s_i 的文档数量;K 为 N 篇文档中正文出现义项的总数;同理,F_{i_title} 为义项 s_i 在 N 篇文档的标题中出现的频率;n_{i_title} 为在 N 篇文档的标题中出现过义项 s_i 的文档数量;L 为 N 篇文档中标题出现义项的总数。式(11)的左边部分反映了该义项在类别正文中的影响力,右边反映了该义项在标题中的影响力。通常标题更具有概括性,所以此处取 $\beta>1$。由此,可以计算出类权重影响概率如下:

$$R_{ij} = \frac{w_j}{\sum_{<s_i,s_k>\in E} w_k} \tag{14}$$

其中,w_j 为 s_j 的类权重。

(3)共现性影响概率。

令 K_{ij} 表示共现性影响概率,则:

$$K_{ij} = \frac{X_{ij}}{\sum_{<s_i,s_k>\in E} X_{ik}} \tag{15}$$

其中,X_{ij} 表示义项 s_i 和 s_j 的共现性得分。此处 X_{ij} 计算公式如下:

$$X_{ij} = \frac{\sum_{d\in D} \log(\mathrm{tf}(s_i,d)+1) * \log(\mathrm{tf}(s_j,d)+1)}{\log N} \tag{16}$$

其中,N 为该类中的文档总数;D 为这些文档的集合;d 代表其中一篇文章;$\mathrm{tf}(s_i,d)$ 与 $\mathrm{tf}(s_j,d)$ 分别表示 s_i 和 s_j 在文档 d 中的归一化频率。

需要注意的是 K_{ij} 分母为 0 的情况,此时为了保证不影响 p_{ij},使概率转移矩阵每列和仍为 1,取 $K_{ij} = 1/T$。其中 T 为 s_i 的边的个数。

最终,就可以得到转移概率 p_{ij}:

$$p_{ij} = \begin{cases} 0, & <s_i,s_j> \notin E \\ aQ_{ij} + bR_{ij} + cK_{ij}, & <s_i,s_j> \in E \end{cases} \tag{17}$$

由此,如前面所述,可以得到修正后的转移矩阵 $\boldsymbol{G}$,计算特征向量 $\boldsymbol{q}$,$\boldsymbol{q}$ 每一维的权重就是图模型中每个语义的最后影响力得分。接下来,对权重进行排序,挑选出前 M 个作为关键语义用于构建用户模板。

2.4　用户初始模板生成

经过前面的几步,现在已经得到了聚类后每个类的关键义项。现在把这些义项进行综合,取并集,得到一个体现用户真实兴趣的义项集,然后可以通过多种方法计算这些义项的权重(如对每篇文章中该义项的改进 tf/idf 权重求平均),得到一个义项向量,这个向量就是用户的初始模板。同样的,对测试文本以这些关键义项生成向量,计算与模板的相似度,确定是否应该推荐该测试文本。推荐后,也可以对该模板进行动态更新。

3　实验结果及其分析

3.1　数据集及相关工具

在文中的实验中,数据集中的文章源于新浪新闻等网站。笔者通过网络爬虫从各个板块进行新闻获取,构建数据集。整个数据集包含军事类新闻 1 877 篇,经济类新闻 1 100 篇,科技类新闻 1 061 篇,体育类新闻 1 600 篇,旅游类新闻 1 005 篇,教育类新闻 1 000 篇。每条新闻均包含正文部分及标题部分。

除数据集外,文中使用了分词工具以及词性标注工具。其中,分词工具为 stanford segmenter,词性标注工具为 stanford postagger。

3.2　度量标准

实验基于以下思想:从类别集合 $\bar{S}$ 中选取少许文章作为用户感兴趣文档集,通过上述算法获取用户模板。再从类别集合 S 中选取大量文章作为待推荐文本,其中 $\bar{S} \subseteq S$(如 $\bar{S}$ = {经济,旅游},S = {经济,旅游,科技,教育})。计算文本与模板的相似度可以得到推荐文本集 Q。若 Q 中属于类别 $\bar{S}$ 的文档数越多,则推荐效果越好。则定义推荐准确度 R 如下:

$$R_{S\bar{S}} = \frac{|C'|}{|C|} \tag{18}$$

其中,$|C|$ 为推荐文本数目;$|C'|$ 为推荐文本集中属于类别 $\bar{S}$ 的文档数目。可知 $0 \leqslant R_{S\bar{S}} \leqslant 1$。

注意事项:由于使用的是层次聚类,文中算法实际可以为每个大类别各自生成一个用户模板,对每一类别进行粒度更加细化的推荐,比如单独为体育类生成一个用户模板,从体育类文章中选择用户感兴趣项目进行推荐。此处,考虑到数据集,为了便于度量以及和其他算法的比较,只为几个大类生成一个用户模板。

3.3　实验结果

3.3.1　相关系数选取

通过多次实验,选取推荐效果较好时所对应的各个系数。其中计算类权重影响概率时,系数 $\beta = 2$;合成转移概率时系数 $a = c = 0.3$,$b = 0.4$。

3.3.2　准确率实验

设用户感兴趣文档集文本数目为 N,总类别集合为 S,感兴趣类别集合为 $\bar{S}$。选取数据集中语料质量较好的旅游类、经济类、军事类、体育类构成 S。即 S = {旅游类,经济类,军事类,体育类}。现以 N = 40 为例,进行实验分析。

为全面检测实验效果,构建待实验类别集合 $\bar{S}_1$ = {旅游类,经济类},$\bar{S}_2$ = {旅游类,军事类},$\bar{S}_3$ = {旅游类,体育类},$\bar{S}_4$ = {经济类,军事类},$\bar{S}_5$ = {经济类,体育类},$\bar{S}_6$ = {军事类,体育类},$\bar{S}_7$ = {旅游类,经济类,军事类},$\bar{S}_8$ = {旅游类,经济类,体育类},$\bar{S}_9$ = {旅游类,军事类,体育类},$\bar{S}_{10}$ = {经济类,军事类,体育类}。在进行 $\bar{S}_1$ 的准确度测试时,40 篇文档平均从旅游类、经济类各取 20 篇,训练用户模板,再从四个类别中随机各选 200 篇文档,构成 800 篇的待推荐文本集,计算待推荐文本与模板间的相似度,取相似度最高的

200 篇文档作为推荐集,计算推荐准确度 R。重复 10 次,取 R 的平均值作为 $\bar{S}_1$ 的准确度。计算 $\bar{S}_i$ 的准确度使用相同的方法。最终得到 $N=40$ 时的准确率统计表,如表 1 所示。

表 1　$N=40$ 时准确率统计表

$\bar{S}$	$\bar{S}_1$	$\bar{S}_2$	$\bar{S}_3$	$\bar{S}_4$	$\bar{S}_5$	$\bar{S}_6$	$\bar{S}_7$	$\bar{S}_8$	$\bar{S}_9$	$\bar{S}_{10}$
$\bar{R}$	0.84	0.835	0.86	0.9	0.91	0.82	0.98	0.96	0.98	0.97

对表 1 中的所有推荐准确率求平均,得到 $N=40$ 的推荐准确率 $R_{N=40}=0.905\ 5$。使用相同的方法对 $N=60$, $N=80$, $N=100$ 进行检查,得到推荐准确率分布,如图 2 所示。可见随着训练文本数量的增多,准确率有所上升。

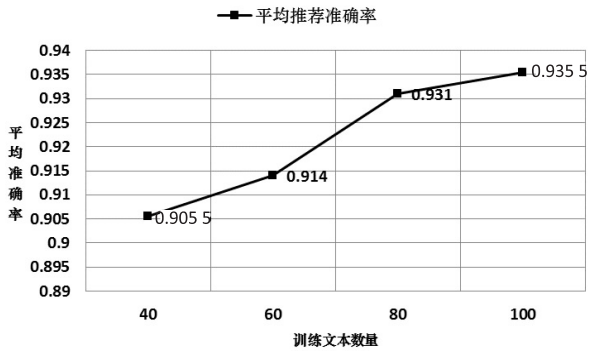

图 2　文本数量与准确度对应关系

3.3.3　对比实验

在有少量训练集的情况下,最初构建用户模板的方法为将整个文档集进行简单的关键词筛选,然后由 tf/idf 为每篇文章生成向量,以这些向量的质心作为用户模板。在此基础上,以语义为基本单位代替以词为基本单位提高了推荐效果。文中先对文档集进行语义确定及筛选,然后为每篇文章以语义为单位生成向量,并且计算质心。和 3.3.2 相同,分别对 $N=40$, $N=60$, $N=80$, $N=100$ 的情况用该模板进行推荐,与文中所提出算法的准确率结果进行对比。比较结果见图 3。

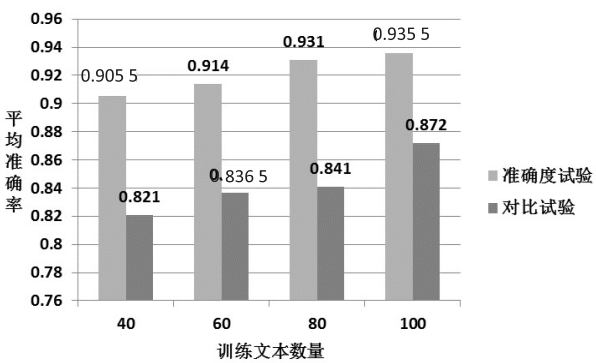

图 3　准确率比较

从图中可见,文中的算法对准确率的确有提升作用。

4　结束语

在基于内容的推荐系统中,当添加新用户时,由于此时新用户已标识的感兴趣文档数量很少,建立一个准确的用户模板相对困难但却至关重要,会对以后的推荐准确性以及模板更新产生很大影响。文中通过确定词义项,聚类,建立 TextRank 图模型等操作,建立一个较为准确的初始用户模板,减少了噪声的引入。实验结果表明,改进后的算法可以取得更好的推荐效果。

参考文献:

[1] Belkin N J, Croft W B. Information filtering and information retrieval: two sides of the same coin[J]. Communications of the ACM, 1992, 35(12): 29-38.

[2] 项　亮. 推荐系统实践[M]. 北京: 人民邮电出版社, 2012.

[3] Rich E. User modeling via stereotypes[J]. Cognitive Science, 1979, 3(4): 329-354.

[4] 刘建国, 周　涛, 汪秉宏. 个性化推荐系统的研究进展[J]. 自然科学进展, 2009, 19(1): 1-15.

[5] Pazzani M, Billsus D. Learning and revising user profiles: the identification of interesting web sites[J]. Machine Learning, 1997, 27(3): 313-331.

[6] Mihalcea R, Tarau P. TextRank: bringing order into texts [C]//Proceedings of empirical methods in natural language processing. [s. l.]: [s. n.], 2004.

[7] Langville A N, Meyer C D. Google's PageRank and beyond: the science of search engine rankings[M]. Princeton: Princeton University Press, 2006.

[8] Degemmis M, Lops P, Semeraro G. A content-collaborative recommender that exploits WordNet-based user profiles for neighborhood formation[J]. User Modeling and User-adapted Interaction, 2007, 17(3): 217-255.

[9] 梅家驹, 竺一鸣, 高蕴琦, 等. 同义词词林[M]. 上海: 上海辞书出版社, 1993.

[10] 田久乐, 赵　蔚. 基于同义词词林的词语相似度计算方法[J]. 吉林大学学报: 信息科学版, 2010, 28(6): 602-608.

[11] Salton G, Buckley C. Term-weighting approaches in automatic text retrieval [J]. Information Processing & Management, 1988, 24(5): 513-523.

[12] 高　珊. 信息检索中的查询扩展及相关技术研究[D]. 武汉: 华中师范大学, 2008.

[13] 夏　天. 词语位置加权 TextRank 的关键词抽取研究[J]. 现代图书情报技术, 2013, 29(9): 30-34.

[14] 黄云平, 孙　乐, 李文波. 基于上下文图模型文本表示的文本分类研究[C]//第四届全国信息检索与内容安全学术会议论文集. 北京: 出版者不详, 2008.

[15] 迟呈英, 李　红. 基于改进 TF * PDF 算法的网络新闻热点话题检测和跟踪[J]. 计算机应用与软件, 2013, 30(12): 311-314.

基于TextRank的用户模板构建方法

作者： 段准， 刘功申， DUAN Zhun， LIU Gong-shen
作者单位： 上海交通大学 信息内容分析技术国家工程实验室,上海,200240
刊名： 计算机技术与发展 ISTIC
英文刊名： Computer Technology and Development
年，卷(期)： 2015(10)