ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.19, No.9, September 2008, pp.2329−2338
DOI: 10.3724/SP.J.1001.2008.02329

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

基于文摘的信息检索模型*

李卫疆+, 赵铁军, 臧文茂

(哈尔滨工业大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

Summary-Based Information Retrieval Model

LI Wei-Jiang+, ZHAO Tie-Jun, ZANG Wen-Mao

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

+ Corresponding author: E-mail: hrbr_richard@163.com

Li WJ, Zhao TJ, Zang WM. Summary-Based information retrieval model. *Journal of Software*, 2008, 19(9):2329−2338. http://www.jos.org.cn/1000-9825/19/2329.htm

Abstract: Summary-Based retrieval is based on the hypothesis that terms in summary should be more important than other terms not in summary. Recent developments in the language modeling approach to information retrieval have motivated the study of this problem within this new retrieval framework. In the proposed research, two approaches to summary-based retrieval, namely ranking documents directly (SQL) and smoothing documents with summaries (SBDM) are investigated. Results on TREC collections show that, with the proposed models, summary-based retrieval models can perform consistently across collections and significant improvements over document-based retrieval can be obtained. There are two main contributions in this paper. On the one hand, summarization method of retrieval-oriented is examed and effect of this method on information retrieval. On the other hand, the new retrieval model for summary-based information retrieval models is proposed.

Key words: information retrieval; language model; summarization; summary-based model; smoothing method

摘 要: 基于文摘的检索模型是基于一个假设,即出现在文摘中的词要比未出现在文摘中的词更能表达文章的主题,因此对检索贡献更大.提出了两个基于文摘的语言检索模型,一个是用文摘模型代替文档模型直接检索文件(SQL),另一个是用文摘模型平滑文档模型(SBDM).在 TREC 数据集上的实验表明,该模型能够提高检索的性能.其中,SBDM 的性能一致接近或优于传统的标准文档查询相似模型.有两个方面的贡献,一方面提出了面向检索的文摘抽取方法并考察了这些文摘方法对检索性能的影响;另一方面提出了新的检索模型,即基于文摘的检索模型.

关键词: 信息检索;语言模型;文摘;文摘检索模型;平滑方法

中图法分类号: TP311 **文献标识码**: A

随着互联网技术的迅速发展和在线资源容量的迅速膨胀,信息检索已成为主要的研究课题之一.信息检索

* Supported by the National Natural Science Foundation of China under Grant No.60736044 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant Nos.863-317-01-04-99, 2006AA01Z150 (国家高技术研究发展计划(863))

Received 2007-06-14; Accepted 2007-09-30

是为了满足用户需求,从一个文件集中选择文档的过程.当前的检索系统,通过响应用户的搜寻请求返回一个可能与用户需求相关的文件列表,并且对这个列表中的文件按照与查询请求的相关度进行了排序.

在文档列表中,文档的标题、文档开始的几个句子以及文档的链接地址都给用户显示出来.可以说,这些信息构成了每个文档的一个摘要.用户必须利用这些信息去判断哪些文档是自己要找的,以决定是否进一步阅读文档的细节.理想状态下,用户不用通过阅读全文而仅凭这些信息就能找到自己需要的.然而,仅从文档的标题和前面的几个句子很难清晰地判断文档是否与查询相关.结果,用户不得不花费一些时间来浏览全文以判断文档的相关性.有时候,因为文档很长,相关信息很分散等原因,即使用户浏览全文也很难快速作出判断.

此外,在传统的 IR 系统中总是假定文档的所有部分都是与主体密切相关的.然而事实并非总是如此.如果把一个来自文档非相关部分的查询词误认为是相关的,则将导致查询偏差.

同时,去除文档中的不相关部分和与主题关系不紧密的部分将会有利于提高检索的性能.为了解决这些问题,许多学者试图提取文档中相关信息丰富的部分.其中一种方法称为段落检索(passage retrieval)[1].这种方法的主要优点是它提供了文档中相关信息片断分布的直观概括,从而使用户可以很容易地判断文档是否与自己的需求相关.然而,这种方法也未能减轻参考文档全文的负担,并且它也没有对检索性能有显著的提高.

我们发现,当用户在搜索待检索的文件时,如果一个句子的包含用户提交的查询词,那么用户很可能认为这个文档是与自己的查询相关.而且,这个词应该与文章的主题相一致,并出现在相关信息丰富的部分.另一个问题是,许多文档是多个主题的,也就是说,这些文档中的每个文档都讨论多个主题.这就意味着查询返回的文档中只有一部分是与查询实际相关的.因此,仅仅看一个词的出现,而不确定出现的环境是片面的,容易造成查询结果的偏差.

本文提出了一个新的检索模型,试图提取文档中与信息需求相关的信息,把查询限定在代表文档信息的部分.此方法的目的在于突出文档的核心部分,同时提供足够的信息支持检索任务.检索文档的自动文摘,同时偏重于用户的查询,能够完成这个功能.

通常,文摘指的是类似于摘要的原文档的浓缩.它简洁地说明了原文档的目标、范围以及结果等信息.一个好的文摘应该反映原文档的内容,帮助读者决定是否值得进一步阅读文档.从这个意义上说,文摘能够提供一个原文档相关性的预先判断.一些文摘也包含丰富的资料,因此它们能够作为原文档的替代品.

从最初的研究开始[2,3],自动文摘的获取主要通过从原文档中选取合适的句子.每个句子按照一些条件计算一个分值,而后由分值最高的一些句子构成文摘.从这个意义上讲,应该把这种方法称为句子抽取而不是文摘更好一些.尽管此方法并没有对文档的内容作深层次的分析,但也能生成一个指示性的文摘来帮助用户作出相关性判断.

虽然许多研究正尝试通过语言产生和人工智能技术来生成条理分明的文摘[4,5].但这些方法仅仅能够应用在一些特定的领域.例如,新闻、商业报道等.相对来说,在这些领域中,领域知识容易预见和理解.到目前为止,还没有足够的迹象表明这些系统能够在可预见的将来处理不受领域限制的文本.另外,因为需要一个鲁棒的文摘产生方法以应对信息检索中可能遇到的不同类型的文档.本文所提文摘方法采用抽取句子的方式产生文摘.

许多研究者尝试利用文摘来改善检索的性能[6−8].文献[7]假设用于查询扩展的词应该从文档中与查询最相关的部分选取并且利用文摘进行查询扩展.实验结果表明,偏重查询的文摘能够有效地提升检索的性能.为了帮助用户进行相关判断,文献[6]对查询返回的文档列表中的每个文档提供偏重查询(query-biased)的文摘.与传统的检索系统仅提供文档标题和前几个句子相比,该方法有利于用户做出正确的相关性判断.这些方法都是把文摘用于检索的后处理阶段,没有真正地利用文摘进行文档检索排序.

在信息检索领域,文摘至少应该有两种应用:一种用于检索的前处理,另一种用于检索的后处理.许多学者在后处理方面做了很多工作,通过为用户提供检索结果文档的文摘来帮助用户进行更有效的相关性判断.然而,很少有关于前处理方面的文章,也就是用文摘代替文档全文作为文档源直接进行检索来改善检索的性能.

本文是关于通过加入文档摘要到检索模型中来提升检索性能的研究.首先抽取文档的摘要,本文认为,查询词出现在摘要中比出现在文档的其他部分对检索的贡献要大,因此将给这部分查询词赋予了一个额外的权重,

而不是像传统检索模型中那样赋予同样的权重.对于摘要的抽取方法,本文采用了两类方法.一种是单纯的文档(context-independent)文摘,即在做文摘时不考虑查询中出现的词的情况.另一种偏重于查询的文摘,即做文摘时考虑查询中出现的词.在 TREC-2&3 的 ad hoc 任务上的实验表明,用文摘能够提高检索的效能.

1 文摘生成

生成文摘的思想就是试图抽取文档中传达最重要信息的部分.当然,所谓的重要性依赖于产生文摘的用途.本文抽取的文摘为检索服务,因此文摘的抽取方法是面向检索的.目前有两种基本的文摘抽取方法:一种是先进行信息抽取,然后再生成文摘,另一种是直接抽取句子或词组组成文摘.由于需要一个鲁棒的文摘产生方法以应对信息检索中可能遇到的不同类型的文档,本文采用后一种方法.

抽取句子式的文摘,把文档中的句子按照某种准则打分并按最终得分降序排列,然后选取排在前面的一定数量的句子作为文摘.研究者提出了很多评价句子重要性的准则用于产生文摘,例如:句子在文档中的位置、词在文档中出现的频率、特定词或短语在文档中出现与否,句子与源文档中其他句子、词以及段落的关系等.

每个句子的分值是由构成句子的有效词和其他分值的和组成.

为了考察文档的哪些部分可用于产生文摘,本文在 TREC 的 SJM 数据集上作了小范围的实验以了解文档的属性.从 SJM 文档集中随机抽取 50 篇文档作为样本,研究这些文档中重要信息的分布.研究包括文章的题目、段落标题以及文档的整体组织结构等.用不同的参数在这个样本集上进行实验,以获得最好的文摘系统的近似参数.根据 TREC 数据集的特性,虽然样本集很小,但整个数据集与样本集在属性上有很强的一致性,因此实验具有普遍意义.

下面介绍本文生成文摘中用到的句子抽取方法.

1.1 查询因子

我们认为,用户在搜索检索文件时,如果一个句子包含用户提交的查询词,那么用户很可能认为这个文档是与自己的查询相关.分值的计算基于查询词在每个句子中的分布.本文认为一个句子包含的查询词越多,则可能传达查询要表达的信息也越多.为了产生偏重于查询的文摘,文档中的有效句子按照它所包含的查询词的数量来打分.查询分值按下式计算:

$$QueryScore=\frac{TermQuery^2}{TotalQuery}.$$

这里,*TermQuery*=句子中包含查询词的个数;*TotalQuery*=查询中的词的数目.

1.2 标题因子

文章的标题往往反映文章的主题.通过对 TREC 文档的简单分析,这个假设得到验证.因此,句子重要性的另一个因子是标题词在句子中出现的次数.

每个句子的标题分按如下公式计算:

$$TitleScore=\frac{TitleTerms}{TotalTerms}.$$

这里,*TitleScore*=句子的标题得分;*TitleTerms*=文档句子中包含标题词的数量;*TotalTitle*=标题中包含的词的数量.

TotalTitle 是归一化因子.是为了确保在句子的整个重要性分值中,标题分值不会占到过多的比重.

1.3 位置因子

Edmundson[3]发现句子在文章中的位置对于确定句子的重要性常常是有用的.为了确定这种根据句子在文档中的位置来给句子打分方法的效果,对 TREC 数据集作了抽样研究.研究表明,一个 TREC 文档的前面句子常常能够提供关于文档内容的重要信息.所以,本文对文档的起首两个句子根据下面的公式赋予位置分值:

$$HeadScore=\frac{1}{NumSen+SenPos},$$

其中,*HeadScore*=文档句的位置分值,*NumSen*=文档中包含的句子的数量,*SenPos*=当前句子在文档中的位置.

此外,文档的段落标题也为每个段落的内容提供重要信息.因此,每个段落的标题也应该给予相似的位置分值.然而,鉴于 TREC 文档通常的结构,文档一般由 2~3 个段组成,大多数段少于 3 个句子,甚至有的文章根本没有分段.同时,考虑到计算复杂性,文章的段落标题在本文中不予考虑.

1.4 重要词因子

为了确定文档中用于文摘的句子,需要一种度量,通过它能够对文档中所有句子的信息进行分析和评分.Luhn[2]提出方法计算词的重要性.根据 Tombrosde[9]对 TREC 文档的摘要作的研究表明,对于一个中等长度的 TREC 文档来说,重要的词最低的词频是 7.中等长度文档是指文档中句子数介于 25~40 之间.对于超出这个范围的文档,其重要性下限按下式计算:

$$\begin{cases} SignMeas = 7 + [0.1 \times (L - Ns)], & \text{if } Ns < 25 \\ SignMeas = 7 + [0.1 \times (Ns - L)], & \text{if } Ns > 40 \end{cases},$$

其中,*SignMeas*=重要性度量,*Ns*=文档中包含句子的数目:

$$L = \begin{cases} 25, & \text{if } Ns < 25 \\ 40, & \text{if } Ns > 40 \end{cases}.$$

一个句子中的重要性根据下面的公式计算:

$$WordsScore = \frac{NumWords^2}{TotalWords},$$

其中,*WordsScore*=句子重要词分值;*NumWords*=文档句子中包含重要词的个数;*TotalWords*=文档句子中包含的所有有效词的数目.

1.5 句子分值

前面几节中论述了对用于产生文摘的句子分值的各个组成部分的计算方法.每个句子的最终分值由每个单项得分经过线形插值相加而得.因此,句子的最终分值为:

$$SenScore = \lambda_1 QueryScore + \lambda_2 TitleScore + \lambda_3 WordsScore + \lambda_4 HeadScore,$$

其中, $\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1$.*SenScore*=句子的重要性分值.

把每个句子分值的单项分值计算方法设计成相互独立的子过程以方便调用.本文实验了多种组成因子的组合以获得最佳的文档摘要来满足检索的需求.

为了产生合适的文摘,对用于产生摘要的句子数量给出一个限制是必须的.在给出一个限制时,必须要考虑原始文件的长度.这里的文摘是为信息检索服务,而不是代表整个文档独立的文摘.因此,最佳的文摘长度是一个折衷,既保证文摘词有利于检索过程,同时还确保不相关的词的数量最小.为此,本文做了相关实验,确定合适的文摘长度以达到一个最佳的词汇选择.文摘的长度设定为原文档长度的 15%.

对于每个句子,计算每个单项的分值,然后加权相加得到最终的分值,同时,对文档所有句子按所得分值的降序排列.每个文档的文摘通过选取前 n 个句子组成,直到达到文档句子总数的 15%.

2 基于文摘的语言模型

统计语言模型是在所有可能的句子或其他语言单元上的一个概率分布.对于信息检索来说,语言模型就是对查询产生过程建模[10].基本思想是:对集合中的每个文档建立一个语言模型 D,然后按照由此文档模型 D 产生查询 Q 的可能性的大小对文档排序.也就是 $P(Q|D)$.通常把这种模型称为查询相似(QL)检索模型.概率 $P(Q|D)$ 可以通过不同的方式估计出来.最通用的方法是假定查询句是由一系列相互独立的词构成,于是,查询概率可以表示为每个查询词概率的乘积[11].

$$P(Q \mid D) = \prod_{i=1}^{m} P(q_i \mid D) \tag{1}$$

其中,q_i是查询句中的第 i 个查询词.$P(q_i|D)$由文档模型确定:

$$P(w|D)=\lambda P_{ML}(w|D)+(1-\lambda)P_{ML}(w|Coll) \tag{2}$$

其中,$P_{ML}(w|D)$是词 w 出现在文档 D 中的极大似然估计,$P_{ML}(w|Coll)$是词 w 出现在文档集中的极大似然估计.λ是平滑参数.对于不同的平滑方法,λ取不同的形式.例如,Jelinek-Mercer 平滑,λ简单地取 0~1 之间的任意数.而带 Dirichlet prior 的 Bayesian 平滑,λ按公式(3)计算:

$$\lambda=\frac{\sum_{w'\in D}tf(w',D)}{\sum_{w'\in D}tf(w',D)+\mu} \tag{3}$$

2.1 SQL模型

本文采用同样的方法建立文摘的语言模型,然后基于产生查询的相似度对文档进行排序,即 $P(Q|S)$.$P(Q|S)$可以按照式(1)和式(2)思想估算出来.

$$P(Q|S)=\prod_{i=1}^{m}P(q_i|S) \tag{4}$$

这里,$P(q_i|S)$是指文摘语言模型:

$$\begin{aligned}P(w|S)&=\lambda P_{ML}(w|S)+(1-\lambda)P_{ML}(w|Coll)\\&=\lambda\frac{tf(w,S)}{\sum_{w'\in S}f(w',S)}+(1-\lambda)\frac{f(w,Coll)}{\sum_{w'\in Coll}f(w',Coll)}\end{aligned} \tag{5}$$

其中,$tf(w,S)$表示 w 出现在文摘中的次数.$tf(w,Coll)$是指 w 出现文档集中的次数.与式(2)一样,λ也表示平滑系数,当采用不同的平滑方法时,这个系数取不同的形式.式(4)和式(5)构成了本文的第 1 个文摘检索模型.本文称其为 SQL 模型.这个模型是一个简单的模型,并且在实验中仅用作参考对照模型.

2.2 SBDM模型

本文的第 2 个文摘检索模型是用文档产生的文摘平滑文档模型.形式化模型如下:

$$\begin{aligned}P(w|S)&=\lambda PML(w|D)+(1-\lambda)PML(w|Coll)\\&=\lambda[\beta PML(w|S)+(1-\beta)PML(w|D)]+(1-\lambda)PML(w|Coll)\end{aligned} \tag{6}$$

其中,$P_{ML}(w|S)$是文摘模型.λ和 β 都是平滑系数,对于不同的平滑方法,它们取不同的参数形式.文档模型首先经过文摘模型平滑,然后再对文档模型利用文档集平滑.

可以认为,这个模型是两阶段平滑方法[12].但是,在概念上,本模型与文献[12]提出的模型是非常不同的.在文献[12]提出的方法中,第 1 阶段,用文档集模型平滑文档语言模型通过一个 Dirichlet prior,然后,把已经平滑的文档语言模型进一步与查询背景模型做插值平滑.而在本文的方法中,文档语言模型在第 1 阶段用文摘语言模型平滑,然后在第 2 阶段用文档集模型进一步平滑已被文摘模型平滑的文档模型.本文把第 2 个模型称为 SBDM 模型.公式(6)在实验中取得了很好的实验结果.

SBDM 检索模型是很直观、易懂的.很多现有的检索语言模型,例如查询相似模型和相关模型[13],都用了公式(2)给出的标准文档语言模型.可以通过把标准的文档语言模型替换成 SBDM 以构成文摘检索模型.我们将进一步进行这方面的比较研究.

SBDM 模型也可以看作是 3 个源的混合模型:文档、文档生成的摘要以及文档集.假定相关文档是由这个混合模型生成的.在文献[14]研究信息过滤时,提出了混合这 3 个源的不同模型.这个模型的形式如下:

$$P(w|D)=\lambda_1P_{ML}(w|D)+\lambda_2P_{ML}(w|Topic)+\lambda_3P_{ML}(w|Collection),$$

其中,$\lambda_1+\lambda_2+\lambda_3=1$;$P(w|Topic)$是用户指定的主题模型.如果用文摘模型替代主题模型并且用极大似然估计去近似表示 $P(w|S)$.那么,公式(7)给出了另一个方式基于文摘构成文档模型.本文称此模型为 SDM 模型.

$$P(w|D)=\lambda_1P_{ML}(w|D)+\lambda_2P_{ML}(w|S)+\lambda_3P_{ML}(w|Coll) \tag{7}$$

其中,$\lambda_1+\lambda_2+\lambda_3=1$.

因为 SDM 是 3 个部分的线性插值,因此不清楚除了 Jerlinek-Mercer 平滑之外,其他平滑技术如何能够用在

这个模型中.本文采用这个模型仅仅是作为参考系统与本文提出的 SBDM 模型进行比较.

3 实验与分析

3.1 数据集

本文在 6 个不同的 TREC(disk 2 &3)数据集上分别对我们提出的文摘模型进行了实验.这 6 个 TREC 数据集的统计属性见表 1.所有文档按以下方式统一处理:用 Porter 对 term 进行 stemming;去除 stopwords.查询选用 TRECdisk2 和 disk3 的 topics 202-250.

采用的主要评测指标是 11 点平均准确率.

Table 1 Statistics of TREC collection

表 1 TREC 数据集属性

Collection	Description	Average document length	Number of documents	Vocabulary
WSJ	Wall Street Journal (1990,1991,1992), Disk 2	313	74 520	126 446
SJM	San Jose Mercury News (1991), Disk3	237	90 257	146 529
AP	Associated Press (1988,1989,1990), Disks 2 and 3	261	158 240	194 799
PAT	U.S. Patents (1993)	2 985	6 711	110 029
FR	Federal Register (1988)	958	19 860	148 753
ZIFF	Articles from Computers	171	217 940	227 886

3.2 参数设定

在实验中,需要确定语言模型的平滑参数.本文用 AP 作为选择参数的训练集.WSJ,FR,SJM,PAT 和 ZIFF 用作测试集,用以测试在训练集 AP 上得到的参数是否在其他数据集上也同样能得到满意的结果.FR 数据集与其余的数据集相比,具有非常不同的特性.因此,在其他数据集上训练得到的参数很难在这个数据集上获得满意的结果,反之亦然.在 FR 数据集上需要单独的参数调整.在现阶段本文的工作中,参数选择采用穷举搜索法.本文在进行参数搜索时,从 100 开始,以步长 100 递增,通过实验发现,当到 4 000 左右时,性能开始下降.在 2 000 左右时,达到极值点,此时的性能最佳,如图 1 所示.

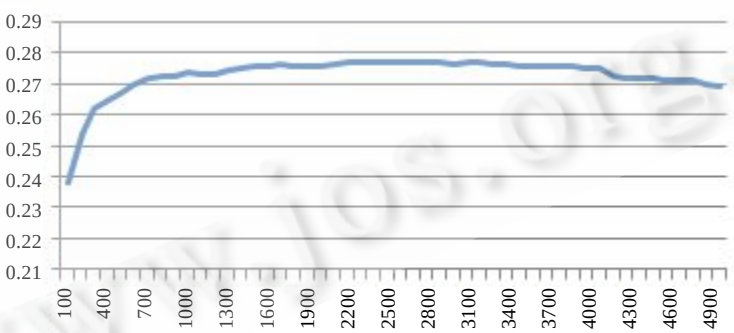

Fig.1 Parameter testing of AP

图 1 AP 数据集参数实验

3.3 实验设计

实验的第 1 部分验证不同的文摘方法对检索性能的影响,第 2 部分检验文摘模型的效能.

在第 2 部分实验中,第 1 组实验研究是否简单的文摘模型 SQL 可以用来对检索文档排序以获得与前人报告一样好的结果.在 AP 和 WSJ 数据集上,用两个平滑方法对文摘模型进行平滑,它们分别是贝叶斯平滑和插值平滑.通过参数设定获得的最优结果与参考系统,即基于标准文档模型的最优结果进行比较.第 2 组实验用本文提出的 SBDM 模型检验基于文摘的检索模型的功效.比较的对象是 SBDM 模型与使用标准文档的模型.此外,

我们还将验证本文提出的基于摘要的检索模型与上文提到的 SDM 模型相比,是否有明显的性能上的不同.

3.4 实验及结果分析

3.4.1 文摘对检索的影响

本文首先验证了不同的文摘抽去方法对检索的影响.实验过程是:对每个文档分别用文中提到的不同方法抽取文摘,再用这几种方法进行线形插值生成一个文摘.把生成的 4 种文摘带入 SQL 模型进行检索实验.实验结果见表 2.由于位置因子抽取文摘只抽取文档的前两句,无法单独构成文摘,因此,本实验没有考虑这种方法.但在对这几种方法做线性插值时,加入了这种方法.表 2 中,Document 表示标准文档.Luhn 是指重要词方法.Query 表示查询因子方法.Title 表示标题因子方法.Combine 是指几种文摘方法经过插值后生成的文摘.FR 数据集中的文档没有标题,所以 FR 实验结果 Title 一栏为空.

Table 2 Effect of summaries on IR (average precision)

表 2 文摘对检索的影响(平均准确率)

Collection	Document	Query	Title	Luhn	Combine
AP	0.274 0	0.158 8	0.135 2	0.121 7	0.165 0
WSJ	0.216 5	0.139 2	0.118 3	0.115 5	0.146 3
SJM	0.237 1	0.144 0	0.116 5	0.112 3	0.153 6
PAT	0.320 4	0.185 0	0.195 2	0.193 6	0.281 0
FR	0.152 6	0.124 5		0.078 5	0.136 0
ZIFF	0.180 9	0.054 0	0.052 2	0.040 1	0.058 5

通过表 2 可以看出,查询因子对检索的贡献最大.这主要归功于文摘抽取时对查询词的考虑.这种文摘方法区别于其他单纯的文档文摘方法,体现了面向检索的文摘的特点.

3.4.2 SQL 模型实验

这组实验的目的是比较基于文摘的检索模型 SQL 和基于文档的检索模型.实验结果显示,在表 3 中,AP+DM 表示用标准的文档检索模型(第 2 列和第 4 列是实验的两个参考系统的平均准确率).实验的过程是:首先,按照在第 1 节提到的式(1)和式(2)进行基于文档的检索,产生一个文档排序列表.然后进行基于文摘的检索模型,同样,产生一个排序列表.最后,用标准的 TREC 11 点平均准确率评价这两个列表.首先在 AP 数据集上训练 SQL 模型以确定平滑参数.通过调整平滑参数以获得最佳的参数设置.这组实验引入的平滑技术包括贝叶斯(Bayesian)平滑和插值平滑.实验发现,用贝叶斯平滑时,Dirichlet prior 设为 2 000,SQL 达到或几乎达到最佳性能.这个参数设定在基于文档检索模型也获得最佳性能.从表 3 和表 4 可以看出,在训练集 AP 上,基于文摘的检索模型 SQL 不如基于文档的检索.在其他测试集上,基于文摘的模型 SQL 同样不如基于文档的模型性能好.

Table 3 SQL vs. DM (11-point average precision)

表 3 SQL vs. DM (11 点平均率)

Recall	AP+DM	AP+SQL	WSJ DM	WSJ+SQL
0.00	0.660 9	0.614 8	0.611 7	0.516 2
0.10	0.524 8	0.421 3	0.464 2	0.359 8
0.20	0.446 5	0.311 6	0.381 4	0.279 8
0.30	0.391 7	0.237 5	0.303 5	0.186 2
0.40	0.331 2	0.186 2	0.245 9	0.132 8
0.50	0.283 8	0.143 5	0.185	0.106 2
0.60	0.219 6	0.100 8	0.159 7	0.080 2
0.70	0.175 9	0.044 5	0.104 9	0.026 2
0.80	0.103 8	0.015 7	0.077 8	0.015 5
0.90	0.060 1	0.002 6	0.051 4	0.012 4
1.00	0.026 0	0.002 0	0.031	0.010 8
Aver. Prec.	0.274 0	0.165 0	0.216 5	0.139 2

Table 4 SQL vs.DM (average precision)

表 4 SQL vs.DM (平均准确率)

Collection	DM	SQL
AP	0.274 0	0.165 0
WSJ	0.216 5	0.139 2
SJM	0.237 1	0.154 4
FR	0.152 6	0.136 0
ZIFF	0.180 9	0.052 2
PAT	0.320 4	0.281 0

总之,实验结果说明:基于文摘的模型 SQL 在训练集和测试集都不及基于文档的模型性能更优.我们用这些结果作为参考基线模型,与本文提出的第 2 个基于文摘的模型 SBDM 进行比较.

3.4.3 SBDM 模型实验

这组实验旨在评价基于文摘的 SBDM 模型和标准文档模型的优劣.通过在 AP 数据集上的实验,我们获得

SBDM 模型的最优参数设定.在图 2 中,把这个结果与标准文档模型进行了比较.图中的 DM 表示标准文档模型(公式(2)). x 轴表示平均准确率,y 轴表示召回率.

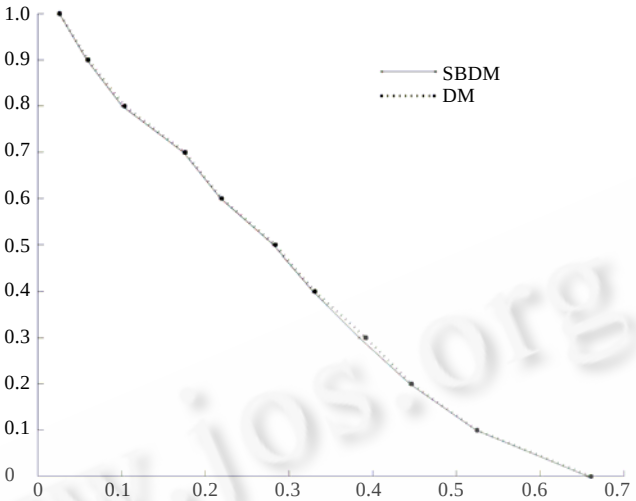

Fig.2 Compared SBDM with DM on AP set

图 2 AP 数据集 SBDM 与 DM 的比较

为了验证在 AP 数据集上得到的 SBDM 模型的最优参数是否在其他数据集上也有效,本文在 WSJ,FR,SJMN,PAT 和 SJM 数据集上分别做了实验,结果显示在表 5 中.通过实验发现:在所有的 5 个实验数据集上,基于文摘的检索模型 SBDM 明显好于基于文档的模型.例如在 FR 数据集上,平均准确率提高了 8.39%.

除了上面的实验,我们也把 SBDM 模型与信息检索的其他经典模型(向量空间模型(SVM),概率模型(Okapi),语言模型(LM))进行了比较.Okapi 模型采用文献[15,16]中的方法,并且没有相关反馈.因为 SBDM 也没有用任何反馈机制.表 6 中的结果表明:SBDM 无论在哪个数据集上都明显优于 SVM.与目前流行的语言模型(LM)相比,SBDM 也一致接近或优于 LM,见表 5.这里,DM 就是 LM,只是为了与作者提出的文摘模型比较,因此称其为文档模型.SBDM 与概率模型的代表 Okapi 相比,在 FR,SJM 和 PAT 集合上接近或优于 Okapi,而在其他两个数据集略低.我们分析这几个集合文档长度较短,文摘的精度不高,可能是造成这个差距的原因.但与目前研究的热点语言模型 LM 相比,SBDM 具有一致的优越性,而 Okapi 则不稳定,如在 PAT 和 SJM 数据集 Okapi 不如传统 LM 模型的情况下,SBDM 则优于 LM 模型.

Table 5 SBDM vs. DM (average precision)

表 5 SBDM vs. DM (平均准确率)

Collection	DM	SBDM	Increase
AP	0.274 0	0.272 5	–0.55
SJM	0.237 1	0.241 0	1.64
WSJ	0.216 5	0.219 9	1.57
FR	0.152 6	0.165 4	8.39
PAT	0.320 4	0.334 4	4.37
ZIFF	0.180 9	0.190 7	5.42

Table 6 Compared SBDM with others models (LM,VSM,Okapi) (average precision)

表 6 SBDM 与其他模型的比较(LM,VSM,Okapi) (平均准确率)

Collection	LM	VSM	Okapi	SBDM
AP	0.274 0	0.263 6	0.284 2	0.272 5
SJM	0.237 1	0.200 6	0.224 3	0.241 0
WSJ	0.216 5	0.191 9	0.231 7	0.219 9
FR	0.152 6	0.133 2	0.168 8	0.165 4
PAT	0.320 4	0.235 9	0.306 8	0.334 4
ZIFF	0.180 9	0.098 2	0.212 7	0.190 7

同时,本文还对 SBDM 模型与前文提到的 SDM(公式(6))模型进行了对比实验.在实验的 3 个数据集 AP, WSJ,ZIFF 上,都是λ_1=0.1,且λ_2=0.3 时性能达到最优.从表 7 中可以看出,SBDM 模型明显优于 SDM 模型.主要因为 SDM 模型只能采用插值平滑,而 SBDM 可以更灵活地选择多种平滑方法.

Table 7 SBDM vs. SDM (average precision)

表 7 SBDM vs. SDM(平均准确率)

Collection	λ_1	λ_2	SDM	SBDM	%chg
AP	0.1	0.3	0.258 5	0.272 5	5.4
WSJ	0.1	0.3	0.206 6	0.219 9	6.4
ZIFF	0.1	0.3	0.179 5	0.190 7	8.7

4 结 论

本文提出了两个语言模型框架下的文摘检索模型.一个模型是以文摘模型替代文档模型直接进行文本检索,另一个模型是用文摘模型平滑文档模型,并在 TREC 数据集上对提出的模型进行了评测.通过实验可以得出以下结论:首先,语言模型框架下的文摘检索模型是可行的.本文提出的两个模型中,SBDM 模型至少获得了与前人实验同样好的结果,甚至有时要好于前人的结果.其次,基于文摘的模型要比标准的基于文档的模型更有效.例如,SBDM 模型的性能一致接近或优于传统的标准文档模型,即使在 Okapi 不如传统查询相似模型的情况下,SBDM 模型也表现优异.第三,对于信息检索来说,用文摘模型平滑文档模型比直接用文摘模型替代文档模型更有效.此外,本文提出的模型引入了不同的平滑方法,并且在一个数据集上获得的优化参数能够适用于其他数据集.

References:

[1] Callan JP. Passage-Level evidence in document retrieval. In: Proc. of the 17th ACM SIGIR. New York: Springer-Verlag, 1994. 302−310.

[2] Luhn HP. The automatic creation of literature abstracts. IBM Journal of Research and Development, 1985,2(2):159−165.

[3] Edmundson HP. New methods in automatic abstracting. Journal of the Association for Computing Machinery, 1969,16(2):264−285.

[4] Jacobs PS, Rau LF. Scisor: Extracting information from on-line news. Communications of the ACM, 1990,33(11):88−97.

[5] McKeown K, Radev DR. Generating summaries of multiple news articles. In: Proc. of the 18th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. New York: ACM, 1995. 74−82.

[6] Tombros A, Sanderson M. Advantages of query biased summaries in information retrieval. In: Proc. of the 21st ACM SIGIR. Melbourne: ACM, 1998. 2−10.

[7] Adesina AM, Jones AM. Applying summarization techniques for term selection in relevance feedback. In: Proc. of the 24th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. New York: ACM, 2001. 1−9.

[8] Strzalkowski T, Wang J, Wise B. Summarization-Based query expansion in information retrieval. In: Proc. of the 17th Int'l Conf. on Computational Linguistics. Montreal: ACL, 1998. 1258−1264.

[9] Tombros A. Reflecting user information needs through query biased summaries. Technical Report, TR-1997-35, Department of Computing Science, University of Glasgow, 1997.

[10] Ponte J, Croft W. A language modeling approach to information retrieval. In: Proc. of the 21st ACM Conf. on Research and Development in Information Retrieval (SIGIR'98). New York: ACM, 1998. 275−281.

[11] Miller D, Leek T, Schwartz R. A hidden Markov model information retrieval system. In: Proc. of the SIGIR 1999. New York: ACM, 1999. 214−221.

[12] Zhai C, Lafferty J. Two-Stage language models for information retrieval. In: Proc. of the 25th ACM SIGIR Conf. New York: ACM, 2002. 49−56.

[13] Lavrenko V, Croft WB. Relevance based language models. In: Proc. of the 24th ACM SIGIR Conf. New York: ACM, 2001. 120−127.

[14] Zhang Y, Callan J, Minka T. Novelty and redundancy detection in adaptive filtering. In: Proc. of the ACM SIGIR 2002. New York: ACM, 2002. 81−88.

[15] Jones KS, Walker S, Robertson S. A probabilistic model of information retrieval: Development and comparative experiments-part 1. Information Processing and Management, 2000,36(6):779−808.

[16] Jones KS, Walker S, Robertson S. A probabilistic model of information retrieval: Development and comparative experiments-part 2. Information Proc. and Management, 2000,36(6):809−840.

李卫疆(1969－),男,四川雅安人,博士,主要研究领域为信息检索,网络信息处理.

臧文茂(1982－),男,硕士,主要研究领域为信息检索.

赵铁军(1962－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为自然语言处理,机器翻译,人工智能.