第 33 卷 第 4 期 计 算 机 工 程 2007 年 2 月
Vol.33 No.4 Computer Engineering February 2007
·软件技术与数据库· 文章编号：1000—3428(2007)04—0049—03 文献标识码：A 中图分类号：TP311

信息检索中语言模型的研究

楼炉群，牛军钰
(复旦大学计算机科学与工程系，上海 200433)

摘 要：介绍了最新被应用于信息检索领域的模型——语言模型。论述了构造应用于信息检索语言模型的 3 个步骤，介绍了这种语言模型的排序方法、反馈和与其它因素结合的方法，以及其在检索领域的应用效果，提出了语言模型在信息检索中的发展方向。
关键词：语言模型；信息检索；平滑；反馈

Research of Language Model in Information Retrieval

LOU Luqun, NIU Junyu
(Department of Computer Science and Engineering, Fudan University, Shanghai 200433)

【Abstract】 This paper focuses on language model——a new kind of model applied in information retrieval. It begins with an introduction to the way of building up such a model through three stages. Some key aspects concerning this model like sort methods, feedback and combination with other factors are represented. And some facts on the application of language model in information retrieval are put forward to prove its efficiency. There is the prospect on its application in IR.
【Key words】 Language model; Information retrieval; Smoothness; Feedback

1998 年 Ponte 和 Croft 将语言模型(Language Model，LM)应用到信息检索(Information Retrieval，IR)中[1]，现在语言模型已经成为信息检索的主要研究领域。

1 语言模型

1.1 语言模型简介

语言模型在应用于 IR 之前，已经成功地应用于语音识别、机器翻译以及中文分词等领域。

使用语言模型进行信息检索的框架与一般的信息检索相同，如图 1 所示。使用语言模型的框架还可以用 Shannon 的 Source-Channel Framework 来解释。

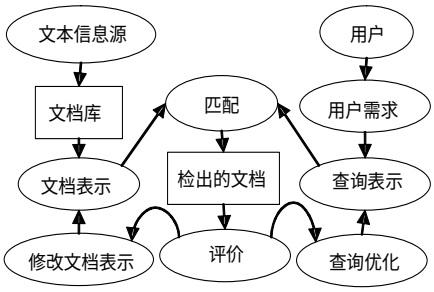

图 1 语言模型应用于信息检索的一般模型表示

如图 1 所示，信息检索主要涉及文档表示、查询表示、匹配过程(排序)、反馈这 4 大模块，许多检索模型的差别主要就表现在这 4 个模块上，下面将分别介绍。

首先，使用语言模型的信息检索，对于文档的表示 D 或查询的表示 Q，可以使用图 2 所示的建模的过程来进行。

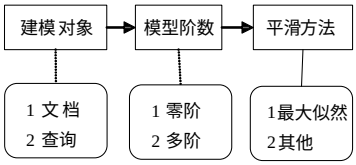

图 2 语言模型建模全过程

建立文档或查询模型，首先需要选择建模对象，然后对选择好的对象建立模型的基本框架，最后选择合适的方法进行平滑(用来调整模型，使模型能尽可能表示文档或查询的方法)优化检索。

1.2 选择建模对象

因为 LM 在 IR 中可对文档和查询建模，所以建模可分为 3 种情况：只对文档建模，只对查询建模以及对二者都建模。对于文档建模生成的模型称之为文档模型，用符号 M_D 表示。对于查询建模后生成的模型称之为查询模型，用符号 M_Q 表示。可以用 M 来统称模型。

建模对象与排序方法联系非常紧密，如果选择了建模对象，也就等同于选择了排序方法，反之亦然。正因为有如此紧密的关系，所以关于如何选择建模对象，也就等同于如何选择排序方法。

1.3 选择模型的阶数

模型可以根据马尔科夫链的阶数分为一元语言模型和多元语言模型。

一元语言模型(unigram language model)假设词与词之间是相互独立的，一个词出现的概率与这个词前面的词没有存在必然联系。这是最简单的语言模型。

多元语言模型(N-gram language model，其中 N>=2)假设词与词之间是相互关联的，一个词出现的概率与这个词前面的词存在一定的关联。根据与前面词的个数的多少，把多元语言模型分为二元语言模型、三元语言模型等。

对于一个句子 $s = w_1 w_2 ... w_n$ （w_i 代表某个词），在语言模型 M 中 S 出现概率 P 则用一元和多元模型可以分别表示为式

基金项目：国家自然科学基金资助项目(60305006)
作者简介：楼炉群(1982－)，男，硕士生，主研方向：Web 检索；牛军钰，副教授
收稿日期：2006-03-09 **E-mail：**louluqun@hotmail.com

(1)和式(2)。

一元语言模型中，根据链式法则可以表示为

$$P(s \mid M) = \prod_{w_i \in s} P(w_i \mid M) \quad (1)$$

其中 $P(w_i \mid M)$ 则指的是在模型 M 中存在词 w_i 的概率。

多元语言模型中，根据链式法则可以表示为

$$P(s \mid M) = \prod_{w_i \in s} P(w_i \mid w_{i-1}, w_{i-2}, ..., w_{i-k+1}, M) \quad (2)$$

其中参数 k 为多元模型中某个词的出现与前 k 个词相关，其余参数与式(1)同。

Song 和 Croft[1]指出，把一元语言模型和二元语言模型混合后的效果比只使用一元语言模型则好 8%左右。不过，Victor Lavrenko 中指出，Song 和 Croft 使用的多元模型得到的效果并不是一直比只用一元语言模型好。

同样，David R.H.Miller 指出一元语言模型和二元语言模型混合后得到的效果也要好于一元语言模型。

从目前的应用来看，大部分语言模型采用一元模型。IR 中倾向使用一元语言模型要原因：一方面，词与词之间的先后次序在 IR 时并没有很大价值；另一方面，多元语言模型将涉及更多参数，为了估计这些参数，需要更多的数据，因而使得计算变得更复杂，也可能使结果变得更难预料。

1.4 选择平滑方法

在 1.3 节中，把计算一个句子 S 在 M 中的概率分解为计算每个词 w 在 M 中的概率。本节将讲述计算 $P(w \mid M)$ 的主要方法。

最简单方法是最大似然估计，以文档模型为例，词 w 在模型 M 中出现的概率可估计为

$$P(w \mid M_D) = \frac{c(w, D)}{|D|} \quad (3)$$

其中 $c(w, D)$ 为词 W 出现在文档 D 的次数；$|D|$ 为文档 D 的长度。

但是最大似然估计方法有其适用范围，不能应用于稀疏事件，而计算词在文档或查询中概率问题往往是稀疏事件。稀疏事件会导致零概率问题，如果多个文档同时发生了零概率问题，那么无法区分这些文档。

零概率问题的解决方法就是平滑(smoothing)。平滑主要通过对文中未出现的词赋以一定的概率，同时可能调整文中已出现词的概率。

现在应用于 IR 中 LM 使用的很多平滑方法都是从应用于语音识别的 LM 里面借鉴过来的。最常用的方法一般为播入方法中的 Jelinek-Mercer，Dirichlet，absolute discounting，二阶段平滑方法以及其它模型中的翻译模型以及 PLSI 方法，也有少数采用了 Good-Turing 方法，几乎没有人使用 backoff 方法。

研究表明[3]，较长类型的查询对于平滑的参数更加敏感，对于 Jelinek-Mercer 平滑、Dirichlet 平滑和绝对折扣(absolute discounting)平滑三者来说，Jelinek-Mercer 平滑更加适合长度较长的查询，Dirichlet 平滑则比较适合长度较短的查询。

平滑方法的功能与传统概率模型中 tf-idf 的功能相类似，可以说具有简单平滑方法的 LM 实现了传统概率模型中的 tf-idf 方法以及文档长度的归一化[3]。平滑具有提高模型估计的精确率和避免一些文档的无用信息二重作用。

平滑也是传统概率模型与 LM 的区别，可以通过平滑让模型能够更加真实地代替文档或查询。

1.5 排序

通过 1.2 节~1.4 节计算所得到的 $P(s \mid M)$ 可以进行排序，其中 s 表示句子。

IR 中最关键的部分就是排序。排序方法几乎决定整个搜索引擎的性能。根据前面建模对象的分排序方法，同样将排序方法分为 3 类：

(1)查询相似(query-likelihood)：主要通过计算 $P(Q \mid M_D)$ 进行排序，即通过计算文档模型能在多大程度上产生查询的概率来排序。

(2) 文档相似(document-likelihood)：主要通过计算 $P(D \mid M_Q)$ 进行排序，即通过计算查询模型能在多大程序上产生文档的概率来排序。

(3) 模型比较(model comparison)：主要通过计算 $D(M_Q \| M_D)$ 进行排序，即通过计算查询模型与文档模型的相似性进行排序。其中对于模型的相似性计算比较有名方法是 KL– divergence(Kullback-Leibler Divergence)[5]。其计算公式如下：

$$D(M_Q \| M_D) = -\sum_w P(w \mid Q) \log P(w \| D) \quad (4)$$

下面将 3 种排序方法作一个比较：

从表 1 可以看出，文档相似不能单独应用于 IR。因为不同长度文档，文档相似的概率不能比较。而模型比较则结合了查询相似和文档相似的部分优点，是目前使用效果最好的排序方法。目前 CMU 的 Lemur 工具就支持模型比较的方法进行排序。

表 1 各种排序方面的优缺的比较

	查询相似	文档相似	模型比较
缺点	不直接支持查询反馈	不同长度的文档的概率不能相互比较	方法较为复杂
	不直接支持结构化查询	内容较少且有高频词的文档排序易靠前	
优点	比较简单	直接支持查询反馈	支持查询反馈

值得一提的是在 2001 年 Lafferty 和 Zhai 提出了一个基于 Bayesian 决策理论的风险最小化框架[4]，该框架把查询相似、文档相似以及模型比较 3 种排序方法有机统一起来。

2 反馈及其它

2.1 反馈简介

传统的概率模型与 LM 在如何反馈时存在不少的区别。

传统的概率模型通过加入查询返回后的文档列表中的某些与查询词相邻的词来进行查询反馈，但是这种做法只能依赖人们经验(与查询词相邻的词比其它词更有可能是查询词)，缺乏必要的理论依据。

LM 则可以通过对反馈回来的文档进行建模 M_F，然后与原有查询模型 M_Q 或文档模型 M_D 相结合，生成新的查询模型 M_Q' 或文档模型 M_D'，再查询、排序，对反馈回来的文档建模。经过多次循环，使最终的查询模型或文档模型趋于稳定。所以 LM 的反馈可以与查询模型或文档模型合并为一个模型。例如查询模型的反馈

$$M_Q' = (1-\alpha) M_Q + \alpha M_F \quad (5)$$

其中 α 为参数。

M_F 的计算可以通过两种方法进行：生成式结合模型(Generative Mixture Model)和最小化经验分歧(Empirical Divergence Minization)。

2.2 反馈分类

可以按建模对象的不同，对反馈进行分类，如表 2 所示。

表 2　LM 反馈分类

	查询模型	文档模型	查询+文档
不扩充查询	无	混合模型 依赖模型 密度分配模型 最大似然模型	与文档相似
扩充查询	基于模型的反馈	相关模型	与查询/文档相似

基于查询模型的反馈一般都会改变原始的查询概率分布；除非仅改变查询词相应权重来进行反馈。当然，这种改变查询词权重的方法效果不明显。而文档模型的反馈一般也会改变的原始文档概率分布，基于文档模型的反馈在改变查询时需要建立一个相关模型。

2.3 反馈分析

相关模型和基于反馈的查询模型在性能上已经比简单的 LM 有了明显的提高。

基于反馈的语言模型在 3 个很有代表性的语料(AP88-89，TREC8，WEB)中，其精确度都比传统的 tf-idf 加反馈的精确度高[5]，而且在大部分情况下语言模型加上反馈后比没有加上反馈时平均提高 10%的精确度。

Ramesh 提出的 4 个反馈模型中有 3 个模型比与普通查询相似的方法在 MRR 上至少有 35%的提高。

2.4 结合其它因素

前面的排序方法假定了文档出现概率是相同的，然而事实上文档出现概率存在区别。所以在 1.4 节介绍排序方法的基础上，乘以文档出现的概率 $P(D)$ 来进行排序。例如对于文档相似排序方法,可以按照 $P(D)*P(D|M_Q)$ 进行排序。文档的出现概率 $P(D)$ 可以通过其它因素进行粗略的估计。

LM 和传统概率模型在与某些因素的结合方式存在区别。与反馈类似，LM 能够将这些因素放入一个模型中。这些因素根据与文档内容有无相关性分成两类。

2.5 与文档内容之外因素的结合

这些因素包括：链接结构，文档长度，URL 深度。

(1)链接结构。最简单使用链接结构的方法就是只计算链入链接的个数。

$$P_{inlink}(D) = C \times inlinkCount(D) \tag{6}$$

其中 C 为参数，$P_{inlink}(D)$ 为文档 D 在 inlink 因素作用下出现的概率。

(2)文档长度

$$P_{doclen}(D) = C \times doclen(D) \tag{7}$$

其中 C 为参数，$doclen(D)$ 为文档 D 长度，$P_{doclen}(D)$ 为文档 D 在文档长度因素下出现的概率。

(3)URL 深度(URL Depth)

可以根据 URL 深度将 URL 分为 4 种：根(root)，子根(subroot)，路径(path)，文件(file)。

$$P_{url}(D) = P(EP \mid URL_{type}(D) = t_i) = \frac{c(EP, t_i)}{c(t_i)} \tag{8}$$

其中 $c(EP, t_i)$ 为文档 D 为属于主页(entrypage)且 URL 类型为 t_i 的文档数目，$c(t_i)$ 为 URL 类型为 t_i 的文档的数目，$P_{url}(D)$ 为文档 D 在 URL 因素影响下出现的概率。

2.6 与文档内容相关因素的结合

(1)文档组成

文档中标题和正文，以及锚文本在重要性上肯定存在区别。因此，文档的不同组成部分赋以不同的权重。

语言模型可以通过定义一个混合模型，与文档的多种表现方式相结合，可以通过

$$P(Q \mid M_D) = \prod_{w \in Q} [(1 - \lambda - \mu) P(w_i \mid C) + \lambda P_{Content}(w_i \mid D) + \mu P_{Anchor}(w_i \mid D)] \tag{9}$$

其中 λ, μ 为参数；

(2)文档类别信息

将所有相似的文档进行聚簇，再把簇的信息加入到簇中的单个文档的文档模型中来提高检索的性能。

Oren 在 2004 年详细介绍了如何建立一个建立簇的算法模板，并且阐述了簇的二重作用——平滑单个文档以及总结语料的某个方面内容。

2.7 结合分析

当文档内容与其它因素，例如链接、URL 和锚文本结合时，文档内容与 URL 结合得到了最好的结果。不过，加入链接也会有所提高。

对相似文档进行聚簇，进而利用簇的信息，是一个很好的方法。可以通过对个人信息或爱好进行聚簇，从而可以进行个性化的搜索。

3 应用效果

3.1 Web 任务

1998 年，Ponte 和 Croft 指出其采用语言模型的效果比标准的 tf-idf 方法在两个不同的语料集及查询中都有很显著的提高。

1998 年，Miller 和 Leek 的 BBN at TREC7[2]中指出，他们的语言模型实现在 trec-6 和 trec-7 的 adhoc 任务中比 tf-idf 检索的结果要好得多。

3.2 基因任务

2004 年 Fujita 公司采用 KL-dir 加上伪反馈取得了 MAP 为 0.356 7，优于采用 BM25 算法。

4 结论

语言模型与传统概率模型在参数估计、反馈、与其它因素的结合等方面具有不少区别。现在结合个人信息的搜索越来越受到青睐，因而有必要使语言模型对个人的兴趣和爱好进行跟踪并且建模，使得查询和排序都可以根据语言模型来提高系统的精确率。另外，有必要使语言模型的参数调整更加自动化，让专业人员能把更多的精力放在其它方面。最后，语言模型应该能应用于检索更加复杂的查询。

参考文献

1 Ponte J M, Croft W B. A Language Modeling Approach to Information Retrieval[C]//Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval, 1998: 275-281.

2 Miller D R H, Leek T, Schwartz R M. BBN at TREC7: Using Hidden Markov Models for Information Retrieval[C]//Proceedings of the 7th Text Retrieval Conference, 1998: 80-89.

3 Zhai Chengxiang, Lafferty J. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval[C]// Proceedings of the ACM-SIGIR, 2001: 334-342.

4 Lafferty J, Zhai Chengxiang. Document Language Models, Query Models, and Risk Minimization for Information Retrieval[C]// Proceedings of the ACM SIGIR, 2001: 111-119.

5 Zhai Chengxiang, Lafferty J. Model-based Feedback in the Language Modeling Approach to Information Retrieval[C]//Proceedings of the 10th International Conference on Information and Knowledge Management, 2001: 403-410.