第42卷 第12期 西 安 交 通 大 学 学 报 Vol. 42 №12
2008年12月 JOURNAL OF XI′AN JIAOTONG UNIVERSITY Dec. 2008

一种编辑距离算法及其在网页搜索中的应用

薛晔伟，沈钧毅，张云
（西安交通大学电子与信息工程学院，710049，西安）

摘要：针对传统方法不能很好地处理网页中简短域与用户查询之间的相关性排序问题，提出一种基于改进的编辑距离排序算法.将以词为单位的用户查询和简短网页域通过匹配编码转化为2个字符串，再利用改进的编辑距离计算2个字符串之间的相似性.由于在用户查询与待比较的简短网页域之间引入了查询词分布的位置、顺序和距离等，以及含有查询词修饰关系的重要信息，所以编码字符串之间的相似程度可以衡量对应的查询与简短网页域之间的相关性.经大规模真实搜索引擎实验表明，该算法较之传统的相关性排序算法，可以显著地提高网页搜索中的简短网页域相关性排序性能，尤其适用于简短域与用户查询之间的相关性比较.

关键词：网页搜索；相关性排序；编辑距离；字符串匹配
中图分类号：TP391 **文献标识码**：A **文章编号**：0253-987X(2008)12-1450-05

Modified Edit Distance Algorithm and Its Application in Web Search

XUE Yewei, SHEN Junyi, ZHANG Yun
(School of Electronics and Information Engineering, Xi′an Jiaotong University, Xi′an 710049, China)

Abstract: Focusing on the problem that the traditional methods cannot perform well on the short web page fields, a modified edit distance algorithm, referred as MED, is proposed. The proposed algorithm encodes the user query and the short web fields into two strings according to the word match, and then the MED is used to calculate the similarity between the two strings. Because the 'position', 'order', and 'distance' information that is very important in expressing the modification relationship between the query words are considered, the similarity between the encoding strings can be used to measure the relevance between the corresponding query and short field. Experimental results on large scale search engine data show that the proposed algorithm can significantly outperform the traditional algorithms for relevance ranking on short web fields, especially for very short fields.

Keywords: web search; relevance ranking; edit distance; string match

网页相关性排序是搜索引擎的核心技术，它决定了用户看到的结果网页的先后顺序.因此，网页相关性排序方法的好坏直接影响用户对搜索引擎的印象.对于一个网页的排序搜索引擎可以参考正文、标题、URL、锚文本等数量丰富的信息，这些信息又被称作域.随着搜索引擎技术的发展，利用域信息来改善整个网页的排序已成为一种共识[1-2].大多数相关性排序算法均采用词频来计算网页与查询之间的相关性[3]，它们常用到TF和DF信息，前者表示用户查询词在候选文档中出现的次数，后者代表查询词在整个文档集合中出现的次数[4].近年来，临近信息模型[5]在前者的基础上做了有限的改进，加入了查询词在文档中的距离信息，但是评价基础仍然是词频信息.

目前，最为常用的网页域就是正文、标题、锚文本和URL.传统的词频排序模型在正文域上取得了

收稿日期：2008-04-15. 作者简介：薛晔伟(1980—)，男，博士生；沈钧毅(联系人)，男，教授，博士生导师. 基金项目：国家自然科学基金资助项目(60673087).

巨大的成功，但是由于标题等其他 3 个域的长度有限，即使存在查询词，也很少有机会重复，因此这些域上基本不存在词频信息. 这恰恰是采用传统网页排序算法计算效果不理想的根本原因.

针对这个问题，本文提出一种改进的编辑距离算法，记作 MED. 该算法并未考察词频信息，而是利用查询词之间的顺序，以及查询词在文档中出现的位置和距离等信息，来表达查询词与文档之间的相关性.

1　问题描述

用户查询是用户提交给搜索引擎的、表达需求信息的字符串，通常由数个查询词构成[6]. 在无法输入完整的自然语言语句的情况下，用户会仔细地选择查询词，并通过查询词的顺序来表达进一步的修饰和限定关系. 相对于正文等篇幅较长的域，标题、锚文本和 URL 等非常简短的域会受到篇幅的限制，并不存在查询词的重复现象，因此不能够以查询词出现多少为相关性计量的标准. 查询词在这些简短域中分布的位置、顺序和距离等信息，才是判断查询与简短网页域之间相关性的关键因素. 表 1 列出了能够用来表达相关性的一些信息以及应用情况.

表 1　各种相关性信息的应用情况

	长篇有效(词频)		简短有效(非词频)		
	TF	DF	距离	位置	顺序
VSM 模型[3]	no	no	no	no	no
BM25 模型	yes	yes	no	no	no
临近模型	yes	yes	no	no	yes

常用的以 BM25[4] 为代表的概率模型，仅使用了 TF 和 DF 这 2 个词频信息，因此它们在正文等长篇网页域中表现良好，而在简短网页域中表现不好. 临近信息模型引入了查询词距离信息，它比 BM25 模型有所改进[5]. 本文提出的 MED 算法，可以充分利用上述的非词频信息，即位置、顺序和距离信息.

2　MED 算法

编辑距离[7-8]是一种著名的字符串相似性衡量方法，它通过计算 2 个字符串之间转化所需要的最少编辑操作的数量，来衡量 2 个字符串之间的相似性. 所需的操作数越少，2 个字符串相关性越高. 本文通过对其进行改进，将以词为单位的查询与网页域之间的相关性计算，转化为以编码字符为单位的查询字符串与域字符串之间的相似性计算，并应用于搜索引擎的网页相关性排序.

2.1　编码过程

编码过程将以词为单位的用户查询和简短网页域，编码为相应的以字符为单位的查询编码字符串和域编码字符串. 首先，本文定义了一个操作过程的字符空间，来代表查询词序列位置的字符集合 $S=\{a,b,\cdots,j\}$ 和与之互补的集合 $\overline{S}=\{\bar{a},\bar{b},\cdots,\bar{j}\}$. 字符 $a,b,\cdots,j$ 代表查询中第 1、第 2，及第 10 个查询词(绝大部分的查询都非常短，如有必要考虑其他位置，则通过简单的扩充字符集合就可以完成). 另外，引入 2 个代表特殊状态的字符 φ 和 ε，前者代表网页域中的非查询词，后者代表了一个空的状态.

对于查询，用字符 $x\in\{a,b,\cdots,j\}\cup\{\bar{a},\bar{b},\cdots,\bar{j}\}$ 表示查询中任意一个查询词的序列位置. 如果某个查询词在相应的域中没有出现，则用 $\overline{S}=\{\bar{a},\bar{b},\cdots,\bar{j}\}$ 中的字符取代 $S=\{a,b,\cdots,j\}$ 中相应字符. 这样，通过 $\{a,b,\cdots,j\}\cup\{\bar{a},\bar{b},\cdots,\bar{j}\}$ 中的字符，就可表示出查询中的任意一个查询词的序列位置，判断该查询词是否在网页域中出现，这 2 个是非常重要的相关性信息.

对于网页域的编码，用 $y\in\{a,b,\cdots,j,\varphi\}$ 表示域中任意一个词对应于查询的序列位置信息，其中 φ 代表网页域中包含的非查询词. 这样，字符集可以表示网页域中所包含的查询词和非查询词这 2 种不同的重要相关性信息. 编码过程描述如图 1 所示.

输入：查询 $Q=q_1,q_2,\cdots,q_m$ 和域 $T=t_1,t_2,\cdots,t_n$.
输出：查询编码字符串 x^m 和域编码字符串 y^n.
1. $x^m=\varepsilon;y^n=\varepsilon;$
2. for $i=1$ to m
3. if $(q_i$ in $T)$ $x_i=S_i;$
4. else $x_i=\overline{S}_i;$
5. for $j=1$ to n
6. if $(t_i$ in $Q)$ $y_i=x_k$ where $(t_i==q_k)$
7. else $y_i=\varphi;$
8. return (x^m,y^n)

图 1　编码过程算法

表 2 展示了 2 个编码的具体例子.

表 2　编码过程举例

用户查询	网页域	编码
machine learning	journal of machine learning	$(ab,\varphi\varphi ab)$
	learning information	$(\bar{a}b,b\varphi)$

经过编码，将由单词构成的查询和网页域，转化

为代表序列位置信息的、完全由有限个字符构成的2个编码字符串. 查询词之间的位置信息和查询词在网页域中是否出现、出现的位置等信息,通过编码被包含在这2个编码字符串中.

2.2 MED 算法

传统的编辑距离算法包含3个操作:替换、插入和删除. 本文针对网页搜索排序的特殊性,对编辑距离算法做如下改进.

(1)放弃替换操作,代之以跳过操作. 跳过操作代表当前待比较的2个字符完全相同,即这个位置上的查询词和域中的词相同,跳过当前位置,不做任何惩罚,接着比较下一个.

(2)通过插入操作,考察来自不同字符集合 S 和 φ 的操作对象,以实现区分插入查询词和插入非查询词2种不同类型的操作.

(3)用字符集合 S 代表查询词在网页域中出现,用字符集合的互补集合 $\overline{S}$ 代表查询词未被该网页域所包含,这样可实现区分查询词是否被网页域包含这2种状态. 通过删除操作,考察代表删除对象的不同字符,以表述彻底删除和临时转移位置2种不同的操作.

(4)引入字符 ε 代表空位置.

改进后的编辑操作如表3所示.

表3 改进的编辑距离操作

操作	符号	作用
跳过	$\langle x_i,y_j\rangle$	如果 $x_i=y_j$,跳过当前比较位置.
插入	$\langle \varepsilon_i,y_j\rangle$	代表特征词的字符 y_j 插入查询编码串的第 i 个位置.
删除	$\langle x_i,\varepsilon\rangle$	删除查询编码串中位置 i 上的字符 x_i.

下面是对改进原因的进一步解释.

(1)对于插入操作$\langle \varepsilon_i,y_j\rangle$,若 $y_j\in\{a,b,\cdots,j\}$,则代表插入操作在查询编码字符串中,插入的是一个查询中原本就存在的词(即查询词);若 $y=\varphi$,即$\langle \varepsilon_i,\varphi\rangle$,则代表插入操作在查询编码字符串中,插入的是一个在查询中原本就不存在的词.

改进的原因是:在网页搜索中,通常后者要付出更高的代价,因为它提供了多余的无关信息.

(2)对于删除操作$\langle x_i,\varepsilon\rangle$,若 $x\in\{a,b,\cdots,j\}$,则代表删除操作在当前位置删除了这个字符,但该字符会在域的其他地方出现,字符所代表的词仍然会在网页域中出现,是一个临时性的删除;若 $x\in\overline{S}=\{\bar{a},\bar{b},\cdots,\bar{j}\}$,则代表所删除的查询词在域的其他位置上并未出现,是一个彻底删除.

改进的原因是:通常搜索技术对于后者会给予非常严厉的处罚,因为这会导致重大的信息缺失.

(3)经过改进的编辑距离算法,依然包括3种操作:跳过、插入和删除,但是要注意的是,跳过操作代价为0,无需调整,而插入操作和删除操作都有2种不同的状态以及相应的代价,这4个代价就是在使用MED时需要进行调节和设置的参数(在实际的使用中,可以将一个参数设为1,然后调节其余3个参数,使之为1的倍数,所以实际需要调节的参数是3个).

通过这些改进的编辑操作和编码字符串,可以很方便地描述编码前查询词在网页域中的距离和顺序信息. 例如,通过查询词编码字符之间的其他字符数量(也就是插入操作的个数),就可以表示2个查询词之间的距离. 这样,通过编码过程和改进的编辑操作,MED可以全面地利用表1中提到的所有非词频信息.

2.3 动态规划求解过程

通常可以用动态规划的方法对编辑距离进行求解[9],将计算复杂度降低到多项式级别,使其具备实际应用能力. 给定查询 Q 和简短网页域 T,它们分别被编码为 x^m 和 y^n. 对于它们之间的相似性,可以用下述动态规划公式求解,即

$$
\begin{aligned}
M_{i,j}&=\min\begin{cases}M_{i-1,j-1}+c_{\boldsymbol{M}}(x_i,y_j)\\ I_{i-1,j-1}+c_{\boldsymbol{M}}(x_i,y_j)\\ D_{i-1,j-1}+c_{\boldsymbol{M}}(x_i,y_j)\end{cases}\\
D_{i,j}&=\min\begin{cases}M_{i-1,j}+c_{\boldsymbol{D}}(x_i,\varepsilon)\\ I_{i-1,j}+c_{\boldsymbol{D}}(x_i,\varepsilon)\\ D_{i-1,j}+c_{\boldsymbol{D}}(x_i,\varepsilon)\end{cases}\\
I_{i,j}&=\min\begin{cases}M_{i,j-1}+c_{\boldsymbol{I}}(\varepsilon_i,y_j)\\ I_{i,j-1}+c_{\boldsymbol{I}}(\varepsilon_i,y_j)\\ D_{i,j-1}+c_{\boldsymbol{I}}(\varepsilon_i,y_j)\end{cases}\\
\mathrm{Dist}(x^m,y^n)&=\min(M_{m,n},D_{m,n},I_{m,n})
\end{aligned}
\tag{1}
$$

式中:矩阵 $\boldsymbol{M}$ 代表跳过操作;矩阵 $\boldsymbol{D}$ 和 $\boldsymbol{I}$ 分别代表删除和插入操作;$c_{\boldsymbol{M}}(x_i,y_j)$ 代表跳过操作的代价,它在 x_i 和 y_j 相同时为0,反之则为无穷大,代表永远不可能进行这样的操作;$c_{\boldsymbol{D}}(x_i,\varepsilon)$ 对应删除操作$\langle x_i,\varepsilon\rangle$,代表将查询编码字符串第 i 个位置上的字符 x_i 删除的代价;$c_{\boldsymbol{I}}(\varepsilon_i,y_j)$ 对应插入操作$\langle \varepsilon_i,y_j\rangle$,代表使用域编码字符中第 j 个字符 y_j 插入查询编码字符串的第 i 个位置的操作代价. 为了减少参数,便于计算,上述的所有操作代价应与 i 和 j 无关,即

与具体操作发生的位置无关，而仅仅是 2.2 节中描述的 4 种操作代价.

图 2 简单描述了基于动态规划的 MED 的计算过程. 这个过程在每一步的选择中都选取使当前操作代价最小的操作，依次积累到最后，就得到一条总操作代价最小的路径，以及这条路径所产生的操作代价，即 2 个字符串之间改进的编辑距离.

输入：2个编码字符串x^m和y^n,分别代表查询和域.

输出：x^m和y^n之间的相似度，代表查询和域之间的相关度.

1. $M_{0,0}=1; D_{0,0}=0; I_{0,0}=0$
2. for $i=0$ to m
3. for $j=0$ to n
4. if $(i>0)$ $D_{i,j}=\min\begin{cases}M_{i-1,j}+c_{\boldsymbol{D}}(x_i,\varepsilon)\\I_{i-1,j}+c_{\boldsymbol{D}}(x_i,\varepsilon)\\D_{i-1,j}+c_{\boldsymbol{D}}(x_i,\varepsilon)\end{cases}$
5. if $(j>0)$ $I_{i,j}=\min\begin{cases}M_{i,j-1}+c_{\boldsymbol{I}}(\varepsilon_i,y_j)\\I_{i,j-1}+c_{\boldsymbol{I}}(\varepsilon_i,y_j)\\D_{i,j-1}+c_{\boldsymbol{I}}(\varepsilon_i,y_j)\end{cases}$
6. if $(i>0$ & $j>0)$
7. if $(x_i=y_j)$ $M_{i,j}=\min\begin{cases}M_{i-1,j-1}+c_{\boldsymbol{M}}(x_i,y_j)\\I_{i-1,j-1}+c_{\boldsymbol{M}}(x_i,y_j)\\D_{i-1,j-1}+c_{\boldsymbol{M}}(x_i,y_j)\end{cases}$
8. else $M^f_{i,j}=0$
9. return $\mathrm{Dist}(x^m,y^n)=\min(M_{m,n},D_{m,n},I_{m,n})$

图 2　改进的编辑距离算法

应用这个动态规划的方法，可以快速地计算出 2 个编码字符串的相似程度，也就是对应的查询与网页域之间的相关程度. MED 算法主要考虑查询词的出现与否、查询词在网页域中的位置、距离和顺序等信息，这些信息只有在简短的网页域中才比较有意义. 由于该算法与计算对象的长度有关，较长的网页域会导致计算性能下降，因此可以说该算法是专门用于计算查询与网页中简短域之间的相关性的，域越简短，计算速度越快.

3　实验分析

3.1　实验设置

为了验证 MED 算法在简短网页域相关性刻画上的优势，本文在来源于 MSN Search 的真实搜索引擎数据上进行了实验. 这个数据集拥有超过 25×10^{15} 个网页，28 250 个真实的用户查询，具有 6 个相关性等级，完全可以模拟当前商业搜索引擎的真实数据分布. 这个数据中包含了丰富的网页简短域，非常适合比较 MED 和传统词频方法.

关于评价指标，列出最为常用的 MAP、P@10、MRR、NDCG@1 和 NDCG@3 [3,10-11] 的结果，其他评价指标上的结果有类似的趋势. 其中：MAP 是计算所有查询的平均查准率的均值，是一个衡量全部搜索结果的指标；P@10 是统计前 10 个搜索结果的精度；MRR 是平均排序的倒转，主要衡量在检索出来的结果中第 1 个正确答案文档的位置；NDCG@n 衡量前 n 个搜索结果是否按照其相关性等级进行排序，用于表达前 n 个搜索结果的序列与标准相关性序列之间的一致性.

本文选取了最具代表性的 BM25 模型[4] 和临近信息模型[1]（记作 PROX），并把它们作为基于词频的传统相关性排序模型的代表与 MED 做比较.

3.2　实验结果

受篇幅限制，仅列出在锚文本和 URL 2 个域上的结果. 实验中用到的参数（无论是对比模型还是 MED 的参数）都是在训练数据集合上得到的最佳参数，然后在测试集合上直接使用它们. 为了克服实验的偶然性，采用了 5-fold 交叉验证方法. 所列结果是 5 个测试集合上得到的平均评测结果，所用参数如表 4 所示（仅列出 5 个结果中最接近平均结果的一组参数，其他 4 组参数与之差异非常小）.

表 4　实验参数

方法	网页域	参数
BM25	锚文本	$k_1=0.2$, $b=0.1$
	URL	$k_1=1.1$, $b=0.99$
PROX	锚文本	$k_1=0.4$, $b=0.1$
	URL	$k_1=1.3$, $b=0.99$
MED	锚文本	$I_1=1$, $I_2=3$, $D_1=7$, $D_2=40$
	URL	$I_1=1$, $I_2=1$, $D_1=4$, $D_2=33$

注：I_1、I_2、D_1、D_2 分别代表插入查询词、插入非查询词、临时删除和彻底删除操作.

首先对比传统的排序模型和 MED 在简短网页域上的排序结果，如图 3 和图 4 所示. 从图中结果可以看出，MED 在 2 个域的所有评价指标上都远远超过传统方法. 显著性测试表明，这些超越均具有统计意义上的显著性（概率 $p<0.05$）. 这说明，相比于传统的基于词频的相关性排序算法，MED 可以更好地表达用户查询与简短网页域之间的相关性.

另外，从实验结果中还可以看到一个比较有趣的现象，即在 URL 域中，引入距离信息的 PROX 的结果反而低于 BM25 结果. 这并不奇怪，因为 PROX 的基本假设就是“查询词在域中分布得越接近，则查询和域之间相关性越高”. 但是，URL 中重要的信息分布于 URL 字符串的两头，所以查询词在 URL 中

并非越近越好. 从锚文本的结果中也可以看出，PROX的上述假设有一定的合理性，所以在各种指标下都较BM25更好.

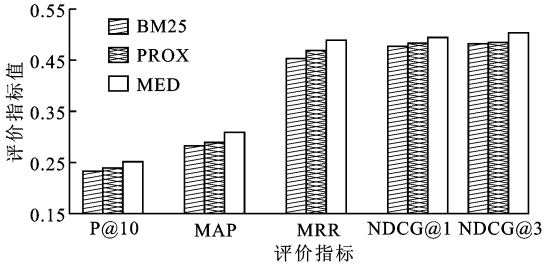

图3 锚文本域的排序结果

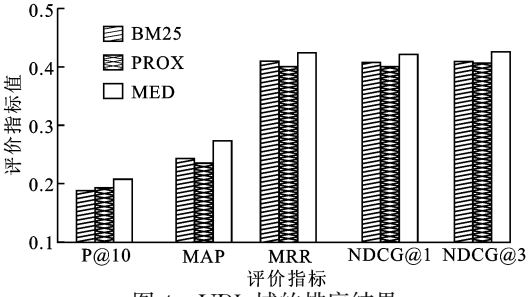

图4 URL域的排序结果

总的来说，MED比传统方法更适合衡量简短网页域与用户查询之间的相关性，对于改善整个网页的排序结果有着重要的意义.

4 结束语

本文的改进编辑距离算法，能有效地解决传统相关性排序算法不能准确表达简短网页域与用户查询之间的相关性问题. 通过引入代表相关性的顺序、距离和位置等信息，精心设计编码过程，将查询与简短域之间的相关性问题转化为编码后的字符串相似性问题，再利用动态规划来降低时间复杂度，使算法具有实用性. 在大规模真实搜索引擎数据上的实验结果表明，本文算法能显著提高排序性能. 这个结论对于搜索引擎算法的选择，具有很好的启示作用. 后续工作将重点放在如何通过机器学习的方法，自动地从标注好的数据集上学习算法的参数等.

参考文献：

[1] BRIN S, PAGE L. The anatomy of a large-scale hypertextual web search engine [J]. Computer Networks and ISDN Systems, 1998, 30 (1/7): 107-117.

[2] OGILVIE P, CALLAN J. Combining structural information and the use of priors in mixed named-page and homepage finding [EB/OL]. [2006-10-11]. http://www.scils.rutgers.edu/~muresan/IR/TREC/Notebooks/t12_notebook/papers/cmu-dir.web.pdf.

[3] BAEZA-YATES R A, RIBEIRO-NETO B A. Modern information retrieval [M]. Reading, MA, USA: Addison-Wesley Publishing Company, 1999.

[4] ROBERTSON S E, WALKER S, HANCOCK-BEAULIEU M. Experimentation as a way of life: OKAPI at TREC [J]. Information Processing & Management, 2000, 36(1): 95-108.

[5] BUTTCHER S, CLARKE C L A, LUSHMAN B. Term proximity scoring for ad-hoc retrieval on very large text collections [C] // Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, USA: ACM, 2006: 621-622.

[6] MITTAL V, BALUJA S, SAHAMI M. The happy searcher: challenges in web information retrieval [EB/OL]. [2006-12-10]. http://www.cs.cmu.edu/~har/PRICAI-2004.pdf.

[7] LEVENSHTEIN V I. Binary codes capable of correcting deletions, insertions, and reversals [J]. Soviet Physics Doklady, 1966, 10(8): 707-710.

[8] 车万翔，刘挺，秦兵，等. 基于改进编辑距离的中文相似句子检索 [J]. 高技术通讯，2004，14(7)：15.
CHE Wanxiang, LIU Ting, QIN Bing, et al. Similar Chinese sentence retrieval based on improved edit-distance [J]. High Technology Letters, 2004, 14(7): 15.

[9] CORMEN T H, LEISERSON C E, RIVEST R L, at el. Introduction to algorithms [M]. Beijing, China: China Machine Press, 2006.

[10] CRASWELL N, HAWKING D. Overview of the TREC-2002 Web track [EB/OL]. [2006-09-22]. http://trec.nist.gov/pubs/trec11/papers/WEB.OVER.pdf.

[11] JARVELIN K, KEKALAINEN J. IR evaluation methods for retrieving highly relevant documents [C] // Proceedings of the 23rd International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, USA: ACM, 2000: 41-48.

（编辑 苗凌）