ISSN 1000-0054 清华大学学报 (自然科学版) 2009 年 第 49 卷 第 S1 期 16/31
CN 11-2223/N J Tsinghua Univ (Sci & Tech), 2009, Vol. 49, No. S1 1328-1332

基于拼音索引的中文模糊匹配算法

曹 鞏[1,2], 邬晓钧[2], 夏云庆[2], 郑 方[2]

(1. 清华大学 计算机科学与技术系, 北京 100084;
2. 清华信息科学技术国家实验室 技术创新和开发部语音和语言技术中心, 北京 100084)

摘 要: 主流商业搜索引擎主要基于关键词精确匹配技术。为提高在用户的输入错误时的检索效率, 提出了有索引的汉语模糊匹配算法。该算法采用汉字、拼音和拼音改良的编辑距离这 3 种汉字相似程度的不同度量方式, 对用户查询进行扩展, 将模糊匹配转化为多个精确匹配, 对精确匹配的结果按与查询串的相似程度进行排序。在实验中, 将该方法应用于网页文本语料库中。在使用基于拼音改良的编辑距离度量方式时, 在时间和空间复杂度增长不大的情况下, 该方法取得了 60.42% 的准确率与 50.41% 召回率。

关键词: 文件信息处理; 拼音索引; 模糊匹配; 查询扩展

中图分类号: TP 391.1 **文献标识码**: A
文章编号: 1000-0054(2009)S1-1328-05

Pinyin-indexed method for approximate matching in Chinese

CAO Jiang[1,2], WU Xiaojun[2], XIA Yunqing[2], ZHENG Fang[2]

(1. Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China;
2. Center for Speech and Language Technologies, Division of Technical Innovation and Development, Tsinghua National Laboratory for Information Science and Technology, Beijing 100084, China)

Abstract: The exact matching of keywords is key to popular commercial search engines. A Chinese approximate matching method with an index structure was developed to achieve better retrieval when the input contains errors. Three types of similarity measurement between two Chinese strings were developed based on the character edit-distance, the Pinyin edit-distance and the Pinyin improved edit-distance. The similarity measurements were used to expand the user's query so that the approximate matching task can be represented as several exact matching sub-tasks. The results of these exact matchings are merged and sorted by their similarity to the original query. Tests on a webpage text database gave a 50.4% recall rate with the Pinyin improved edit-distance with a 60.4% precision with a small increase in time and space complexity.

Key words: pinyin-indexed; Pinyin (spelling of Chinese) index; approximate matching; query expansion

现有的主流商业信息检索系统大部分采用基于关键词精确匹配的检索技术, 取得了一定的成果[1]。但是在实际应用中, 用户的查询输入与检索系统数据库的构建都不可能完全正确[2]。用户对于搜索主题所处的领域不了解, 采用不合适的查询词, 会导致查询词的覆盖范围大大缩小[3]; 在中文信息检索系统中, 用户还常会输入同音或近音的错别字。模糊检索根据用户输入的模糊特征来检索匹配内容, 可处理精确的关键词匹配所无法解决的这些问题[4]。

在英文检索系统中, 通常对用户输入的单词进行拼写纠错, 就能解决大多数问题[5]。系统先搜索单词表, 找出所有与查询串中单词的编辑距离 (edit distance) 在一定限度之内的所有词汇, 再根据这些词汇来执行精确检索, 即可在一定程度上实现模糊检索[4]。所以, 英文模糊检索的主要研究集中在快速

收稿日期: 2009-03-19
基金项目: 国家自然科学基金资助项目 (60703051)
作者简介: 曹鞏(1984—), 男(汉), 湖北, 硕士研究生。
通讯联系人: 郑方, 研究员, E-mail: fzheng@tsinghua.edu.cn

简捷的字符串的模糊匹配算法方面[6-7]。考虑到查询串的整体性, 文[8]提出了块索引的方法, 通过二步的模糊匹配过程, 找到与查询串整体编辑距离在一定限度之内的串的位置。

汉语是典型的非字母语言, 把任意两个汉字的差别都算成同一个值不够精确。绝大多数汉字都是表意单元, 词语的搭配灵活多样, 难以建立完整的词表用于纠错。所以汉语的模糊检索无法照搬英文中的方法, 目前的研究主要集中在快速的汉字串模糊匹配算法方面[9-10]。其中, 文[9]的研究工作改善了模糊匹配的时空复杂度, 在实际系统中算法的时间复杂度可以达到子线性, 在实际的实验中也取得了很好的效果。然而, 遍历整个文本集来寻找相似串的出现位置, 虽然可以比较准确地完成模糊检索任务, 但随着数据集规模的进一步增长, 时间开销依然难以接受。

本文提出一种基于汉语拼音的模糊检索方法。与前述思路不同, 通过扩展原始的查询串, 将模糊检索任务转化为若干精确匹配的检索任务, 从而大大降低算法复杂度。

1 汉字串相似度度量

参考英文基于字母编辑距离的度量方式, 本文提出3种汉字串相似度的度量方式: 基于汉字的编辑距离、基于拼音的编辑距离, 以及基于拼音改良的编辑距离。

1.1 基于汉字的编辑距离

将单个汉字作为编辑距离中的距离度量单位, 即: 两个汉字串之间的距离, 等于使它们完全一样所需的最少替换、插入或者删除的汉字个数。

1.2 基于拼音的编辑距离

由于汉语拼音输入法的广泛使用, 大部分用户的输入错误都表现为同音字或者近音字的替换误用, 对Sogou实验室所提供的用户日志的分析结果也证实了这一点。基于此, 本文提出了基于拼音的编辑距离来衡量汉字串的相似度。如果把拼音串简单地看作广义的英文字母串, 则替换、插入或者删除一个字母后, 所得结果不一定是合法的拼音串。因此应从音节的角度来分析拼音串的差别。

对于一个单独的音节来说, 它与另外一个音节的差异总可以分解为以下三种变化: 声母变化、韵母变化和声调变化。声母、韵母和声调的可能取值都是有限的, 可以枚举定义从一种取值变为另一种取值的编辑距离。所以, 对于一个现有的音节, 容易找到所有与它编辑距离为n的音节。例如, 要找到所有与它编辑距离是2的音节, 那么变化可能是声母改变1个距离单位, 韵母改变1个距离单位, 声调改变0个距离单位; 或者声母改变2个距离单位, 韵母和声调没有发生改变。这只是一个排列组合的问题。

如果给所有音节编号, 将音节整体看作一个特殊的单字, 那么基于拼音的编辑距离可认为是基于汉字的编辑距离的细化, 即不同的汉字之间根据拼音的近似程度有不同的距离, 而不是笼统地将任意两个汉字的距离都计为1。

1.3 基于拼音改良的编辑距离

根据前述基于拼音的编辑距离定义, 音节/li3/和/ni3/的编辑距离是1, /li3/和/pi3/的编辑距离也是1。但是这2组音节的差异从发音机理角度来看, /li3/与/ni3/更加近似。类似的例子如: 音节/lin3/与/ling3/的差异较小, 而/lin3/和/lan3/的差异较大。

基于以上考虑, 提出改良的编辑距离计算方式如下。

1) 发音相似(容易发生替换错误)的声母或韵母之间差异小于1。例如, /l/和/n/、/z/和/zh/、/c/和/ch/等声母对, /in/和/ing/、/en/和/eng/等韵母对, 对音节编辑距离的贡献小于1(本文实验中赋予0.5的替换代价), 这样的声母和韵母对一共是9对。对于其余的声母或者韵母对的替换代价的计算, 则依然采用方法二中的使用字符编辑距离的计算方法。

2) 若同一音节的声母和韵母同时发生改变, 则在计算编辑距离时给予一个正的惩罚值(本文实验中取值为2)。根据这一计算方式, 对于音节串$A = I_1I_2I_3\cdots I_n$(其中I_i, $i = 1, 2, \cdots, n$代表一个音节), 若I_2在声母和韵母上同时发生差异为1的变化得到新的音节串$B = I_1I_2'I_3\cdots I_n$, I_1和I_3各发生一个声母或韵母的差异为1的变化得到新的音节串$C = I_1'I_2I_3'\cdots I_n$, 则$C$与$A$之间的编辑距离为2, 小于$B$与$A$之间的编辑距离4。

3) 音调变化导致的差异小于1。由于音调错误比较常见和普遍, 而且广泛使用的各种拼音输入法都不要求用户输入音调, 所以可认为音调差异小于一般声母和韵母之间的差异, 因而, 按照发音相似的韵母和声母对一样的处理方式, 这种差异在本文实验中赋予0.5的替换代价。

在本文实验中, 由于将所有小于1的差异都赋

值为0.5, 因此将所有的差异都乘以2之后, 可以得到结果为整数的编辑距离, 而这并不影响不同串之间相似度大小的比较。

2 索引与查询扩展

依据具体的距离度量方式, 可以扩展原始的查询串, 将模糊匹配转化成多个相关的精确匹配, 实现检索任务, 步骤是: 首先对查询串进行编辑距离由小到大的扩展, 然后对扩展出的查询串进行精确匹配, 精确匹配的结果在去重之后再按照查询串的编辑距离由小到大进行排序, 最后将排序的检索结果返回给用户。

2.1 建立索引

在本文提出的模糊检索系统中, 以离线方式对文本数据集中的单字或者音节建立索引。这是因为, 在第1节中所提出的3种距离度量方式都以单字或音节作为最小的考察和计算单位。当采用2种基于拼音的距离度量方式时, 要先将文本逐句地转成拼音串(逐句进行拼音的自动标注可以更好地联系上下文处理多音字、变音字等拼音现象), 然后再以音节为单位构建索引。

在文本数据集的索引表中, 需要记录索引头(本索引对应的单字或者音节), 以及索引头在文本数据集合中出现过的所有位置(文本号以及在文本中出现的具体位置)。

2.2 汉字串近邻空间

在对原始查询串进行扩展时, 需要引入汉字串近邻空间的概念。

定义1 在具体的汉字串相似度度量方式下, 所有与目标串编辑距离为m的汉字串组成的集合, 称为该目标串的m近邻空间。m近邻空间中每一个汉字串, 称为该目标串的一个m近邻串。

将查询串按编辑距离进行由小到大的扩展, 其实质就是依次计算查询串的各个近邻空间。而近邻空间内汉字串的数量, 直接影响遍历该近邻空间的时间复杂度。下面以基于汉字的编辑距离为例, 对近邻空间进行分析。

令汉语体系全部汉字的集合为Σ, 则所有长度为n的汉字串总数为$|\Sigma|^n$。对于一个长度为n(n远小于$|\Sigma|$)的汉字串X, 它的0近邻空间显然只包含它自身。

对于X的1近邻空间, 编辑距离可能由替换、插入和删除这3种方式之一造成。以替换为例, 长度为n的汉字串共有n个可能的替换位置, 每个替换都有$|\Sigma|-1$个候选的替换汉字, 因此只考虑替换可得到$n(|\Sigma|-1)$个与X编辑距离为1的汉字串。对插入和删除操作进行类似的分析可知: 只考虑插入可以得到约$(n+1)|\Sigma|$个汉字串, 只考虑删除则可以得到n个汉字串。所以总的来说, 1-近邻空间内所有汉字串的数目大约是$(2n+1)|\Sigma|$。在实际应用中, n一般不超过10, $|\Sigma|$的大小至少在10^3数量级上, 所以1-近邻空间内汉字串的数目也至少是10^3的数量级。

对于X的m近邻空间内的串, 一共发生了m处替换、插入或删除。可近似地认为串X共有$(n+1)$个位置能够插入汉字, 有n个位置的汉字能够删除或者替换。假设在m处变化中有a次插入, $m-a$次删除或替换, 则可产生新的汉字串约$C_{n+1}^{a}|\Sigma|^{a}\times C_{n}^{m-a}|\Sigma|^{m-a}$个(其中$C_{n+1}^{a}$表示组合数, 下同), 因此$m$近邻空间内所有串的数目大约为

$$\sum_{a=0}^{m} C_{n+1}^{a} C_{n}^{m-a}|\Sigma|^{m} = C_{2n+1}^{m}|\Sigma|^{m}.$$

在实际应用中, 虽然m的取值通常不超过$n/2$, 但m近邻空间内的串仍然数量巨大。

对于如此数量级的近邻空间, 不可能逐一访问其中每个近邻串来进行检索。即使有可能在较小的时间开销内判断某个汉字串是否符合中文语法, 从而去掉许多不合理的近邻串, 这一判断过程需要进行的次数也是惊人的。因此, 必须通过其他的方式来遍历近邻空间, 完成对原始查询串的扩展。

2.3 查询扩展和模糊检索

依然以基于汉字的编辑距离度量方式为例, 介绍本文的模糊检索系统进行查询扩展并检索的过程。

令用户输入的查询串为$A=a_1a_2\cdots a_n$, 其中a_i($i=1,2,\cdots,n$)是汉字。在A的1近邻空间中, 所有因a_i被替换所生成的串有$B_i=a_1a_2\cdots a_{i-1}x\,a_{i+1}\cdots a_n$的形式, 其中$x\in\Sigma$且$x\neq a_i$。$B_i$实际上代表$|\Sigma|-1$个不同汉字串, 称$B_i$为通配串。在检索时, 先分别精确地检索两个子串$B_{i1}=a_1a_2\cdots a_{i-1}$和$B_{i2}=a_{i+1}a_{i+2}\cdots a_n$在数据集中出现的位置, 然后对两个子串的位置进行比较, 找到串B_{i1}和串B_{i2}同时出现且B_{i1}在B_{i2}的$i+1$个位置前出现的文本。由于替换可能发生在n个不同的位置, 所以类似的通配串有n个。对于插入和删除两种改变方式, 可构造类似的通配串扩展并检索。

分析上述过程的复杂度。考虑插入、删除和替换这3种操作, 则在1-近邻空间内通配串的总数目大

约是3n, 每个串通过至多2 次的精确匹配实现模糊检索。考虑到精确匹配的结果可以被不同的通配串共享, 实际上只需经过2n- 1 次精确匹配即可实现1-近邻空间内的所有串的检索, 这相对于1-近邻空间内串的总数量大大减少了。

对于A 的m 近邻空间(m > 1), 方法是类似的。在编辑距离为m 时, 可近似认为串有2n+ 1 个位置可能发生改变, 相应通配串的数目约为C^{m}_{2n+1}。由于进行精确匹配的通配串的子串实际上都是原输入串的子串, 而原输入串的子串总数量为$n(n+1)/2$, 所以实际最多进行$n(n+1)/2$ 次精确匹配。

随着m 的增长(m < n), 通配串的数目也逐渐增大。在实际应用过程中, 可以设置返回给用户的查询结果的数量上限, 以此作为算法结束的阈值。通常在m 还处于一个比较小的数值时, 扩展就可停止。此外, 由于实际检索时n 不会很大, 与扩展串精确匹配所需要的时间开销相比, 计算扩展串本身的时间开销可以忽略, 实验的相关数据也证实了这一点。

上述查询扩展是在基于汉字的编辑距离度量方式下实现的。当采用基于拼音的两种编辑距离度量方式时, 总体思路类似, 区别在以下2 点: 1) 在获取用户查询串之后, 首先将它转换为拼音串; 2) 在查询串的扩展过程中, 替换生成的通配串B_i 中的x 不可能取所有音节, 而是与a_i 差异为1 的音节集合W 中的元素, 因此在考察子串B_{i1}和B_{i2}的相对位置关系时, 需要判断它们之间的那个音节是否属于集合W。

对于以上得到的查询结果, 首先要进行去重, 然后按照它们被检索出来时所考虑的编辑距离, 从小到大依次排列, 反馈给用户。

综上所述, 通过将用户查询扩展并分割成多个精确匹配的查询, 避免了逐一遍历整个近邻空间的操作, 仅仅在通常的精确检索的几倍时间内, 完成模糊检索的任务。

3 实 验

对于本文中提出的查询扩展算法的正确性, 也就是说查询扩展算法是否按照定义的距离度量方式从"近"到"远"的返回相应的文本, 在算法层面上已经得到了证明, 对于输出结果的抽查也保证了这一点。

同时, 本文提出的基于索引的匹配方法相比较常见的依次顺序匹配的方法在时间复杂度上的差异十分明显, 所以并不是关注的焦点。

因而, 设计实验的主要目的在于考察以上几种不同的距离度量方式, 是否能够很好的满足用户的需求, 因而设计了以下几组实验进行测试比较:

3.1 实验设计

实验中使用的文本数据来源于搜狐—清华大学联合实验室(Sogou, 搜狗实验室)提供的网页文本数据集合, 从中选取了约4 万篇长度处于整个数据集合平均长度附近的文本。这些文本全部来源于真实网页, 并且去除了html 标引符号, 只保留了文本内容。文本数据集的总大小为79.3MB, 每个文本的平均长度为930 个汉字。将这些数据作为实验中待匹配的文本。

实验中一共采用了400 组用户查询串作为匹配的目标串。每组查询串由一个错误查询和与之对应的正确查询组成。错误查询全部来源于Sogou 实验室提供的用户查询日志, 由人工选取其中一些明显有错的查询, 并且人工对它进行了更正。为了不失一般性, 在选取的过程中没有刻意挑选错误类型为拼音的查询串, 因而这些错误查询能够在一定程度上代表信息检索系统实际的用户错误输入情况。经统计, 这些错误查询的平均长度为6 个汉字, 其中95% 左右是拼音相关错误。

在实际的实验中, 错误串可以被认为代表用户给予搜索引擎的错误输入, 而人工更正则代表用户的真实意图。因而使用错误输入进行匹配的结果, 是否能够满足用户的真实意图, 就是要考察的重要指标。

3.2 评测指标

假设查询串B 是错误查询, 串A 是它的人工更正。对A 进行精确匹配的检索, 得到结果集包含有n 个结果: $\Omega=\{W_1, W_2, W_3, \cdots, W_n\}$, 把$\Omega$ 视作本组查询的标准输出。

如果对串B 进行精确匹配的检索, 则无法得到正确的查询结果。在实验中, 分别利用第2 节中提到的3 种编辑距离度量方式来对错误串进行模糊检索。令排序后输出的前p 个文本构成集合$\Delta=\{X_1, X_2, X_3, \cdots, X_p\}$, 而且在这$p$ 个文本中, 若有x 个文本在集合Ω 中, 则认为模糊匹配系统Top$p$ 的准确率为x/p, 召回率为x/n。这两个指标可以用来实际反映本文提出的模糊匹配系统对于用户意图的匹配程度。

3.3 实验结果和分析

实验所用的文本集合规模为4 万, 正确的查询

串一共400个,它们得到的检索结果Ω平均含大约7篇文本。其中,包含结果数不多于3的query有31.25%,包含结果数不多于10的query有78.75%,包含结果数不多于30的query有97.5%。

完成了3组对比实验。实验数据结果如表1。其中,Top 3、Top 10和Top 30分别指在前3个、前10个和前30个查询返回结果的集合上进行统计和计算的结果。

表1 模糊匹配系统的准确率和召回率

距离度量方式	准确率/%			召回率/%		
	Top 3	Top 10	Top 30	Top 3	Top 10	Top 30
基于汉字	31.48	18.15	11.77	45.33	60.98	75.57
基于拼音	53.70	28.23	16.82	51.40	76.55	89.52
基于拼音的改进	60.42	34.17	19.62	54.31	84.45	91.70

从表1可以看出,在准确率和召回率这2项指标上,2种基于拼音的度量方式都比基于汉字的度量方式有较大提高,这是由于拼音能够更好地刻画汉字串之间的相似程度。基于拼音改良的度量方式,在准确率和召回率上,都取得了最好的实验结果,说明引入语音学知识对性能提高有帮助。

从表1还可以看出,前30个查询结果已经能够达到90%左右的召回率,而返回前30个结果,一般只需要对原始的错误查询串扩展出不多于10次的扩展。此外,对于4万个文本组成的共79.3MB的文本集合,建立的索引(单级索引,未压缩)大小在100 MB左右,基于音节建立的索引与基于单字建立的索引相比,大小并没有显著增加。

4 结论与未来工作

本文所提出的基于拼音改良的编辑距离度量方式,依然存在一些局限性。虽然在实际检索系统的输入中,拼音相关错误占有非常高的比例(在Sogou用户日志的统计中,这一比例超过了90%),但是对于其他类型的错误,例如形近字和近义词带来的错误,本方法提出的距离度量方法并不能很好地进行度量和表征,依然有待进一步的研究。

同时,工作在方法依然有较大的改进空间:可以引入语言学方面的知识,以及根据用户日志的实际统计结果,对于每一种改变赋予的权值采用更细致合理的规定。如果两个汉字串的相似度不能方便地转换成整数,如何完备地对查询进行扩展,也是值得研究的问题。

未来,计划研究本文提出的查询扩展和检索思路在多级索引下的应用,考察模糊检索的性能,并进行相应的算法改进。同时,结合汉语中的词汇和二元拼音文法进行拼音索引,引入这些先验知识可以有效地降低查询扩展的次数,同时实际的用户大部分也是以词为单位进行输入,引入词汇知识可以更好地匹配用户的实际情况。

致谢 本文在研究工作中,使用了搜狗(Sogou)实验室无偿开放的文本数据和用户日志,在此表示衷心的感谢。

参考文献 (References)

[1] Manning C, Raghavan P, Schütze H. Introduction to Information Retrieval [M]. Cambridge University Press. 2008.

[2] Mitra M, Singhal A, Buckley C. Improving Automatic Query Expansion [C]//Proc of the 21 st Ann Int ACM-SIGIR Conference on Reser and Dev in Info Retrieval, 1998.

[3] Araujo M, Navarro G, Ziviani N. Large text searching allowing errors [C]//Proc WSP97. Valparaiso, Chile: Carleton University Press. 1997, 8: 2-20.

[4] Navapro G. A guided tour to approximate string matching [J]. *ACM Computing Surveys*, 2001, **33**(1): 31-88.

[5] Boyer R, Moore J. A fast string searching algorithm [J]. *Communications of the ACM*, 1977, **20**(10): 762-772.

[6] Tarhio J, Ukkonen E. Approximate Boyer-Moore string matching [J]. *SIAM J on Computing*, 1993, **22**(2): 243-260.

[7] Knuth D, Morris J, Pratt V. Fast pattern matching in strings [J]. *SIAM J on Computing*, 1977, **6**(2): 323-350.

[8] Baeza-Yates R, Navarro G. Block-addressing indices for approximate text retrieval [J]. *J Am Soc Info Sci*, 2000, **51**(1): 69-82

[9] 王静帆, 邬晓钧, 夏云庆, 等. 中文信息检索系统的模糊匹配算法研究和实现 [J]. 中文信息学报, 2007, **21**(6): 59-64.
WANG Jingfan, WU Xiaojun, XIA Yunqing, et al. An approximate string matching algorithm for Chinese information retrieval systems [J]. *J Chin Info Proc*, 2007, **21**(6): 59-64. (in Chinese)

[10] 陈儒. 面向短信过滤的中文信息模糊匹配技术 [D]. 哈尔滨: 哈尔滨工业大学信息检索实验室, 2003.
CHEN Ru. Approximate String Matching Algorithm for Cell-phone Notes Filtering [D]. Info Retrieval Lab, Harbin Institute of Technology, 2003. (in Chinese)