利用正则表达式进行查找/替换*

翟自洋[1,2)]　林昌东[1,2)**]

收稿日期:2007-10-15
修回日期:2008-08-29

1)《浙江大学学报(英文版)》编辑部, 310027　杭州市浙大路 38 号, E-mail: jzus_zzy@ zju. edu. cn
2) 浙江大学出版社, 310028　杭州市天目山路 148 号

摘　要　查找/替换在文本编辑中常会用到,看似简单,实则内涵丰富。正则表达式(Regular Expression)的采用,实现了对复杂文本的匹配,极大地增强了查找/替换的功能。正则表达式与文本编辑软件(如 EmEditor™等)所支持的"在多个文件中查找/替换"相结合,使编辑人员可对各种纯文本文档(如 txt,xml,tex,htm,等)同时进行批量查找/替换,大大提高了工作效率。本文从笔者编辑工作的典型实例出发,简要介绍了正则表达式的基本概念,部分展现了其应用于查找/替换的巨大潜力。指出结合正则表达式的查找/替换还可用于数据验证,作为人工审校的补充。针对英文期刊总结了一些验证表达式。并指出,对中文或中英文混排的书刊同样大有用武之地。

关键词　查找/替换　正则表达式(Regular Expression)　在多个文件中查找/替换　数据验证　EmEditor™

查找/替换(以下简称"查/替")是编辑软件的一项基本功能。一般编辑软件只能处理简单的查/替,但目前越来越多的编辑软件在查/替功能中实现了正则表达式(Regular Expression),其内涵丰富,功能极为强大,可使一些烦琐费时且容易出错的工作变得轻松有趣。

笔者在将《浙江大学学报(英文版)》B 辑的 Word 文档转换成 xml 文件,提交 PubMed Central 数据库[1]时,被查出一个全局性的问题:公式编号(1),(2),……误为 1,2,……括号全部漏掉。

总共 15 期数据、260 多篇文档,如果手工逐一修改,至少得花几个小时,并且极可能错漏。

xml 文件中,与公式编号相对应的内容为: < label > 1 < /label > , < label > 2 < /label > ,……正确的应为 < label > (1) < /label > , < label > (2) < /label > ,……如果在编辑软件中直接将前者对应地替换为后者,公式编号的问题是解决了,但同时又引入了新的错误。因为公式以外也有 < label > 1 < /label > ,……出现,例如作者单位 < aff id = "aff1" > < label > 1 < /label > ... < /aff > 中的 1 不能改为(1)!

利用正则表达式,以及某些文本编辑软件(如 Emurasoft 公司的 EmEditor™)支持的"在多个文件中查找/替换"功能,可以准确、快速地解决这一问题。

1　正则表达式简介

DOS 命令中用通配符(Wildcard Character)"?"和"*"分别表示任意单个字符和任意字符串,例如命令"dir *. doc"将列出当前目录下所有 Word 文档。这两个符号可看作正则表达式最简单的子集。

Microsoft Word 定义了一套通配符,在其帮助文件给出了说明[2]。对通配符的支持使得 Word 的查/替功能大大增强。但 Word 所支持的通配符同样只是完整正则表达式体系的一个小的子集。

限于篇幅,这里只对正则表达式作简要介绍,以文本编辑软件 EmEditor 采用的正则表达式体系为例。更详尽地了解正则表达式,请参见文献[3,4,5]。

正则表达式是由普通字符(Literals)和元字符(Metacharacters)构成的文本模式。元字符表示特殊的含义,包括"\","^","$","*","+","?",".","(",")","{","}","[","]"和"|"。元字符前加"\"转义为符号自身。**元字符均为半角字符**。普通字符是除元字符外的所有字符,普通字符只匹配自身[6]。

表 1[6]列出了元字符的含义,其中包括三组括号(分别成对出现)构成的常用组合,同时给出简单示例。正则表达式的语法初看上去复杂,其实理解起来不难,并不需要专门的计算机知识。

构造正则表达式,就是将元字符与普通字符通过各种方式结合在一起,得到需要的匹配或捕获。如果把构造正则表达式比作创建数学表达式,元字符就相当于加减乘除等运算符号。字符的结合存在先后法则。元字符的优先级从高到

* 国家自然科学基金重点学术期刊资助项目(No. 30824802)
** **通讯作者**: 310028　杭州市天目山路 148 号, E-mail: lcd@ zju. edu. cn

表 1　正则表达式中元字符的含义[①]

元字符	含义
\	转义。正则表达式中最重要、最常用的字符。所谓转义,就是改变紧跟其后的字符(普通字符或元字符,只能是半角字符)的本来意义,并与后一字符一起构成特殊字符、原义字符或反向引用[②]。例如,“\n”匹配换行符;“\\”匹配“\”;\1 ~ \9 表示反向引用
^	匹配输入字符串的开始位置。例如,“^e”匹配任何以 e 起始的字符串中打头的 e
$	匹配输入字符串的结束位置。例如,“e$ ”匹配任何以 e 结束的字符串中末尾的 e
{*n*}	匹配前面的子表达式 *n* 次,*n* 为非负整数
{*n*,}	至少匹配前面的子表达式 *n* 次,*n* 为非负整数
{*m*,*n*}	至少匹配前面的子表达式 *m* 次且至多匹配 *n* 次,*m* 和 *n* 均为非负整数,*m* < = *n*。逗号和两个数字之间不能有空格
*	匹配前面的子表达式零次或多次。等价于{0,}
+	匹配前面的子表达式一次或多次。等价于 {1,}
?	匹配前面的子表达式零次或一次。等价于 {0,1}
?	非贪婪模式。跟在限制符——*, +, ?, {*n*}, {*n*,}, {*m*,*n*}——之后,尽可能少地匹配所搜索的字符串;而默认的贪婪模式则尽可能多地匹配所搜索的字符串。例如,对于字符串 oooo,“o +?”将匹配单个 o,而“o + ”将匹配所有 o
.	匹配除“\n”以外的任何单个字符(包括可见和不可见字符)
(pattern)	分组捕获。匹配 pattern 并获取这一匹配供以后引用。捕获的分组可供反向引用
(?:pattern)	非捕获匹配,表示限定。常与“|”组合使用,例如,“industr(?:y|ies)”比“industry|industries”更简略
(? =pattern)	正限定,在任何匹配 pattern 的字符串之前匹配待查找的字符串,为非捕获匹配。例如, “x(? =abc)”只匹配在“abc”之前出现的 x
(?! pattern)	负限定,在不匹配 pattern 的字符串之前匹配待查找的字符串,为非捕获匹配。例如,“x(?! abc) ”匹配不在“abc”之前出现的 x
x|*y*	匹配 *x* 或 *y*。例如,“z|food”匹配“z”或“food”,“(z|f)ood”则匹配“zood”或“food”
[*xyz*]	字符集合。匹配包含的任意一个字符。例如,“[abc]”可以匹配“plain”中的“a”
[^*xyz*]	匹配未包含的任意字符。例如“[^abc]”可以匹配“plain”中的“p”,“l”,“i”,“n”
[*x* - *y*]	字符范围。匹配指定范围内的任意字符。例如,“[a - z]”匹配“a”到“z”的任意小写字母
[^*x* - *y*]	匹配不在指定范围内的任意字符。例如,“[^a - z]”匹配除了 26 个小写字母以外的任意字符

注:[①]译自 EmEditor V5.00 帮助文件[6],有删改

[②]反向引用(back reference)标识由正则表达式中的匹配组捕获的子字符串,由转义字符“\”与数字 1 ~ 9 构成;引用时,从左至右,分别对应于\1,\2,……,\9。例如,“(a)\1”捕获“a”作为第一个反向引用,同时匹配“aa”。反向引用也常用于替换表达式。例如,“(h)(e)”找到“he”,替换表达式若为“\1”,则将“he”替换为“h”;若为“\2\1”,则替换为“eh”

低为[4]:转义符——\;括号和中括号——( ),(?:),(? =),[ ];限定符——*, +,?, {*n*}, {*n*,}, {*m*,*n*};定位点和序列——^, $,\任意元字符,任意字符;替换—— |。相同优先级从左到右,不同优先级先高后低。

为便于使用,某些普通字符和元字符一起构成具有特定含义的组合,这种组合往往作为构造正则表达式的基本组件。例如,下面要提到的字符集(Character Classes)、表示单个字符及字符集的转义序列(Escape Sequences)等。借助数学表达式来理解:表达式 $a^2 + b^2$ 中的 a^2,由字母 a(相当于正则表达式的普通字符)和平方运算(相当于元字符)构成,但通常将 a^2 视为一个整体。

字符集(Character Classes)

由小写字母和元字符“[”“]”构成,使用格式为“[:classname:]”,包含全角和半角字符。这里“classname”可以

是[6]：alnum——字母、数字；alpha——a－z，A－Z 及其他字母；space——空白符；digit——数字 0～9；punct——标点；xdigit——16 进制数 0－9，a－f 和 A－F；word——字母、数字及下划线；unicode——非 ASCII 码字符，等等。例如，“[[：digit：]]”表示任意数字 0～9，“[[：punct：]]”表示任意标点。

转义序列(Escape Sequences)

所谓转义序列，是指通过“\”转义得到的具有特殊含义的组合，表示单个字符或字符集。

表示单个字符的转义序列(Single Character Escape Sequences)，如[6]：

\c*x* ASCII 控制字符。比如\cC 代表 Ctrl＋C。*x* 的值须为 A－Z 或 a－z；

\f 换页符；\n 换行符；\r 回车符；\s 任何空白字符，包括空格、制表符等；\t 制表符；\v 垂直制表符；\0dd，\xXXXX 分别表示八进制数 dd 和十六进制数 0xXXXX(d，X 分别是八进制和十六进制字符)。等等。

表示字符集的转义序列(Character Class Escape Sequences)，如[6]：

\w 任意字母、数字或下划线；\s 任意空白符；\d 0～9 的任意数字；\l 任意小写字母 a－z；\u 任意大写字母 A－Z。以上，“\”与对应的大写字母结合，表示相应的补集。

正则表达式通常用于查找表达式；部分可用于替换表达式，例如[6]：

\0 对整个正则表达式匹配对象的反向引用；\1～\9 反向引用(见表 1)；\n 匹配新的一行；\r 多个文件中查/替时，匹配回车符；\t 制表符。

用于替换表达式，EmEditor Professional 中还有以下强制转换符：

\L 转为小写；\U 转为大写；\H 转为半角；\F 转为全角；\E 关闭以上强制转换字符。

为加深理解，表 2 给出了正则表达式应用的一些简单示例。

表 2　正则表达式应用举例

任务	查找表达式	替换表达式
在行首插入//	^	//
首字母大写，其余小写	([a－zA－Z])([a－zA－Z]＊)	\U\1\L\2
将日期格式“月/日/年”统一修改为“年.月.日”	([0－9]{1,2})/([0－9]{1,2})/([0－9]{2,4})	\3.\1.\2
将半角数字0～9统一改为全角	([[：digit：]])	\F\0

总的来说，正则表达式使得查/替不局限于字面上一致，同时能够为匹配作出严格限定，所以一方面扩大了匹配范围，一方面又可以精确匹配(缩小匹配范围)。引入正则表达式后，查/替的功能几乎是无限的，或者说只限于你的想象。

须特别指出，虽然正则表达式已有国际标准[7]，但不同编辑软件采用的正则表达式会有一些差别。EmEditor 采用的正则表达式，功能远远超出 Word 的通配符，其语法基于 Dr. John Maddock 的 Perl regular expression syntax，也只是其子集[6]。对中文有良好支持的文本编辑软件 TextPro，其正则表达式是在 Henry Spencer 源码基础上增加对双字节字符的支持得到的[8]。例如，Word 中“?”“＊”分别表示“任意单个字符”和“任意字符串”[2]，EmEditor 和 TextPro 中“?”“＊”则为限定符(见表 1)。EmEditor 和 TextPro 中“＋”的功能，Word 则由“@”来完成[2]。

2　正则表达式与“在多个文件中查/替”

正则表达式在文本编辑中强大功能的发挥需要好的文本编辑软件支持。EmEditor™ 是 Emurasoft 公司开发的一款简单好用的文本编辑器，支持多种编码(字符集)，可无限撤消、恢复，支持二次开发(提供类似 Word 的宏功能)，它的查/替功能尤有特色[6]。多种文件类型、在多个文件中查/替、正则表达式，对这三者的支持，使得 EmEditor 的查/替功能无比强大。打开 EmEditor Professional(Version 5.00，汉化版)，在菜单栏选择：搜索—＞在文件中替换，将看到如图 1 所示的界面。

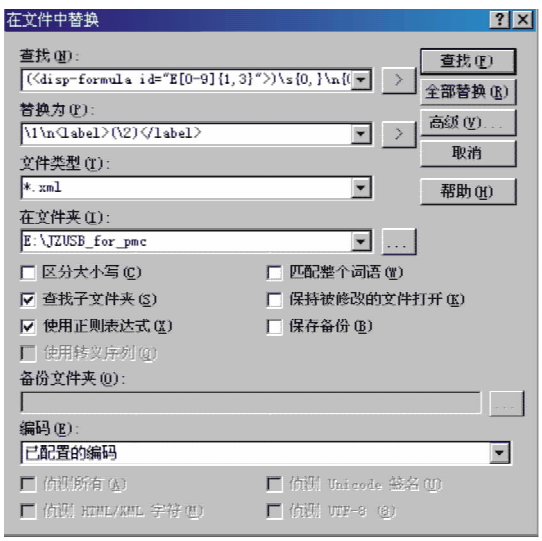

图 1　EmEditor“在文件中替换”对话框

对于纯文本格式,如 txt,xml,tex,htm 等,可以直接对某个文件夹中的所有文件(可选择是否查找子文件夹,见图 1)进行替换。如果同时对不同格式的文件进行查/替,可在“文件类型”栏以分号分隔,如“ *.txt; *.xml; *.tex”。

须特别注意的是,对多个文件的替换不能撤销! 为防误操作,务必谨慎,必要时做好备份。

非纯文本格式的文件,如 Word 文档,因其编码的复杂性,经测试,EmEditor 对多个 Word 文档只支持常规的查找,不支持正则表达式。

3 问题的解决

再看前面提出的问题。正则表达式可以实现满足指定条件的精确匹配,因此提供了排除如作者单位中“ <label>1</label>”干扰的可能,而在多个文件中查/替使得对大量文件的批处理成为可能。

为实现精确匹配,须对公式中 <label>... </label> 在文中出现的情形作全面分析。分析发现,公式中 <label>... </label> 总是紧跟 <disp-formula id = "E[0-9]{1,}" > 出现。这里“E[0-9]{1,}”是公式 id,“[0-9]{1,}”表示 1 位以上数字(见表 1)。具体有以下几种情形:

(1)… <disp-formula id = "E1" >

<label>1</label>

…

说明:<label> 另起一行。<disp-formula id = "E1" > 之后可能有一个以上空格。

(2)… <disp-formula id = "E5" >

<label>3</label>

…

说明:与(1)对比,表明公式 id 中的数字与 <label> 中可能不同。

(3)… <disp-formula id = "E12" > <label>10</label>

…

说明:二者处于同一行,中间可能有零至多个空格。

以上三种情况包含了公式中 <label>… </label> 的所有情形。

解决办法:

在 EmEditor 中打开图 1 所示界面,文件类型为 *.xml,在文件夹 E:\JZUSB_for_pmc(各期数据置于该目录下各自独立的文件夹),选中“查找子文件夹”及“使用正则表达式”,在“查找”和“替换为”栏分别输入:

查找:(<disp-formula id = "E[0-9]{1,}" >)\s{0,}\n{0,1}\s{0,} <label> ([0-9]{1,}) </label>

替换为:\1\n <label> (\2) </label>

点击“全部替换”,几分钟内(具体时间与计算机配置有关)任务完成。

对查找表达式的解释:“()”中的内容被获取,供替换时引用原文;这里是对原文的引用,故不必关注公式 id 的数字与 <label> 中是否相同。\s{0,} 表示 0 个或 1 个以上空格。\n{0,1} 包含了 <label> 与 <disp-formula…在同一行或另起一行两种情形。这些限定排除了干扰,同时包含了公式中 <label> 的所有情形。

值得特别注意的是,对多个文件的替换不能撤消,所以务必对替换对象进行准确、全面的分析,以免错漏。编辑人员须熟练运用正则表达式,才能在 EmEditor 中对多个文件替换。如不确定,可先进行测试,必要时做好备份。替换后应立即检查,万一操作失误,须冷静分析,并根据具体情况选用新的替换表达式,及时纠正错误。

4 在编辑工作中的应用

查/替的功能包含两个方面:搜索/替换文本和验证数据。搜索/替换功能在上面的示例中有了生动体现。正则表达式与“在多个文件中查/替”相结合,也提供了强大的数据验证功能。笔者在把 Word 文档转换成 PMC 所要求的 xml 文件[1]时,利用 EmEditor 提供的数据验证功能,确保了 xml 数据的准确性,提高了编辑效率。

Word 是一款优秀的排版软件,得到了广泛应用。如果能利用 EmEditor 同时对多个 Word 文档进行有效的数据验证,将会是对人工审校的有益补充。EmEditor 对多个 Word 文档的查/替不支持正则表达式。可以将 Word 文档拷贝到 EmEditor,转换为同名的 txt 文件。转换后,图、表、公式等将会丢失,因为只是为了验证文本,故不必考虑。EmEditor 支持多种编码,通常情况下,格式转换后文本能够保持一致。值得注意的是,Word 有自己的编码规则,有些字符,如“符号”(symbol),在 EmEditor 中无法正确显示。

国内中文图书、期刊大多采用方正软件完成。方正排版的文档实际上也是纯文本文档,对于其中的全局性差错,完全可以用 EmEditor 等软件实现复杂的查/替,尤其是公式中的全局性错误。另一种常用的排版软件是 TeX/LaTeX。TeX/LaTeX 文档由 ASCII 码组成(支持中文时,采用国际标准码,CTeX[9]),同样是纯文本格式,可以方便地利用 EmEditor 进行数据验证,并且可以直接修改文档。

笔者在编辑实践中,针对英文期刊容易出现的错误,总结了一些验证表达式,列举如下:

(1) 错用了全角标点

查找:[,;.、。?! “”‘’()[]{}……——<>《》]

说明:将在 EmEditor 一个页面下显示出所有目标文档(存于同一目录)中出现的全角标点。

补充说明:

① 对中文文档,可用类似方式查半角标点。

② 仅有验证还不够,如何修改? 利用 EmEditor Professional 提供的强制转换符“\H”(转为半角)和“\F”(转为全角),可以方便地实现全角、半角标点的互换。例如,修改为半角字符:

查找:[,;.:?!()[]{}]

替换为:\H\0

说明:\0 后留一空格。

若查找[[:punct:]],替换为\H\0,可将所有全角标点强制转换为半角,但有些中文标点(如“”,“。”,等)无法转为对应的英文标点(“”,“.”,等)。

(2) 查找英文文档中是否出现中文字符

查找:[\x3400-\x9fff\xf900-\xfa2d]

说明:汉字对应的 Unicode 码(以 4 位十六进制字符表示)。

(3) 文本与标点之间有多余空格,或标点后缺少空格

如,The scalable extension of H.264/AVC , known as “scalable video coding” or SVC, is currently... 这里第一个逗号前面多了空格,第二个后面则少一空格。英文文章易出现这样的错误。

查找:\s[,;.!?”]以及[,;!?](?!)

说明:前者匹配出现在空白字符之后的“,;.!?””,后者匹配其后未紧跟空格的“,;!?”。因标点在文中出现的情况较复杂(如右引号常出现在其他标点之后),须对查找结果加以分析。

(4) 英文科技文献常对反复出现的术语采用缩写(通常为大写字母),缩写词前的冠词 a/an 容易用错

查找:a[AEFHILMNORSUX]及 an[FHLMNRSUX]

说明:有些缩写习惯上视为独立单词,并形成新的读音,这时冠词用 a 还是 an 应视具体情形而定。

(5) 检查 4 位以上的阿拉伯数字是否分节(Word 中可采用不间断空格为数字分节)

查找:[0-9]{4}

说明:编辑规范要求,4 位以上数字,以小数点为界,向左或向右,每 3 位之间留空。这里找出连续 4 位数字。应从结果中排除年份。

(6) 查找重复的单词

查找:\b(\w+)\b\s+\1\b

说明:\b(\w+)\b 表示一个单词:\b 表征单词的开头或结尾,\w+表示一个以上(包括一个)字母。\s+表示 1 个或多个空白符,\1 是对前面匹配的单词的反向引用。如可以找出“is is”,“a a”,等。

以上列举了查/替在数据验证中应用的一些实例,主要针对英文期刊;对中文或中英文混排书刊的数据验证,理论上是相通的,不难从以上实例举一反三,具体的有待大家在实践中总结和挖掘。除文中介绍的 EmEditor 以外,文本编辑软件 TextPro,支持双字节的正则表达式和首创的自定义替换,功能强大。TextPro 是综合处理中文文本文件和超文本文件的免费软件,包括查/替在内,共支持 20 种功能[8],中文期刊不妨选用。

参考文献

1 翟自洋,林昌东,林汉枫,伍秀芳. 加入 PubMed Central 的实践及其对期刊的积极影响. 中国科技期刊研究. 2007,18(5):761-765

2 Microsoft Corporation, 1999. Microsoft Word 2000 帮助文件.

3 正则表达式 30 分钟入门教程(第二版), 2007. [2008-01-25]. http://unibetter.com/deerchao/zhengzhe-biaodashi-jiaocheng-se.htm

4 Microsoft Corporation, 2008. JScript.NET 正则表达式介绍. [2008-01-25]. http://msdn.microsoft.com/library/chs/default.asp?url=/library/CHS/jscript7/html/jsreconintroductiontoregularexpressions.asp

5 Jan Goyvaerts, 2006. Regular-Expressions.info. [2007-8-28]. http://www.regular-expressions.info/

6 Emurasoft, Inc., 2005. EmEditor Professional Version 5.00 RC 16 Help.

7 The Open Group, 1997. Regular Expressions. The Single UNIX Specification, Version 2. [2008-1-25]. http://www.opengroup.org/onlinepubs/007908799/xbd/re.html

8 TextPro V5.1 (for Win9x/NT, Windows 2000), 2003. [2007-02-14]. http://www.fodian.net/tools

9 中文 CTeX 网站. [2007-12-12]. http://www.ctex.org