

资源 | 正则表达式的功夫大全，做NLP再也不怕搞不定字符串了

机器之心 今天

选自Medium

作者：Jonny Fox

机器之心编译

参与：思源

在自然语言处理中，很多时候我们都需要从文本或字符串中抽取出想要的信息，并进一步做语义理解或其它处理。在本文中，作者由基础到高级介绍了很多正则表达式，这些表达式或规则在很多编程语言中都是通用的。

正则表达式（regex 或 regexp）对于从文本中抽取信息极其有用，它一般会搜索匹配特定模式的语句，而这种模式及具体的 ASCII 序列或 Unicode 字符。从解析/替代字符串、预处理数据到网页爬取，正则表达式的应用范围非常广。

其中一个比较有意思的地方是，只要我们学会了正则表达式的语句，我们几乎可以将其应用于多有的编程语言，包括 JavaScript、Python、Ruby 和 Java 等。只不过对于各编程语言所支持的最高级特征与语法有细微的区别。

下面我们可以具体讨论一些案例与解释。

基本语句

锚点：^ 和 \$

<code>^The</code>	匹配任何以“The”开头的字符串 -> Try it! (https://regex101.com/r/c08lqs/2)
<code>end\$</code>	匹配以“end”为结尾的字符串
<code>^The end\$</code>	抽取匹配从“The”开始到“end”结束的字符串
<code>roar</code>	匹配任何带有文本“roar”的字符串

数量符：*、+、? 和 {}

abc*	匹配在“ab”后面跟着零个或多个“c”的字符串 -> Try it! (https://regex101.com/r/c08lqs/)
abc+	匹配在“ab”后面跟着一个或多个“c”的字符串
abc?	匹配在“ab”后面跟着零个或一个“c”的字符串
abc{2}	匹配在“ab”后面跟着两个“c”的字符串
abc{2,}	匹配在“ab”后面跟着两个或更多“c”的字符串
abc{2,5}	匹配在“ab”后面跟着2到5个“c”的字符串
a(bc)*	匹配在“a”后面跟着零个或更多“bc”序列的字符串
a(bc){2,5}	匹配在“a”后面跟着2到5个“bc”序列的字符串

或运算符：|、[]

a(b c)	匹配在“a”后面跟着“b”或“c”的字符串 -> Try it! (https://regex101.com/r/c08lqs/3)
a[bc]	匹配在“a”后面跟着“b”或“c”的字符串

字符类：\d、\d、\s 和 .

\d	匹配数字型的单个字符 -> Try it! (https://regex101.com/r/c08lqs/4)
\w	匹配单个词字（字母加下划线） -> Try it! (https://regex101.com/r/c08lqs/4)
\s	匹配单个空格字符（包括制表符和换行符）
.	匹配任意字符 -> Try it! (https://regex101.com/r/c08lqs/5)

使用「.」运算符需要非常小心，因为常见类或排除型字符类都要更快与精确。\\d、\\w 和 \\s 同样有它们各自的排除型字符类，即 \\D、\\W 和 \\S。例如 \\D 将执行与 \\d 完全相反的匹配方法：

\\D	匹配单个非数字型的字符 -> Try it! (https://regex101.com/r/c08lqs/6)
-----	--

为了正确地匹配，我们必须使用转义符反斜杠「\\」定义我们需要匹配的符号「^\\.(\$|)|*+?{\\}」，因为我们可能认为这些符号在原文本中有特殊的含义。

\\\$\\d	匹配在单个数字前有符号“\$”的字符串 -> Try it! (https://regex101.com/r/c08lqs/9)
---------	--

注意我们同样能匹配 non-printable 字符，例如 Tab 符「\\t」、换行符「\\n」和回车符「\\r」

Flags

我们已经了解如何构建正则表达式，但仍然遗漏了一个非常基础的概念：flags。

正则表达式通常以/abc/这种形式出现，其中搜索模式由两个反斜杠「/」分离。而在模式的结尾，我们通常可以指定以下 flag 配置或它们的组合：

- g (global) 在第一次完成匹配后并不会返回结果，它会继续搜索剩下的文本。
- m (multi line) 允许使用^和\$匹配一行的开始和结尾，而不是整个序列。
- i (insensitive) 令整个表达式不区分大小写（例如/aBc/i 将匹配 AbC）。

中级语句

分组和捕获：()

a(bc)	圆括弧会创建一个捕获性分组，它会捕获匹配项“bc” -> Try it! (https://regex101.com)
a(?:bc)*	使用 “?:” 会使捕获分组失效，只需要匹配前面的“a” -> Try it! (https://regex101.com)
a(?<foo>bc)	使用 “?<foo>” 会为分组配置一个名称 -> Try it! (https://regex101.com/r/c08lqs/)

捕获性圆括号 () 和非捕获性圆括弧 (?:) 对于从字符串或数据中抽取信息非常重要，我们可以使用 Python 等不同的编程语言实现这一功能。从多个分组中捕获的多个匹配项将以经典的数组形式展示：我们可以使用匹配结果的索引访问它们的值。

如果需要为分组添加名称（使用 (?<foo>...)），我们就能如字典那样使用匹配结果检索分组的值，其中字典的键为分组的名称。

方括弧表达式：[]

[abc]	匹配带有一个“a”、“ab”或“ac”的字符串 -> 与 a b c 一样 -> Try it! (https://regex101.com)
[a-c]	匹配带有一个“a”、“ab”或“ac”的字符串 -> 与 a b c 一样
[a-fA-F0-9]	匹配一个代表16进制数字的字符串，不区分大小写 -> Try it! (https://regex101.com)

`[0-9]%`

匹配在%符号前面带有0到9这几个字符的字符串

`[^a-zA-Z]`匹配不带a到z或A到Z的字符串，其中^为否定表达式 -> Try it! (<https://regex101.com>)

记住在方括弧内，所有特殊字符（包括反斜杠\）都会失去它们应有的意义。

Greedy 和 Lazy 匹配

数量符（* + {}）是一种贪心运算符，所以它们会遍历给定的文本，并尽可能匹配。例如，`<.+>` 可以匹配文本「This is a <div> simple div</div> test」中的「<div>simple div</div>」。为了仅捕获 div 标签，我们需要使用「?」令贪心搜索变得 Lazy 一点：

`<.+?>`一次或多次匹配 “<” 和 “>” 里面的任何字符，可按需扩展 -> Try it! (<https://regex101.com>)

注意更好的解决方案应该需要避免使用「.」，这有利于实现更严格的正则表达式：

`<[^\>]+>`一次或多次匹配 “<” 和 “>” 里面的任何字符，除去 “<” 或 “>” 字符 -> Try it! (<https://regex101.com>)

高级语句

边界符：\b 和 \B

`\babc\b`执行整词匹配搜索 -> Try it! (<https://regex101.com/r/c08lqs/25>)

`\b` 如插入符号那样表示一个锚点（它与\$和^相同）来匹配位置，其中一边是一个单词符号（如\w），另一边不是单词符号（例如它可能是字符串的起始点或空格符号）。

它同样能表达相反的非单词边界「\B」，它会匹配「\b」不会匹配的位置，如果我们希望找到被单词字符环绕的搜索模式，就可以使用它。

`\Babc\B`只要是被单词字符环绕的模式就会匹配 -> Try it! (<https://regex101.com/r/c08lqs/25>)

前向匹配和后向匹配：(?=) 和 (?<=)


`d(?=r)` 只有在后面跟着“r”的时候才匹配“d”，但是“r”并不会成为整个正则表达式匹配的一部分 -> Try
`(?<=r)d` 只有在前面跟着“r”时才匹配“d”，但是“r”并不会成为整个正则表达式匹配的一部分 -> Try it

我们同样能使用否定运算符：

`d(?!r)` 只有在后面不跟着“r”的时候才匹配“d”，但是“r”并不会成为整个正则表达式匹配的一部分 -> T
`(?<!r)d` 只有在前面不跟着“r”时才匹配“d”，但是“r”并不会成为整个正则表达式匹配的一部分 * *->* **

结语

正如上文所示，正则表达式的应用领域非常广，很可能各位读者在开发的过程中已经遇到了它，下面是正则表达式常用的领域：

- 数据验证，例如检查时间字符串是否符合格式；
- 数据抓取，以特定顺序抓取包含特定文本或内容的网页；
- 数据包装，将数据从某种原格式转换为另外一种格式；
- 字符串解析，例如捕获所拥有 URL 的 GET 参数，或捕获一组圆括弧内的文本；
- 字符串替代，将字符串中的某个字符替换为其它字符。 

原文链接：<https://medium.com/factory-mind/regex-tutorial-a-simple-cheatsheet-by-examples-649dc1c3f285>

本文为机器之心编译，转载请联系本公众号获得授权。

✂️-----

加入机器之心（全职记者 / 实习生）：hr@jiqizhixin.com

投稿或寻求报道：content@jiqizhixin.com

广告 & 商务合作：bd@jiqizhixin.com

