第153封信 | Google面试题——如何实现拼写纠错功能和拼写提示功能

吴军

第153封信 | Google面试题—...
11:52 5.56MB

信件朗读者：宝木

小师弟，你好！

今天我们通过Google的一道面试题，回顾一下这一周的内容，也复习一下之前的一些内容。这道面试题是这么出的：怎样实现拼写纠错的功能?

在讲解这道题目之前，我还是要强调一下这是一道没有标准答案的开放式问题，只有好的和不好的答案。什么是不好的答案呢？见过比较奇葩的回答是，到Google上搜索一下即可。这些面试者也不想想，他们面试的就是Google公司，没有工程师开发这个功能，哪里来的拼写纠错?

既然是开放式问题，就会有不同的答案。在介绍如何纠错之前，先讲一个简单的问题，如何查错。注意这是两个相关的，但并不相同的问题。查错比较简单，只要告诉书写者所写的单词是对了还是错了即可；纠错就比较难了，不仅要判断对错，还要提示正确的写法。比如你写"华盛顿"这个词，写成了Wasingdon，查一下字典很容易判断是否有这个单词，但是，要找到正确的拼写Washington，就颇费周章了。

对于查错这个问题，最直截了当的回答是把字典中所有英文单词在计算机中建立一个词典，大家无论是使用word写文档，还是在Google提供的搜索中，在相应的软件里实现一个直接查字典的功能就好了。不过，字典怎么设计，就有讲究了。

第一种设计方法是采用排好序的线性数组，这样采用二分查找，就可以判断一个字符串的拼写是否是正确的（在构词学上称为"合法"）。如果英语的单词有25万个，那么做18次单词的比对即可（2的18次方大约等于25万）。如果再考虑到单词的平均长度是6个字母，每次比对要进行六次，这样一共就是108次比对。

当然，如果你还记得我们前面讲到的哈希表和哈希查找，我们利用哈希表来存储词典，大约只要进行两次单词查找，就可以判断一个单词的拼写是否合法。这样只要比对字母12次左右。

第二种设计方法是使用一种类似于我们前面讲过的二叉树的树状结构。回顾一下第78封来信的内容，在二叉树中，一个根节点，有左右两个子节点，或者说左右两个枝杈。在字典使用的这种树中，每个根节点下面有26个枝杈，因为英语的字母表里有26个字母。当然，如果做德语的拼写检查就要有30个枝杈，做西班牙语的，就要有27个或者32个（西班牙语有5个双字母），如果做一种适合所有语言的，分岔就更多了，但只要不超过256个，计算机实现起来都不困难。

采用树状检查拼写，办法和我们人查字典差不多。先看第一个字母，假设是w，然后看第二个，发现是a，就沿着a的那个树杈往下走，如果检查"华盛顿"这个词，到了Wasing就查不下去了，就知道有错了。

采用树状的数据结构有两个好处。首先就是速度快，一个要检查的单词有多少字母，只要查找那么多次即可，比如平均只需要六次。当然，程序实现起来要比使用数组或者哈希表麻烦。但是这个结构一旦建成，可以做一件更有意义的事情，就是提示拼写，因为当敲了一个单词的前几个字母后，根据树状查找，很容易被开头那几个字母的常用单词提示出来，这个功能特别有用。这是第二个好处。

当然，如果在特定的场合，比如手机上，为了节省存储空间，还可以用一种叫做"布隆过滤器"的算法压缩存储。总之，这样一个看似答案很直截了当的问题，可以有很多延伸的空间。好的面试官可以不断追问下去，而好的候选人需要脑子非常灵活，对所学的东西活学活用。

接下来再谈谈如何纠错，这并非是个很容易的问题，我们需要从理论和工程两个角度来回答。先从理论上回答。

当我们知道一个单词写错了，我们如何判断它所对应的正确拼写可能是哪一种呢？一般来讲我们要看错误的拼写和哪个正确的拼写"相近"。但是，这个相近其实并不好度量。比如"华盛顿"的错误拼写Wasingdon和正确的拼写Washington是否相近呢？我们感觉它们确实相近，但是这个错误拼写其实和另一个英国人的名字Watingdon似乎更相近。

因此，有必要量化地度量一下这些单词之间的相近程度，或者反过来说，差异程度。要做这件事，就要定义一种差异程度的度量，在构词学中，它被称为"编辑距离"。接下来我们就说说什么是编辑距离（Editing Distance），它是如何计算的。

对比一下Wasingdon和Washington。我们刻板地把相同位置的字母一个一个地比较，那么它就错了六个或者七个字母，因为从Was之后所有的字母都错位了。下图一致的字母用蓝色表示，后面错掉的用红色表示。

得到喜马拉雅知识共享 防断更微信：699250

W	a	s	i	n	g	d	o	n	
W	a	s	h	i	n	g	t	o	n

但是，我们一般会认为Wasingdon只错了两个字母，即少了一个h，把t写成了d。如果我把缺少掉的字母h用空格"－"表示，让它和h对应，那么后面的字母大多能够对应起来。大家可以扫一眼下面的表格，我们依然用蓝色表示正确的匹配，红色表示错误的匹配。从这个表中你可以看出，标红色的错误只有两处。

W	a	s	–	i	n	g	d	o	n
W	a	s	h	i	n	g	t	o	n

接下来的问题是，上述两个拼写的编辑距离到底应该如何定义?

很清楚，应该用最小的差异来定义。也就是说，编辑距离是2，而不是6。那么接下来怎么计算编辑距离呢？这就要用到和我们前天说的维特比算法相类似的一种动态规划算法了。这个算法的细节我们省略了，它的思想和维特比算法一样，把一个指数复杂度的问题，变成一个平方复杂度的问题。具体到"华盛顿"这个单词上，大约可以把18万次计算，缩减到100次计算。

下面一个表格是采用动态规划的方法，计算出的"华盛顿"正确拼法和错误拼法不同的对应方式字母之间的差异。表中横线上的是错误拼法，竖线上的是正确拼法。红色的是差异最小的对应方法，即用一个空格"－"对应丢失的h，用d对应t。

	W	a	s	i	n	g	d	o	n
W	0	1	2	3	4	5	6	7	8
a	1	0	1	2	3	4	5	6	7
s	2	1	0	1	2	3	4	5	6
h	3	2	1	1	2	3	4	5	7
i	4	3	2	1	2	3	4	5	6
n	5	4	3	2	1	2	3	4	5
g	6	5	4	3	2	1	2	3	4
t	7	6	5	4	3	2	2	3	4
o	8	7	6	5	4	3	3	2	3
n	9	8	7	6	5	4	4	3	2

对于所有的英语单词，我们都可以两两找出它们的编辑距离。比如a和an差1，but和cut差1，tell和tail差2，等等。因此，进行拼写纠错的第一步，就是对于那些在词典里没有的单词，找到和它们编辑距离比较小，比如相差只有一到两个字母的单词作为候选。对于上面例子中Wasingdon这个写法，"华盛顿"的正确写法Washington和英国人的名字Watingdon都可以作为候选。

接下来到底挑选哪个单词来修正就有讲究了。虽然前者相差两个字母，后者只相差一个字母，看似后者更接近，但是，如果考虑到"华盛顿"这个词本身出现的概率要比Watingdon高得多，通常似乎改正成为"Watingdo"更合理。

不过，凡事也有例外，如果上下文是讨论1066年威廉征服这件事，还真有一个叫做Watingdon的家族跟随威廉一世来到英国。因此，正确使用上下文很重要。事实上，稍微好一点的拼写矫正都需要考虑上下文。而利用上下文做选择，其实采用的还真是维特比算法。

如果一个候选人能够将上面的内容说清楚，那么这道题可以说回答得比较好了。但是到此为止也才解决了科学上的问题，如果面试官有时间，他还可以往工程上考核候选人。比如说下面这样一些：

1. 在利用上下文对可能正确的拼写做选择时，如何建立一个描述上下文特性的统计模型?

2. 很多时候，拼写正确与否是和语言相关的。比如照片photo这个词，在德语中的写法是foto。如果在英语中看到foto，可以认为是错别字，但是德国人有时也用photo。因此这个在德语中虽然是错误拼写，在非正式的书写中，未必需要纠正它。此外，英国人把颜色写成colour，美国人中间少一个u，写成color，这些在工程上都要注意。

3. 我们在前面定义编辑距离的时候，假定了任何两个字母如果不一样，都算错一个，