query改写-拼写纠错（Spelling Correction）

这是转载斯坦福NLP课程中讲拼写纠错的一节，感觉讲的比较好，虽然课程很老，但是讲的知识，在目前的query改写、拼写纠错还是很实用的，原文链接在文章开头。

# 1、任务定义

拼写纠错（Spelling Correction），又称拼写检查（Spelling Checker），往往被用于字处理软件、输入法和搜索引擎中，如下所示：

在这里插入图片描述

在这里插入图片描述在这里插入图片描述

拼写纠错一般可以拆分成两个子任务：

Spelling Error Detection：按照错误类型不同，分为Non-word Errors和Real-word Errors。前者指那些拼写错误后的词本身就不合法，如错误的将“giraffe”写成“graffe”；后者指那些拼写错误后的词仍然是合法的情况，如将“there”错误拼写为“three”（形近），将“peace”错误拼写为“piece”（同音），将“two”错误拼写为“too”（同音）。

Spelling Error Correction：自动纠错，如把“hte”自动校正为“the”，或者给出一个最可能的拼写建议，甚至一个拼写建议列表。

Non-word拼写错误

Spelling error detection：任何不被词典所包含的word均被当作spelling error，识别准确率依赖词典的规模和质量。

Spelling error correction：查找词典中与error最近似的word，常见的方法有Shortest weighted edit distance和Highest noisy channel probability。

Real-word拼写错误

Spelling error detection：每个word都作为spelling error candidate。

Spelling error correction：从发音和拼写等角度，查找与word最近似的words集合作为拼写建议，常见的方法有Highest noisy channel probability和Classifier。

基于Noisy Channel Model的拼写纠错

Noisy Channel Model即噪声信道模型，或称信源信道模型，这是一个普适性的模型，被用于语音识别、拼写纠错、机器翻译、中文分词、词性标注、音字转换等众多应用领域。其形式很简单，如下图所示：

在这里插入图片描述

噪声信道试图通过带噪声的输出信号恢复输入信号，形式化定义为：

在这里插入图片描述

应用于拼写纠错任务的流程如下：

在这里插入图片描述

noisy word（即splling error）被看作original word通过noisy channel转换得到，现在已知noisy word（用x表示）如何求得最大可能的original word（用w表示），公式如下：

在这里插入图片描述

P(w)为先验概率，P(x|w)为转移概率，二者可以基于训练语料库建立语言模型和转移矩阵（又称error model，channel model）得到。

下面通过一个Non-word spelling error correction的例子加以解释：

给定拼写错误“acress”，首先通过词典匹配容易确定为“Non-word spelling error”；然后通过计算最小编辑距离获取最相似的candidate correction，需要特别说明的是，这里的最小编辑距离涉及四种操作：

在这里插入图片描述在这里插入图片描述

据统计，80%的拼写错误编辑距离为1，几乎所有的拼写错误编辑距离小于等于2，基于此，可以减少大量不必要的计算。

通过计算最小编辑距离获取拼写建议候选集（candidate w），此时，我们希望选择概率最大的w作为最终的拼写建议，基于噪声信道模型思想，需要进一步计算P(w)和P(x|w)。

通过对语料库计数、平滑等处理可以很容易建立语言模型，即可得到P(w)，如下表所示，计算Unigram Prior Probability（word总数：404,253,213）

在这里插入图片描述

接下来，可以基于大量<misspelled word x=x1 x2 x3 … xm, correct word w=w1 w2 w3 … wn>pair计算del、ins、sub和trans四种转移矩阵，然后求得转移概率P(x|w):

在这里插入图片描述在这里插入图片描述

计算P(“acress”|w)如下：

在这里插入图片描述

计算P(w)P(“acress”|w)如下：

在这里插入图片描述

“across”相比其他candidate可能性更大。

上面建立语言模型时采用了unigram，也可以推广到bigram，甚至更高阶，以较好的融入上下文信息。

如句子“a stellar and versatile acress whose combination of sass and glamour…”，计算bigram为：

P(actress|versatile)=.000021 P(whose|actress) = .0010

P(across|versatile) =.000021 P(whose|across) = .000006

则联合概率为：

P(“versatile actress whose”) = .000021\*.0010 = 210 x10-10

P(“versatile across whose”) = .000021\*.000006 = 1 x10-10

“actress”相比“across”可能性更大

Real-word拼写纠错

Kukich（1992）指出有25%~40%的拼写错误都属于Real-word类型，与Non-word类型相比，纠错难度更大，因为句子中的每个word都被当作待纠错对象。通常，解决方法分两步：

在这里插入图片描述

例如，给定句子S=w1,w2,w3,…,wn，为每个wi生成candidate set，如下：

Candidate(w1) = {w1, w’1 , w’’1 , w’’’1 ,…}

Candidate(w2) = {w2, w’2 , w’’2 , w’’’2 ,…}

… …

Candidate(wn) = {wn, w’n , w’’n , w’’’n ,…}

最后，选择概率最大的序列W为自动纠错后的句子，与中文分词、音字转换等应用相同，可以表示成词网格形式，转化为HMM的解码过程：

在这里插入图片描述

为了简化起见，一般规定一个句子中最多有一个word存在splling error（事实上，所出现的情况也的确如此）。

应用

实际的拼写纠错系统一般会遵守如下HCI（Human Computer Interface）准则：

在这里插入图片描述

根据应用场景不同（Domain Sensitivity），需要对语言模型进行特别的处理，如：

在这里插入图片描述

除了字面上的拼写错误，还有可能同音导致，所以，有些系统将“error model”转化为“Phonetic error model”解决拼写纠错问题。

另外，键盘上临近的按键更容易引入spelling error pair，据此可以对转移矩阵进行加权。在这里插入图片描述

我们还可以将拼写纠错问题转化为分类问题，通过构建训练语料库，抽取features，训练分类模型，预测新实例等一系列过程解决，如下：

在这里插入图片描述