检测中文错别字

一、概述

输入一个中文文本，模型自动发现并改正句子中存在的错别字. 根据错误的来源，可以将纠错划分为错字和别字。错字可理解为一个字或者一个词组不在字典中出现的字或词，别字可理解为不符合上下文语境的字或词。本方案简单的描述了使用 n gram，smt，svm 和 rnn 等模型识别文本中的错别字。

二、数据准备

1.混淆集，是包括 5,401 的中文字符集，具有相似形状，相同和相似发音的普通汉字

主要分为五类：（1）相同的拼音和相同的音调；（2）相同的拼音和不同的音调；（3）不同的拼音和相同音调；（4）不同的拼音和不同的音调；（5）相似的形状

http://ir.itc.ntnu.edu.tw/lre/sighan7csc.html

2. 字典。 CC-CEDICT，由 Creative 发布的免费繁体中文词典 ，总计 包括大约 71,886 个中文单词，其中包含两个或更多中文 字符

https://www.mdbg.net/chinese/dictionary?page=cedict

3. 中文词语搭配库，

http://www.sogou.com/labs/resource/r.php

4. 基于百度百科和维基百科的 word n gram

https://pan.baidu.com/s/1XEmP_OFkQwOjipCjI2OPEw

https://pan.baidu.com/s/1ZKePwxwsDdzNrfkc6WKdGQ

5. 生成的人工数据。因训练的规模数据不足以估计用于训练 SMT 模型的误差模型。从训练中产生了大约 200 万个句子通过用候选人替换提供的 700 个句子中的每个字符来获取数据模型选择

三、模型选择

1、语言模型

在中文错别字查错情景中，判断一个句子是否合法可以通过计算它的概率来得到，基于马尔科夫模型假设，一个词的出现概率仅依赖于该词的前 1 个词或前几个词，则有一个词的出现仅依赖于前 1 个词，即 Bigram（2-gram）：

$$P(s) \approx P(w_1)*P(w_2|w_1)*\ldots*P(w_n|w_{n\text{-}1})$$

一个词的出现仅依赖于前 2 个词，即 Trigram（3-gram）：

$$P(s) \approx P(w_1)*P(w_2|w_1)*P(w_3|w_1w_2)\ldots*P(w_n|w_{n\text{-}2}w_{n\text{-}1})$$

一个词的出现仅依赖于前 3 个词，即 4-gram：

$$P(s) \approx P(w_1)*P(w_2|w_1)*P(w_3|w_1w_2)*P(w_4|w_1w_2w_3)\ldots*P(w_n|w_{n\text{-}3}w_{n\text{-}2}w_{n\text{-}1})$$

n-gram 模型通过计算极大似然估计（Maximum Likelihood Estimate）构造语言模型，这是对训练数据的最佳估计，对于一个数据集，假设$Count(w_i)$,

$$P(w_i|w_{i\text{-}1}) = Count(w_i, w_{i\text{-}1}) / Count(w_{i\text{-}1})$$

接下来只需要训练确定一个阀值，只要概率值≥阀值就认为句子合法。

为了避免数据溢出、提高性能，使用取 log 后使用加法运算替代乘法运算，

因矩阵存在 0 值，在语料库数据集中没有出现的词不能就简单地认为他们的概率为 0，采用拉普拉斯矩阵平滑，把 0 值改为 1 值，设置成该词对出现的概率极小，这样就比较合理。

当 n-gram 的 n 值越大时，对下一个词的约束力就越强，因为提供的信息越多，但同时模型就越复杂，问题越多，因此通过 bigram, trigram, 4-gram 结合，并对每个字的分数求平均以平滑每个字的得分，能得到更好的效果。

2、统计机器翻译模型

鉴于将错误视为源语言的句子，目标是找到最好的纠正句。因此，给出一个可能包含错误字符的句子 S. 在它作为源句，输出是目标语言中的句子 C 不同替换的最高概率 C. 结合使用贝叶斯规则

$$\hat{C} = argmaxp(S|C)*p(C)$$

这里，$p(S|C)$被称为错误模型，表示一个汉字写错了的概率;而$p(C)$是语言模型，它评估纠正的中文句子概率。与基于语言模型的候选生成模型不同，SMT 模型检测并通过结合错误模型和语言来纠正拼写错误模型。如果有一个大的训练语料库来估计一个更好的错误模型，那么一个汉字被错误写成的可能性，可以获得更好的结果。但是，由于错误模型的训练数据的稀缺性，很难估计误差模型的真实参数。为了解决这个问题，通过替换提供的每个字符生成 200 万个人工训练数据。训练集中有 700 个句子，混淆集用于替换生成人工训练数据。

3、支持向量机

中文拼写错误检测任务的目标是检测给定句子中是否存在任何错误，我们可以将其视为一系列错误的二分类问题：如果当前字符是拼写错误，则标签为 0，否则标签为 1.

将原训练数据进行人工标签之后，由语言模型和机器翻译模型分别预测可能单词的概率分数。使用 SVM 的置信度能确定当前字符出现拼写错误的可能性。通过合并原始输入字符和先前模型的错误可能性字词，系统创建输入文本中每个字符的可能性词组列表。然后列表中的所有词组将基于 SVM 分类器计算的置信度得分进行排名。最后具有最高置信度分数的词组将被视为正确的词组。 然后合并输入字符和预测可能性词组并创建包含两个字符的列表。最后，根据它们进行排名在 SVM 分类器计算的置信度分数上，选择可能行词组得分最高的输出。

4、RNN

基于文本的错别字纠错可以更为方便的获取上下文信息，这为文本纠错提供了更大的空间, 文本端的纠错应更多的基于上下文信息，而不是简单的字音字形,为了实现基于文本的错别字识别，两组 RNN 分别构成 encoder 和 decoder，作为 encoder 的 RNN 学习句子语意表示。RNN 由一系列相同的网络构成，上一个词语的向量表示作为计算下一个网络的输入，如此循环。整个句子每个词计算完成， 便得到了一个句子的语意向量。Seq2seq 模型使用 RNN（此时被称为 encoder）将输入句子表示为一个向量， 再使用另一个 RNN（此时被称为 decoder）解码这个向量获取输出。为了充分获取上下文的语意信息，初始模型在基本 seq2seq 模型的基础上增 加了 bi-direction 以及 attention 机制。在初始模型的基础上，考虑到纠错任务中前后两个词语的 attention 值之间存 在比较强的关联。如 decoder 中第一个词语 attend 到 encoder 中第一个词语，则 decoder 中第二个词语应该 attend 到 encoder 中的第二个词语。

四、 测评方法

1、困惑度是评估 n-gram 模型的重要指标之一。困惑度是定义为统计语言模型的熵。由于熵定义为概率分布 P 相乘的总和,困惑度被定义为

$$H_p = -\frac{1}{i}\log P\ (w_1 w_2 \dots w_i),$$

困惑度越低，语言模型越好。

2、 在中文错别字识别任务中，需要对测试数据集进行人工标签，对对比模型生成的数据进行评估指标数据，包括错别字的句子误检数，句子正确检查数等。 以下几个指标可以用于错别字句子级别度量绩效评估，定义为

$$\text{错误预警率} = \frac{\text{句子误检数}}{\text{正确句子}}$$

$$\text{检测正确率} = \frac{\text{句子正确检查数}}{\text{所有句子}}$$

$$\text{检测精确率} = \frac{\text{句子正确检查数}}{\text{正确句子} + \text{被错误检查成正确的错误句子}}$$

$$\text{检测召回率} = \frac{\text{句子正确检查数}}{\text{错误句子}}$$

$$\text{检测 f1} = \frac{2*\text{检查精确率}*\text{检测召回率}}{\text{检测精确率} + \text{检测召回率}}$$

$$\text{错误位置准确率} = \frac{\text{正确检查出错误位置的句子}}{\text{所有句子}}$$

$$\text{错误位置精确率} = \frac{\text{正确检查出错误位置的句子}}{\text{正确句子} + \text{被错误检查成正确的错误句子}}$$

3、gleu 分

用于评估每个句子的流畅度指标。如果识别错别字之后句子的流畅度有提升，则是有效的错别字识别的概率比较大，可用 nltk 的 gleu score 等第三方包来实现。