知乎 首发于 机器学习小知识

中文文本纠错算法--错别字纠正的二三事

mountai...
机器学习爱好者

关注他

123 人赞同了该文章

本文首先介绍一下：

1）错别字的类型有哪些

2）错别字纠正的关键技术和关键点

3）简要介绍我们项目中采用的文本纠错框架

4）介绍错别字项目的个人体会

5）几个现成的工具包，百度nlp平台最近也推出了文本纠错模块，处于内测中，所以没有进行比较。如果有小伙伴能告知比较好的中文文本纠错package，欢迎留言讨论！感谢！

2. 人名，地名错误：哈蜜（正：哈密）

3. 拼音错误：咳数（ke shu）—> ke sou,

4. 知识性错误：广州黄浦（埔）

5. 用户发音、方言纠错：我系东北滴黑社会，俚蛾几现在在我手上。（我是东北的黑社会，你儿子现在在我手上。）

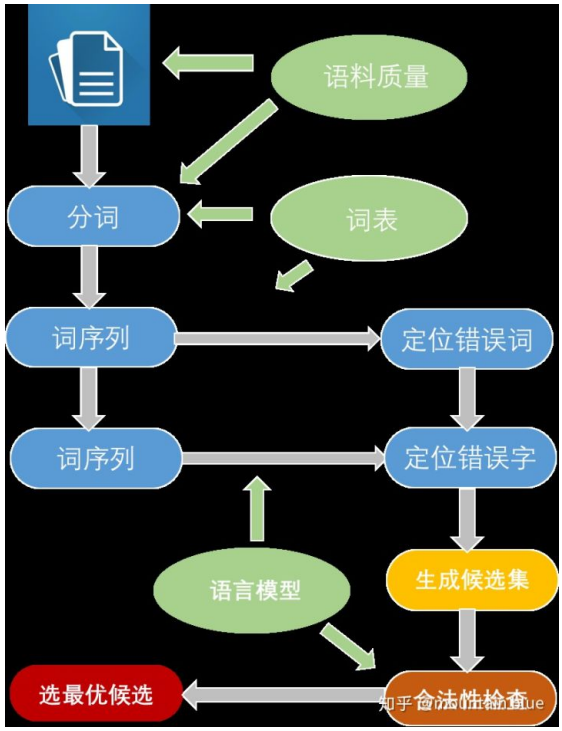
6. 重复性错误：在 上 上面 上面 那 什么 啊

7. 口语化问题：呃 。 呃 ，啊，那用户名称是叫什么呢？（正：那用户名称是叫什么呢？）

错别字纠正的主要技术：

- 错别字词典，编辑距离，语言模型（ngram LM，DNN LM，基于字的模型？基于词的模型？）
- 三个关键点：分词质量、领域相关词表质量、语言模型的种类和质量

在最近的项目中，我们采用了pycorrector的纠错逻辑，如下图所示：



写在前面：

1. 效果：现有错别字纠正package大部分是通用领域的错别字检查，缺乏统一的评判标准，效果参差不齐。长句效果差，短句、单词效果好一些，未来应用到产品中，也要根据标点符号截成短句，再进行错别字检查。

2. 口语化、重复性的问题，所有package不能解决此类问题。

3. **误判率的问题！！！**错别字纠正功能有可能把正确的句子改成错误的。。这就要求，正确率x要远大于误判率y。假设有m个问题，其中2%是有错别字的，m*2%*x>m*(1-2%)*y，根据个人的经验，误判率y是可以控制在1%以下的，如果有比较好的词表，可以控制在0.5%以下。根据上述不等式，误判率控制在0.5%以下，正确率达到24.5%就能满足上述不等式。

4. 项目中，若测试数据不含重复错别字样本（错别字：帐单，其中的帐这个错别字只出现过一次），错别字纠正的正确率达到了50%，误判率0.49%左右。若包含重复样本，正确率达到了70%以上。

后面这三点比较关键：

5. 项目中使用了基于n-gram语言模型，使用kenLM训练得到的，DNN LM和n-gram LM各有优缺点，这里卖个关子，感兴趣的可以思考一下二者区别。另外，基于字的语言模型，误判率会较高；基于词的语言模型，误判率会低一些（符合我个人的判断，在我的实验里情况也确实如此）。

6. 训练语言模型的语料中并不clean，包含了很多错别字，这会提高误判率。使用更干净的语料有助于降低误判率，提高正确率。

7. 专业相关词表很关键，没有高质量的词表，很多字也会被误认为是错别字，所以也会提高误判率。

测试样本：

'感**帽**了','你儿**字**今年几岁了', '少先队员**因**该为老人让**坐**','**随**然今天很热','传**然**给我','呕**土**不止','哈**蜜**瓜','广州黄**浦**','在 上 上面 上面 那 什么 啊','呃 。 呃 ,啊,那用户名称是叫什么呢？ ', '我生病了,咳**数**了好几天', '对京东**新人**度大打折扣','我想买**哥**苹果手机'

效果评价简介：

a. 单词、短句效果：一共13个测试样本，9/13表示13个样本中，纠正了9个错误。（长句效果差，没有考虑）

其中, with print 表示该纠错方法的用时包含了“输出到terminal的时间”， without print表示该纠错方法的用时没有包含“输出到terminal的时间”。这么划分的原因是 输出到terminal比较耗时，部分package可以选择输出or不输出。

c. 可扩展性：主要（1）考虑该纠错方法是否包含 自定义的错别字词典，方便个性化定制；（2）考虑该纠错方法，是否提供模型代码方便，在小娜的文本上进行训练语言模型。

几个现成的工具包：

1. github.com/shibing624/p... pycorrector

简介：考虑了音似、形似错字（或变体字）纠正，可用于中文拼音、笔画输入法的错误纠正，能够给出给出出错位置。

语言模型：

- Kenlm（统计语言模型工具）
- RNNLM（TensorFlow、PaddlePaddle均有实现栈式双向LSTM的语言模型）

代码：

```
import pycorrector

corrected_sent, detail = pycorrector.correct('少先队员因该为老人让坐')

print(corrected_sent, detail)
```

单词、短句效果：9/13 效果尚可

速度：0.366050 all， 0.028157692 avg ；

可扩展性：词典可扩展，可使用自己的语料进行训练，该repo使用的是人民日报数据。扩展性强。

测试样本效果：'感帽了','你儿字今年几岁了', '少先队员因该为老人让坐','随然今天很热','传然给我' '呕土不止' '哈蜜瓜' '广州黄浦' '在 上 上面 上面 那 什么 啊' '呃 呃 啊 那用户名称是叫什么

2. https://github.com/ischeng/Correction 10 months ago

简介：

- 使用语言模型计算句子或序列的合理性
- 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
- 根据Median Absolute Deviation算出outlier分数，并结合jieba分词结果确定需要修改的范围
- 根据形近字、音近字构成的混淆集合列出候选字，并对需要修改的范围逐字改正
- 句子中的错误会使分词结果更加细碎，结合替换字之后的分词结果确定需要改正的字
- 探测句末语气词，如有错误直接改正

特点：

训练的语言模型很多，根据介绍看，整体比较完善，看起来高大上。不过code跑不起来，作者没回应—--后面再改一下作者代码，看看能否跑起来。

3. github.com/PengheLiu/Cn... 2 years ago

简介：

针对医学数据训练出来的，基于编辑距离，可自行训练--效果一般，统计词频和共现信息，不太完善，返回大量candidates

特点：

- 人们通常越往后字打错的可能越大，因而可以考虑每个字在单词中的位置给予一定权重，这中方法有助于改进上面的第一种“传然” - "虽然"的情况；
- 考虑拼音的重要性，对汉语来讲，通常人们打错时拼音是拼对的，只是选择时候选择错了，因而对候选词也可以优先选择同拼音的字。

单词、短句效果：1/13 效果差,因为训练语料是医学文章

速度：None

可扩展性：词典+模型。扩展性还可以。

测试样本效果：'感帽了','你儿字今年几岁了', '少先队员因该为老人让坐','随然今天很热','传然给

词频字典+bi-gram

github.com/apanly/proof...

模型比较老旧，不考虑

5. github.com/taozhijiang/... 3 years ago

京东客服机器人语料做的中文纠错--更接近我们的应用场景，主要解决同音自动纠错问题，比如：

对京东**新人度**大打折扣 -- > 对京东**信任度**大打折扣

我想买**哥**苹果手机 纠正句:我想买**个**苹果手机

但代码多年未更新，目前跑不起来。

6. github.com/beyondacm/Au... 9 months ago

original sentence:感帽，随然，传然，呕土

corrected sentence:感冒，虽然，传染，呕吐

original sentence:对京东新人度大打折扣，我想买哥苹果手机

corrected sentence:对京东**新人度**大打折扣，我国买卖苹果手机

单词、短句效果：5/13 效果差

速度：2.860311 all， 0.220023 avg； with print

我','呕土不止','哈蜜瓜','广州黄浦','在 上 上面 上面 那 什么 啊','呃 。 呃 ,啊,那用户名称是叫什么呢？', '我生病了,咳数了好几天', '对京东新人度大打折扣','我想买哥苹果手机'

7. github.com/SeanLee97/xm... 3-4 months ago

nlp工具包，包含分词、情感分析，没有专注于错别字纠正，效果较差

单词、短句效果：3/13 效果差

速度：2.860311 all， 0.220023 avg； without print: 0:00:00.000017 all

可扩展性：既没发现词典、也没发现模型。扩展性较差。

测试样本效果：'感帽了','你儿字今年几岁了', '少先队员因该为老人让坐','随然今天很热','传然给我','呕土不止','哈蜜瓜','广州黄浦','在 上 上面 上面 那 什么 啊','呃 。 呃 ,啊,那用户名称是叫什么呢？', '我生病了,咳数了好几天', '对京东新人度大打折扣','我想买哥苹果手机'

项目做的比较急，调研的package不多，如果有更好的方案，求告知，谢谢啦！

如果觉得文章对您有帮助，可以关注本人的微信公众号：机器学习小知识

机器学习小知识

微信扫描二维码，关注我的公众号