语言模型在纠错任务中的应用及kenlm的安装和使用

笔记本:	我的第一个笔记本		
创建时间:	2019/1/9 14:55	**更新时间:**	2019/1/11 13:55
作者:	wangjiaqi_ys@163.com		
URL:	https://github.com/kpu/kenlm		

一、主题:
1.语言模型
2.kenlm的安装和使用

二、中文文本纠错，常见的错误类型
1.人名错误：陆虞候---陆虞侯
1.别字：感帽，随然，传然，呕土
2.地名错误：广州黄浦（埔）
3.拼音错误：咳数（ke shu） -->ke sou
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）

三、中文文本纠错任的思路和方案:

思路:

错误识别(Error Detection)：检测错别字错误出现的位置，返回值一般为(word,begin_idx,end_idx,error_type)

错误纠正(Error Correction)：纠正错误

方案:

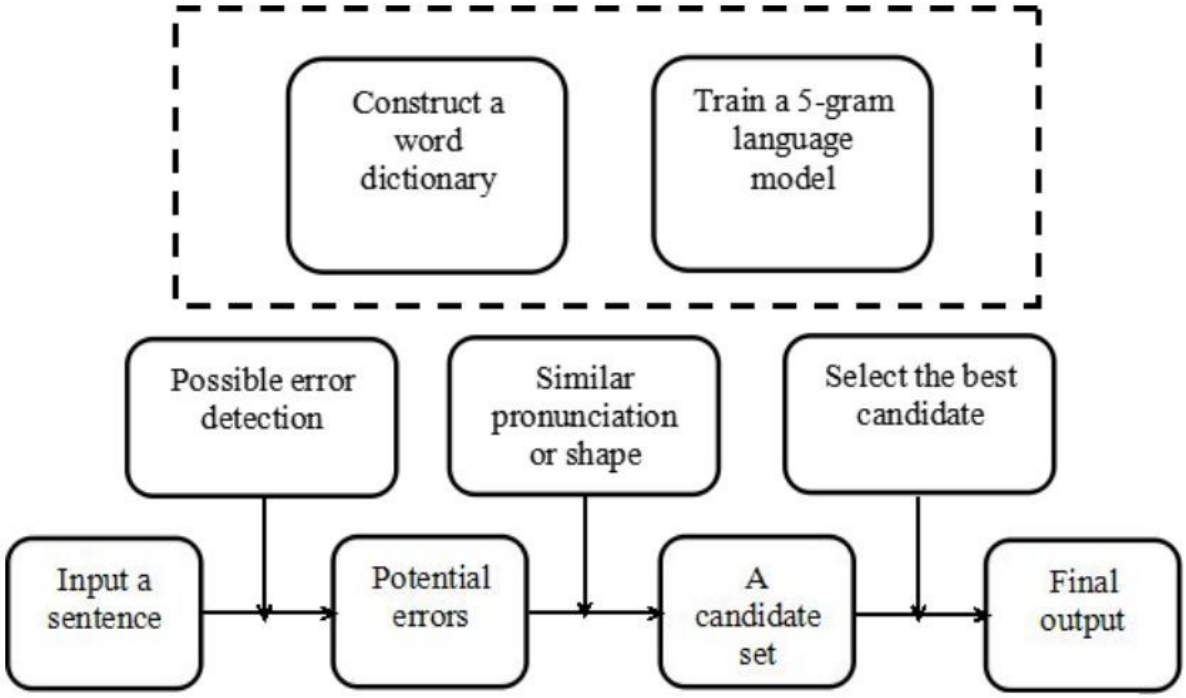
1.基于统计的方法：语言模型的检测和纠错

2.基于深度学习的方法:

rnn_attetion:英文文本纠错取得过第一名

sequence2sequence

四、基于语言模型纠错的算法流程:



1.切词：
例如：少先队员因该给老人让坐
tokens：[('少先队员', 0, 4), ('因该', 4, 6), ('为', 6, 7), ('老人', 7, 9), ('让', 9, 10), ('坐', 10, 11)]
2.检测疑似错误的字或词：
根据词频字典检测，如果不在文件word_freq.txt中，则认为该词是错误词的候选词
3.根据语言模型对句子打分：
少先队员因该给老人让坐
2-gram打分：[(' 少 先'),('先 队'),('队 员'),('员 因'),('因 该'),('该 给'),('给 老'),('老 人'),('人 让'),('让 坐')]
3-gram打分：[(' 少 先 队'),('先 队 员'),('队 员 因'),('员 因 该')....]
将两个得分取均值后，得到一个打分列表score_list
4.计算score_list的平均绝对离差，并获得疑似错别字的index
scores = np.array(score_list)
median = np.median(scores,axis=0)
margin_median = np.sqrt(np.sum((scores - median)**2,aixs=1))
med_abs_deviation = np.median(margin_median)
y_score = ratio * margin_median / med_abs_deviation#和平均离差相关的量
maybe_error_indices = np.where((y_score>threshold) & (scores < median))
5.经过步骤2和步骤4，可以得到：
[['因该',4,6,2],['坐',10,11,3]]
6.纠错：
遍历所有疑似错误位置，使用形似，音似字分别代替错误位置，然后放到语言模型中计算句子困惑度，困惑度越低，说明该词或字越适合这句话。

五、数据集：
1.同音/近音词：比如 '晴'

- 同音同调: 檠 晴 擎
- 同音异调: 青 轻 清 顷 请 庆 磬
- 近音同调: 擒 禽 噙 琴 勤 秦 芹
- 近音异调: 精 经 京 颈 井

2.形近字：清 晴 请 青 债 渍 啧 惆 悸 惟

3.词频字典：根据Wiki中文数据构建,格式：{'word1':count1,'word2':count2...}

六、kenlm工具使用：

1.说明：统计语言模型工比较好用的是srilm及kenlm，kenlm训练速度比srilm快很多，并支持单机大数据的训练

2.安装：

a.在ubuntu下安装，kenlm依赖于boost，xz，zlib，bzip，libbz2-dev，之后下载kenlm包

```
cd kenlm
mkdir build
cmake ..
make
python setup.py install
```

b.在windows下安装：

```
pip install https://github.com/kpu/kenlm/archive/master.zip
```

3.训练：

kenlm/build/bin/lmplz，使用生成arpa文件

```
bin/lmplz -o 3 --verbose_header --text ../text-18-03/text_18-03-AU.txt --arpa MyModel/log.arpa
```

```
root@wjq:/home/wjq/kenlm/build# bin/lmplz -o 2 --verbose_header </media/wjq/MyPrj1/corpus/reduce_zhiwiki.txt> /media/wjq/MyPrj1/corpus/wiki_2_gram.arpa
=== 1/5 Counting and sorting n-grams ===
Reading /media/wjq/MyPrj1/corpus/reduce_zhiwiki.txt
----5---10---15---20---25---30---35---40---45---50---55---60---65---70---75---80---85---90---95--100
**********************************
```

arpa文件转bin文件：`bin/build_binary -s log.arpa log.bin`，生成bin文件比.arpa文件小，方便调用时加载

4.使用：

```
import kenlm
model = kenlm.Model('data/kenlm/people_chars_lm.klm')
model_bin = kenlm.LanguageModel('data/kenlm/peolpe_chars_lm.bin')
#打分
score = model.score('少 先',bos=False,eos=False)#bos和eos表示不自动添加句首和句末标记符
#困惑度计算
model.perplexity('少 先 队 员 因 该 给 老 人 让 坐')#474.422
model.perplexity('少 先 队 员 因 该 给 老 人 让 座')#373.975
```

备注：

在信息论中，perplexity(困惑度)用来度量一个概率分布或概率模型预测样本的好坏程度。它也可以用来比较两个概率分布或概率模型。低困惑度的概率分布模型或概率模型能更好地预测样本，即困惑度越低，句子越通顺。

参考：

1.pycorrect:https://shibing624.github.io/pycorrector/
2.how to write a spelling correct:http://norvig.com/spell-correct.html
3.language model:http://www.aclweb.org/anthology/W/W14/W14-6835.pdf
4.困惑度：https://blog.csdn.net/jiaqiang_ruan/article/details/77989459
5.kenlm: https://github.com/kpu/kenlm
6.kenlm官网：https://kheafield.com/code/kenlm/
7.语言模型打分：https://blog.csdn.net/asrgreek/article/details/81979194