无监督中文文本纠错

本项目目标为无监督中文文本纠错, 下面是常用的一些基于语言模型的开源方法:

基于语言模型的纠错方法

- pycorrector[Github]
  - 中文纠错分为两步走，第一步是错误检测，第二步是错误纠正；
  - 错误检测部分先通过结巴中文分词器切词，由于句子中含有错别字，所以切词结果往往会有切分错误的情况，这样从字粒度和词粒度两方面检测错误， 整合这两种粒度的疑似错误结果，形成疑似错误位置候选集；
  - 错误纠正部分，是遍历所有的疑似错误位置，并使用音似、形似词典替换错误位置的词，然后通过语言模型计算句子困惑度，对所有候选集结果比较并排序，得到最优纠正词
- **correction[Github]**
  - 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
- **Cn_Speck_Checker[Github]**
  - 使用了贝叶斯定理
  - 初始化所有潜在中文词的先验概率，将文本集（50篇医学文章）分词后，统计各个中文词的出现频率即为其先验概率
  - 当给定一待纠错单词时，需要找出可能的正确单词列表，这里根据字符距离来找出可能的正确单词列表
  - 对构造出来的单词做了一次验证后再将其加入候选集合中，即判断了下该词是否为有效单词，根据其是否在单词模型中

N-Gram模型使用方法

- [berkeley提供的自然语言处理工具包]
- [py-kenlm-model]
- [N-Gram-1]
- [N-Gram-2]
- [语言模型kenlm的训练及使用]
- [kenlm语言模型]

具体做法

- 将pdf转为txt文件, 同时使用多种规则过滤句子(包括数字,字母变星号, 长句剪断，根据逗号、句号、问号、感叹号裁剪句子, 去掉停用词)
- 中文分词, 由于我们探索的不是通用领域, 领域特定的分词效果可能不太友好, 因此我们需要获取一个领域特定的字典.
- 由于我们事先不知道错误类型, 可能是常用语错误, 也可能是领域特定词语出错, 针对这两种情况, 我们应该训练两个不同的N-Gram模型, 从而发现错误位置.

- 找到错误词后, 计算得到所有候选集(候选集的产生根据错误的类型惊行调整), 选择分数最大的作为正确的输出.

参考文献

- [开源项目]
- [基于语言模型的拼写纠错]
- [中文语音识别后检错纠错：n-gram + 拼音相似度 + 词语搭配]
- [NLP上层应用的关键一环——中文纠错技术简述]
- [Guest Editoral: Special Issue on Chinese as a Foreign Language]
- [A Study on Chinese Spelling Check Using Confusion Sets and N-gram Statistics]
- [A Hybrid Chinese Spelling Correction Using Language Model and Statistical Machine Translation with Reranking]
- [A Study of Language Modeling for Chinese Spelling Check]
- [Chinese Spelling Check System Based on N-gram Model]
- [N-gram Model for Chinese Grammatical Error Diagnosis]
- [Chinese Word Spelling Correction Based on N-gram Ranked Inverted Index List]