中文文本纠错系统

赖泽强，刘文卓，钱泽，谭超

北京理工大学 计算机学院

摘　　要: 本文提出了一种基于分词，语言模型和混淆集的中文文本纠错系统。在检错上，通过分词识别词级别的错误，通过语言模型识别字级别的错误。在纠错上，使用混淆集构造纠错候选，通过语言模型对候选进行排序选取。

关　键　词：中文文本纠错, 语言模型, 混淆集，分词

1　引言

文章自动纠错一直是自然语言处理的一个热门研究课题。前几年，出版业电子化的迅猛发展使得校对环节的工作量大大增加，人工校对是一项极其烦琐、费时的工作，因此自动纠错的课题被提上了日程。如今，互联网迅速发展，把中文自动纠错运用于互联网文本是一个刚刚兴起并且很有前景的研究项目。自动纠错功能可以很大程度上提升用户使用搜索、聊天等需要文本输入的功能时的用户体验。

在不考虑语法错误的情况下，英文纠错的任务主要集中在拼写纠错。相较于拼写错误，中文的错别字、形近字和音近字更难识别并且纠错。这主要是因为英文与中文的差异所造成的。

下文的组织如下：第二节简要阐述了一下文本纠错领域前人的工作；第三节从整体上讲解我们的文本纠错系统；第四节详细阐述系统的核心算法与概念；第五节是相关的实验；第六节对整篇文章进行总结并对未来工作作出展望。

2　相关工作

2.1 英文纠错

英文拼写纠错采用的主流方法为最短编辑距离。所谓编辑距离（Edit Distance），是指两个字符串之间，由一个转成另一个所需的最少编辑操作次数。许可的编辑操作总共有三个:将一个字符替换成另一个字符、插入一个字符或者删除一个字符。显然，编辑距离越小，两个字符串就越相似。这个概念是在 1965 年由俄罗斯科学家 Vladimir Levenshtein 提出的。

2.2 中文纠错

目前在学术界提出的中文文本纠错的策略有以下几种：

（1）根据大规模语料库的统计数据，通过 N-Gram 模型进行校对的方法，如：山西大学提供基于字字二元接续、词性二元接续和语义接续的中文文本的自动检错方法。

（2）基于混淆集的特征方法，如：清华大学和微软中国研究院提出的基于字串混淆集并采用 Winnow 方法对上下文特征进行机器学习来实现校对的方法。

（3）基于模式匹配的方法：清华大学和微软中国研究院提出的基于模糊词匹配的校对方法；东北大学提出的最长匹配的方法。

（4）基于散串技术的校对方法：哈尔滨大学提出的基于散串处理的校对方法。

3　系统架构

在方法选择上，我们的文本纠错系统采用了基于语料库的方法。该方法具有编码简单，速度较快的特点。

3.1 总体架构

整个中文文本校对系统主要包含两部分，前端用户界面，和后端文本校对核心。

用户输入待改错句子后，提交改错请求，后端接受请求后，提交给文件校对器校对，将校对结果返回给用户，并以一个友好的方式显示出来。

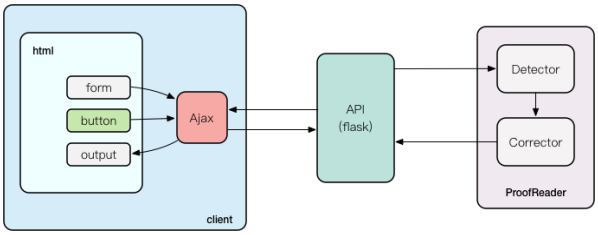

图 1、系统总体架构

3.2 文本校对模块

文本校对模块为本系统的核心，其主要分为两部分，文本检错器(Detector)和文本纠错器(Corrector)。

(1) 文本检错器（Detector）

检错器主要分为三个层次，用户级别、词级别和字级别。三个级别的错误检测结果汇总形成最终的错误候选。

用户级别：该级别主要依赖用户自定义的混淆集。当句子中含有用户自定义混淆集中的词或字时，立即将该词加入错误候选。

词级别：该级别依赖于分词器（Tokenizer）的分词能力，尤其是未登录词的识别能力。识别思路为 -- 对于分词器分出的词/字，如果其未出现在词库中，则将其加入错误候选。

字级别：该级别依赖于语言模型。其基本原理为利用语言模型给句子中的每个字打分，使用绝对中位差和中位方差计算每个字分数的异常情况，选出异常大于某个阈值的字加入错误候选。该部分将会在下一节核心算法中的语言模型中详细描述。

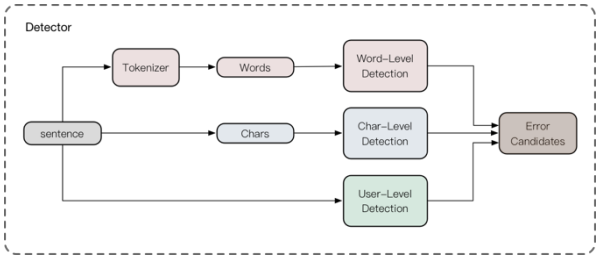

图 2、检错器总体架构

(2) 文本纠错器（Corrector）

纠错器主要根据不同的错误类型进行针对性的处理。对于包含在用户混淆集中的错误，直接将正确结果作为纠错候选返回；否则，利用音近和形近字字典生成大量候选，用候选替换错误片段生成新的句子，然后通过语言模型对句子进行打分，取得分高的候选作为纠错候选。

在生成纠正候选的过程中，如果错误片段是单字，则直接将该字替换为音近和形近的字加入纠正后选；对于错误是多字词的情况，我们只考虑两个字的词。其原因在于大部分多字的错误均为双字词，即时出现多字词错误（如成语打错），分词器大多数情况下也无法正确切分。与此同时，只考虑单字和双字的情况使得编码更加简单，性能也得到提升。

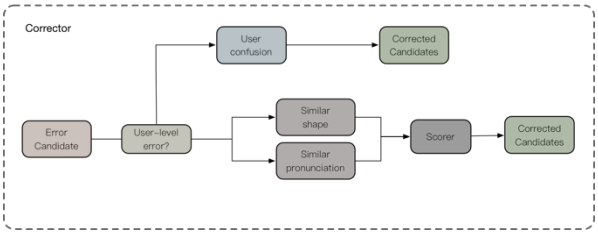

图 3、纠错器总体架构

3.3 数据准备模块

某种程度上，该部分属于预处理的内容，因此该部分未在图 1 系统总体架构中体现出来。但该部分仍然是系统不可或缺的一部分。

（1） 音近与形近字典

音近与形近字典，也就是混淆集，由容易被混淆的字符及词组成的数据集，在文本的拼写错误检测和纠正中起着关键作用(Wang 等人，2013)。大多数汉字在形状或发音上都有相似的特征。由于拼音输入法是当前最流行的中文输入法，因此在构建我们系统中使用的混淆集时，相似发音占优势。此外，形似的字符不是特别多，但仍然存在着相当大的比例（Liu 等人，2011）。书写相似的字符也被添加到我们的混淆集中。因此，系统使用的混淆集是由许多带有约束的规则创建的，包括相似的发音和类似的字形。

我们使用的是 SIGHAN-7 提供的混淆集，其中有很多繁体字。因此，我们在使用前对其进行了预处理，根据汉语简体繁体转换字典，将混淆集转换为简体。

（2） 语料库

语料库为大量报刊文本或者网络文本的数据集。我们需要根据语料库训练出语言模型，因此语料库的数量、准确度、全面性很大程度上决定了训练出来语言模型的适应能力。

我们选择多个目前比较常用的数据集作为语言模型的语料库：北京大学整理 2014 年人民日报的数据集、维基百科的中文语料。因为语言模型输入有格式要求，所以我们对这些数据集进行了合并、去除不必要成分、按句切分、分词等操作，最终生成一个词级别的语料库和一个字级别的语料库，分别用于词级和字级语言模型的训练。

3.4 可视化模块

当我们用命令行对校对结果进行展示时，其结果如表 1 所示，可以看出其效果不太直观。

冬冬今天戴来了一本好看的童话书

[(['戴来', 4, 6, 'word'], ['带来', '共来', '专来', '惠来', '泰来'])]

表 1. 命令行显示结果

为此，我们制作了一个网页（如图 2 所示）对校对结果进行展示。在网页中，被校对器识别出来的错误将会被标红显示，当用户选中某个具体的错误时，纠正候选将会以浮窗的形式显示出来（按可能性排序）。

图 4、网页显示结果

由于其具体实现不属于自然语言处理课程中的内容，这里不再做具体描述。

4 核心算法与概念

这部分将对第三节系统架构中未详细阐述部分进行介绍。

4.1 混淆集

混淆集是由常见的错别字/词/片段，及其修正后正确的字/词/片段构成的集合。一般以键值对的形式存储，如{尼好: 你好}。音近和形近字典也属于混淆集的范畴。

一个高质量庞大的混淆集可以极大提升文本纠正的效果，但该类资源较难获取，一般由用户自定义或由大公司进行收集构建。

4.2 语言模型

语言模型赋予每个句子一个概率，合法的句子获得的概率更大。由于每个句子由许多字组成，因此可以使用以下公式计算字粒度的句子概率。

$$P(S) = P(c_1, c_2, \cdots, c_n)$$
$$= P(c_1) \cdot P(c_2|c_1)$$
$$\cdot P(c_3|c_1, c_2) \cdots P(c_n|c_1, c_2, \ldots, c_{n-1})$$

通过引入马尔可夫假设，我们使每个字的出现概率只与其前 n 个字有关，这样就形成了 n-gram 模型。

在我们的检错系统中，我们使用一个窗口，大小通常与 n-gram 的 n 相同，我们分别计算以某个字为第 1，2,…,n 个字的长度为 n 的片段的语言模型分数的平均数作为该字的分数。在我们的实验中,我们使用了不同大小的 n-gram 分别对每个字打分，然后取平均。

在纠错系统中，我们对替换后的句子计算整个句子的分数，因为替换正确的句子得分通常更高，所以该方法能在一定程度上找出正确候选。

4.3 错字检测算法

在通过语言模型对句子中的每个字打分过后，需要根据这些分数选出错别字候选。

一个简单的思想是直接设置一个阈值，当某个字的分数低于这个阈值时则将该字加入候选。但是由于语言模型分数浮动较大，两个正确的字在不同句子中的分数也可能相差很大，因此该方法不具有可行性。

为此，我们采用了相对的方法。计算每个字的分数与其他字分数的偏离情况。如果偏离比例大于某个阈值则将其加入候选。

具体计算过程如下：

1. 计算所有分数的中位数（不取平均数是为了防止极端情况的影响）。
2. 计算每个分数相对于中位数的方差。
3. 计算这些方差的中位数（也即绝对中位差）。
4. 计算每个方差与方差中位数的比例。
5. 判断每个比例是否大于某个阈值，如果大于且分数小于中位数则视为错别字。

5 实验

5.1 数据集

我们使用了 Sighan7[1]组织者提供的音近和形近字典作为纠错器的纠错资源。

在语料库方面,我们使用人民日报 2014 语料库，并利用 kenlm[2]工具训练了 2-gram 和 3-gram 的语言模型。

5.2 实验

我们在 sighan7 的测试集下测试了该系统的性能。其中检错效果和纠错效果如下表所示：

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

[2] https://kheafield.com/code/kenlm/

阈值	Precision	Recall	F1
15	0.396	0.561	0.464
2.5	0.169	0.691	0.271

表 2、不同阈值下的检错效果

阈值	Precision	Recall	F1
15	0.551	0.411	0.471
2.5	0.250	0.521	0.338

表 3、不同阈值下的纠错效果

5.3 分析

检错：从召回率的值可以看出，当阈值为 15 时，字级别的检错几乎被完全屏蔽，也就是说词级别的检错能检测出约 50%的错误。

纠错：有检错的分析可知，阈值提高之后，字级别检错几乎被屏蔽，与此同时检错的精确率提高。从表 3 可以看出，阈值提高后，精确率提升明显，说明该系统对于词级别的纠错能力较强。而召回率的降低说明系统在字级别仍有一定的纠错能力，尽管该纠错能力较弱。

但是即便如此，从 Precision 来看，仍有大量误判，这是本系统的不足，因为从实际使用角度来看，高 Precision，中等 Recall 会更实用。

图 5 为一个文本纠正实例：

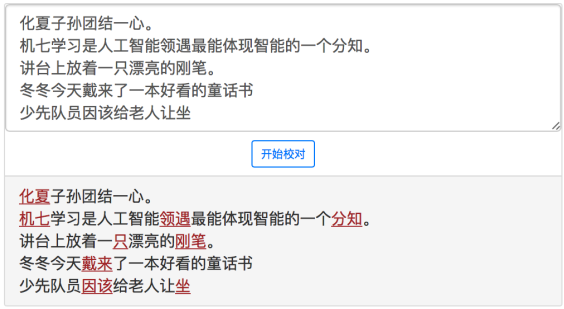

图 5、文本纠正实例

可以看出，对于未登录词，该系统有较强的识别能力和纠错能力。同时，语言模型也能够识别部分字错误。但是实际上“一只钢笔”，虽然只被识别出来了，但是纠正的时候，系统给出的候选却是[双，只，堆，支]，正确的支没能排到前面。这说明基于语言模型的纠错仍然具有一定的局限性。

6 结论

通过实验，我们验证了语言模型在文本纠错中应用的有效性。但系统的实用性仍然较差，分析原因，我们认为有以下几点：

1. 语料： 语料库的好坏对于语言模型的训练至关重要，相对来说，人民日报语料库仍然较小，使用更大更好的语料库去训练语言模型可能会获得更好的效果。
2. 数据预处理：分词效果对于词级别的错误识别十分重要，选择一个更好的分词工具，同时在分词时加入特定专业的词典对于文本校对会有更好的帮助。
3. 打分阈值：在字级别的识别中，阈值是一个超参数，其值越小，识别出的错字越多，但是识别错误的概率也越大，其值的选择对模型效果也很关键。

参考文献（References）

[1] Jui-Feng Yeh, etc. Chinese Word Spelling Correction Based on Rule Induction.

[2] Jui-Feng Yeh*, Sheng-Feng Li,etc. Chinese Word Spelling Correction Based on N-gram Ranked Inverted Index List

[3] Junjie Yu and Zhenghua Li. Chinese Spelling Error Detection and Correction Based on Language Model, Pronunciation, and Shape

[4] 基于语言模型的拼写纠错 https://cloud.tencent.com/developer/article/1156792

[5] 一些纠错相关的论文笔记 http://mlnote.com/2017/04/09/Reading-Notes-of-Error-Correction/

[6] 使用kenlm模型判别a/an错别字 https://zhuanlan.zhihu.com/p/39722203

[7] pycorrector https://github.com/shibing624/pycorrector