文章编号:1007-5321(2016)04-0092-06 **DOI**:10.13190/j.jbupt.2016.04.018

基于语料库的英语文章语法错误检查及纠正方法

谭咏梅, 王晓辉, 杨一枭
(北京邮电大学 智能科学与技术中心, 北京 100876)

摘要: 提出一种基于语料库的规则自动抽取方法,在此基础上提出了有限回退算法对英语文章进行语法错误检查及纠正. 该方法在 2013 年 CoNLL 语法自动检查及纠正评测数据上总体 F_1 为 31.96%,超过第 1 名的 31.20%,在冠词错误的纠正方面 F_1 为 33.45%,超过 2013 年最好成绩 33.40%,在名词错误的纠正方面 F_1 为 45.31%,超过 2013 年最好成绩 44.35%.

关 键 词: 语料库; 自动规则抽取; 有限回退; 错误检查及纠正
中图分类号: TP10 **文献标志码**: A

Grammatical Error Correction Based on Corpus

TAN Yong-mei, WANG Xiao-hui, YANG Yi-xiao
(Intelligence Science and Technology Center, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: Grammatical error correction (GEC) is the task of detecting and correcting grammatical errors in text written by non-native English writers. A limited back-off algorithm and corpus approach was proposed to handle the grammatical error problem in English text, useful and effective for GEC task. The GEC system yields F_1 score of 31.96% on the publicly available CoNLL-2013 shared task data, outperforming the first one with 31.20%.

Key words: corpus; automatic rule extraction; limited back-off; grammar error correction

英语是世界上最多国家使用的官方语言,也是世界上使用最广泛的第二语言. 对于英语为第二语言的学习者来说,语法错误是写作中最常见也是最难解决的问题之一. 语法自动检查及纠正(GEC, grammatical error correction)就是利用计算机自动对英语文章进行语法错误检查并纠正. 笔者提出了一种基于语料库的规则自动抽取方法,基于语料库从标注好的训练集中获取大量的错误的语法规则,在此基础上提出了基于有限回退(LB, limited back-off)算法的 GEC 方法,针对不同错误类型的特点,进行相应语法错误检查及纠正.

1 相关工作

GEC 最早开始于 20 世纪 80 年代, Writer's Workbench 主要是通过规则来进行语法错误的识别及纠正,随后出现了基于句法分析的 Epistle 系统, 1993 年微软的 word 使用了基于拓展短语结构语法(APSG, augmented phrase structure grammar)的分析方法对输入文本进行语法检查. 上述方法大都是依赖规则库来进行语法检查[1].

随着各种规模的语料库的出现,将语料库和统计方法相结合成为有效语法检查的方法. 相关的评

收稿日期: 2015-12-11
作者简介: 谭咏梅(1975—), 女, 副教授; 王晓辉(1990—), 男, E-mail: w9x2h1@qq.com.

测任务 HOO 在 2011 和 2012 年连续举办了 2 年，2013，2014 年 CoNLL 主办了 GEC[2-5].

研究者将 GEC 问题分解为多个子问题，针对每个问题建立模型，即针对每种错误类型建立纠正模型. CoNLL-2013 评测任务很多优秀的系统都基于这种方法[6]. 然而，这种方法中的模型性能依赖于先验知识且为每个模型设计不同的特征需要大量的人工操作. 所以，笔者采用基于语料库的方法对英文文章语法错误检查及纠正. 首先，利用有限训练集，通过规则自动抽取方法获取错误的语法规则. 然后，基于规则和大规模语料库在有限回退算法下进行纠正. 该方法对先验知识有较少的依赖且不需要设计大量的人工特征，大大减少了人工操作的量，并且可以实时增加语料规模.

2　基于语料库和回退算法的 GEC 方法

提出了一种基于语料库的规则自动抽取方法及基于有限回退算法的语法错误检查及纠正方法，系统架构如图 1 所示.

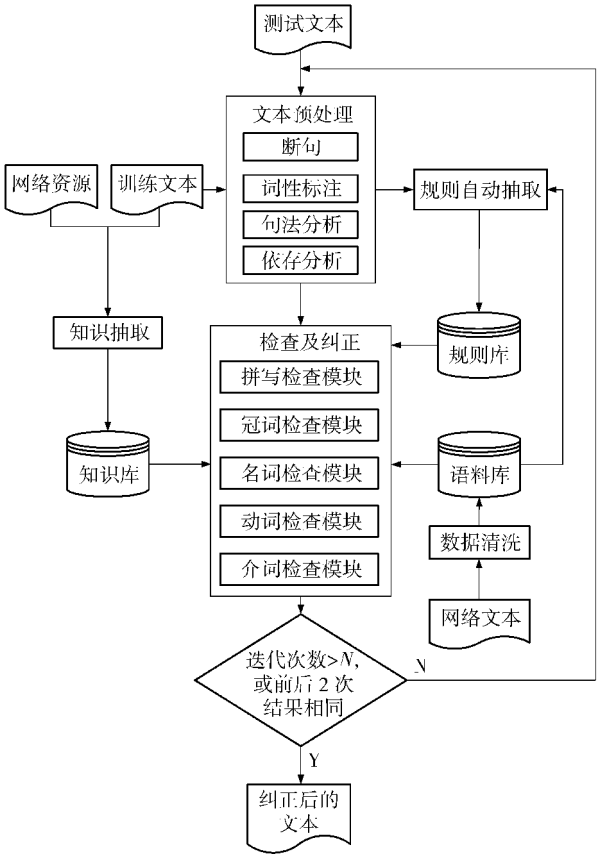

图 1　基于语料库的 GEC 架构

2.1　文本预处理

文本预处理主要包括断句、词性标注、句法分析、依存分析.

采用基于规则的方法进行断句，同时采用基于最大熵的词性标注算法. 使用基于概率上下文无关文法（PCFG，probabilistic context free grammer）的生成式句法分析模型以及基于神经网络（NN，neural-network）的依存分析器.

2.2　语料库构建

2.2.1　收集语料

为保证语料质量，语料来源为纽约时报. 对抓取的正文内容进行断句处理及清洗工作，及对标点符号做特殊的处理.

2.2.2　建立搜索服务

所提出的有限回退算法依赖语料库需要对语料库进行 n-grams 检索. 为了提高检索效率，对语料中的句子建立倒排索引.

2.3　基于语料库和回退算法的 GEC 方法

CoNLL-2013 评测任务中，主要关注英语文章中的冠词及定冠词错误、名词单复数错误、介词错误、主谓不一致和动词错误这 5 种错误. 笔者提出了规则自动抽取方法进行错误规则的抽取，然后利用有限回退算法结合规则进行纠正的方法.

符号说明：w 表示句子，w_i 表示句子 w 中的第 i 个单词，k 表示移动窗口的大小.

2.3.1　规则自动抽取方法

抽取算法描述如下，以冠词为例：

1）从训练集中取出含有标注为冠词错误类型的句子，进行词性标注，句法分析和依存分析；

2）定位错误单词的位置，并从句法分析和依存分析中分别获取含有错误单词的 NP 短语，det（限定词修饰）依存关系，并恢复成单词和词性标注结果的混合的 NP 短语；

3）获取“单词_词性”的模式，例如“the_dt _jj life”，“the_dt _jj _nn”，“_dt _jj life”等，统计在语料库中频次，如果频次低于某个阈值 T，则该模式是错误的，将这个模式和对应的修改方式放入规则库.

移动窗口

在识别句子 w 语法错误的过程中，给定位置为 i 的单词 w_i 后，可以通过窗口的大小为 k 的移动窗口 $MW_{i,k}(w)$ 获得 w_i 前后相关的短语[7]，以此来判断语法是否正确，如式（1）所示.

$$MW_{i,k}(w) = \{w_{i-j}, \cdots, w_{i-j+(k-1)}, j = 0, k-1\} \quad (1)$$

窗口的大小 k 的选择和 j 的最大取值对 GEC 的检查有一定的影响,在实际情况下,会根据某个具体的错误类型进行 k 和 j 的微调.

表1 窗口大小 k 的取值举例

窗口取值	n-grams
$k=5, j=[0,4]$	the accessibility in the forms, accessibility in the forms of in the forms of cards the forms of cards to, forms of cards to gain
$k=4, j=[0,3]$	accessibility in the forms, in the forms of, the forms of cards, forms of cards to
$k=3, j=[0,2]$	in the forms, the forms of, forms of cards
$k=2, j=[0,1]$	the forms, forms of

知识库

对于语法错误检查及纠正的过程而言,为了排除语言使用时的固定搭配,以及获取被纠正单词的候选集等,笔者构建了相应的知识库. 知识库主要包括从训练集中抽取的对应的修改方式,及从网络上获取的有关资料.

2.3.2 有限回退算法

纠正过程使用如下回退算法,比值越高说明替换后出现的频度越大,应该被纠正;比值越低说明原单词在语料库中的频度更大,不应该被纠正;如果比值在一定范围,则进行回退,缩短 n-grams 的长度,继续进行查询比较,直到 n-grams 长度为 1. 待纠正单词所在的 n-grams 可用式(2)来描述.

$$S_{i,k}(w) = \Sigma_{\text{ngram} \in MW_k(w)} \text{count}(\text{ngram}) \quad (2)$$

其中:$S_{i,k}(w)$ 代表包含单词 w_i 且由 k 个单词组成的 n-grams 在语料库中频度和. k 从大到小取值回退,来保证尽可能的准确. 由于不同的窗口大小取出的值可能存在较大的差异,为了提高算法的准确率,采用有限回退算法.

即在回退的过程中不仅要满足比值的条件,同时对于当前统计值 $S_{i,k}(w)$ 也要进行判断,对于不同的 k 值,统计值高于阈值才进行回退,以提高算法的准确率.

具体的纠正函数的算法描述如下.

function Replace(i,k,w')

$M = \{m_1, m_2, m_3, m_4, m_5\}$

$$r = \frac{S_{i,k}(w')}{S_{i,k}(w)}$$

if $r > \lambda$ and $S_{i,k}(w) < M_k$

return True

else if $k > 2$ and $r > \epsilon$:

return Replace (i,k,w')

else

return False

2.3.3 循环检查策略

由于句子中可能含有多个错误,且不同的错误之间可能具有一定的依赖性. 例如,"In supermarket monitor is needed because we have to track thieves.",只有判断出"monitor"应该纠正为"monitors",才能将"is"纠正成"are".

为了尽可能找出所有的错误,笔者采用循环检查的策略. 即在完成一遍检测之后,继续重复原来的步骤,直到多次纠正结果一致.

2.3.4 拼写检查模块

如果句子中存在错误的单词拼写,对后续的语法检查会产生很大的影响. 笔者针对单词的拼写检查使用规则和统计相结合的方法. 对于非词错误的单词 w_i,给出编辑距离小于 2 的所有候选集合,用 $G(w_i)$ 表示,然后通过在语料库中查询出现频度最大的候选词作为纠正的单词. 为了更准确地找到候选项,笔者使用了候选单词的 3 种组合[8],如式(4)所示.

$$G(w_i) = \{w_j, j = 0, \cdots, k-1\} \quad (3)$$

$$S(w) = \max_{j=\{0,\cdots,k-1\}} \text{sum}[\{\text{count}(w_{i-1}w_j) + \text{count}(w_jw_{i+1}) + \text{count}(w_{i-1}w_jw_{i+1})\}] \quad (4)$$

其中:i 为被给定单词的位置,w_i 为被给定的单词,$G(w_i)$ 为 w_i 的候选集的集合. $S(w)$ 为所有候选项中二元组以及三元组在语料库中总和最大的候选单词.

2.3.5 冠词检查模块

冠词检查模块主要针对于定冠词 the,以及冠词 a、an 是否正确使用的情况.

首先,采用句法分析获取所有的内部不包含嵌套 NP 的最小 NP. 然后,在知识库中查询是否有触发的规则,如果有,则根据规则进行纠正. 最后,利用有限回退算法进行二次检查和纠正.

2.3.6 名词检查模块

名词检查模块主要针对于名词单复数使用错误的情况. 由于名词单复数形式存在一致的情况,首先建立关于名词单复数的列表,如表 2 所示.

表 2　名词单复数的列表

单词原形	复数形式	标记
apple	apples	1
people	peoples	2
tomato	tomatos	3
tomato	tomatoes	1
…	…	…

表 2 中对于存在复数且正确转换的标记为 1，对于单复数一致的单词标记为 2，对于存在复数但是错误转换的标记为 3.

名词检查模块具体的检查过程描述如下：

1）首先根据词性标注的结果，找出所有标记为 NNS，NN 的单词；

2）对于标注为 NNS 的单词，查询单复数知识库，如果标记为 2，则直接将该单词修改为原形. 如果标记为 3，则先找到该单词的原形，再利用原形找到该单词的正确复数形式，即此时后面标记为 1 的记录，并跳转到第 3）步. 如果标记为 1，直接跳转到第 3）步；

3）利用有限回退算法，根据在此处名词单数和复数在语料库中出现的频率以及比值，判断是否应该完成纠正.

2.3.7　动词检查模块

动词检查模块主要针对于动词的使用情况，包括动词的使用形式以及主谓一致. 由于动词的形式复杂多样，笔者建立一个动词的转化表，如表 3 所示.

表 3　动词各种形式列表

动词	可能的变化
take	taking，took，taking，taken，takes
taking	take，took，taking，taken，takes，to take
took	take，taking，taking，taken，takes
…	…

动词检查模块具体的检查过程描述如下：

1）首先根据词性标注结果找出标注为 VB，VBD，VBG，VBN，VBP，VBZ 的单词；

2）依次在知识库里找到该单词的所有可能变化；

3）最后利用有限回退算法，进行错误检查及纠正.

2.3.8　介词检查模块

介词检查模块主要针对介词使用错误. 由于介词种类有限，笔者利用训练集构建了一个介词替换的列表，如表 4 所示.

由于介词使用的灵活性，笔者增加了介词短语常用搭配列表，如果句中出现了常用介词短语，则不进行纠正，否则利用有限回退算法进行检查及纠正.

表 4　介词的候选集的存储方式举例

原介词	替换候选项
of	in，with，to，for，between，over，by，on，against，“ ”
in	of，on，for，by，to，over，from，at，along，during，with，“ ”
on	for，of，to，with，over，in，“ ”
…	…

注：“ ”代表为空，即去掉该介词.

3　实验

3.1　实验设置

利用 StanfordNLP 对语料进行断句、词性标注、句法分析和依存分析. 利用开源软件 enhant 生成非词错误的候选集，利用 mysql 搭建提供候选集的知识库，并利用 elasticsearch 搭建提供 *n*-grams 查询的搜索引擎.

3.2　实验数据

实验所用数据来源于 CoNLL-2013 的 GEC 评测任务，训练数据和测试数据统计结果如表 5 所示.

表 5　CoNLL-2013 的 GEC 评测任务训练数据和测试数据统计

错误类型	训练集		测试集	
	数目	占比	数目	占比
冠词	6 658	14.8	690	19.9
介词	2 404	5.3	312	9.0
名词	3 779	8.4	396	11.4
主谓一致	1 453	3.2	122	3.5
动词形式	1 527	3.4	124	3.6
5 种类型	15 821	35.1	1 644	47.4
所有类型	45 106	100.0	3 470	100.0

从表 5 可以看出，训练集和测试集中的错误类型占比并不统一，例如对于动词而言，在测试集中占了更高的比例.

3.3　评测方法

系统的性能根据 CoNLL-2013 评测标准，采用 F_1[4]，定义如式(5)所示.

$$F_1 = \frac{(\beta^2+1)PR}{\beta^2(P+R)},\ (\beta=1) \tag{5}$$

其中:P 与 R 分别表示准确率和召回率,其定义如下.

$$P = \frac{N_{\text{correct}}}{N_{\text{predicted}}} \tag{6}$$

$$R = \frac{N_{\text{correct}}}{N_{\text{target}}} \tag{7}$$

3.4 实验结果及分析

为了验证笔者方法的有效性,与 CoNLL-2013 评测单项第 1 名进行了对比.

如表 6 所示,回退模型改进为有限回退模型后,准确率有了大幅度的提升,并且提高了 F 值. 使用规则结合有限回退算法后,最终 F 值高于 CoNLL-2013 评测的单项第 1 名.

表 6 冠词检查模块的结果比较

模型	准确率/%	召回率/%	F 值/%
规则 + 语料库 + 回退算法	23.25	48.55	31.44
规则 + 语料库 + 有限回退算法	43.49	27.12	**33.45**
(Rozovskaya et al,2013)	47.84	25.65	33.40

如表 7,8,9 所示,在将回退算法为有限回退算法,F 值都有一定程度的提升. 其中,名词错误的 F 值高于 CoNLL-2013 评测单项第 1 名.

表 7 名词检查模块的结果比较

模型	准确率/%	召回率/%	F 值/%
语料库 + 回退算法	63.99	30.05	40.90
语料库 + 有限回退算法	40.60	51.26	**45.30**
(Rozovskaya et al,2013)	52.23	38.38	44.25

表 8 动词检查模块的结果比较

模型	准确率/%	召回率/%	F 值/%
语料库 + 回退算法	12.79	22.36	16.27
语料库 + 有限回退算法	19.37	22.36	20.76
(Rozovskaya et al,2013)	38.94	17.89	24.51

表 9 介词检查模块的结果比较

模型	准确率/%	召回率/%	F 值/%
语料库 + 回退算法	8.01	25.32	12.17
语料库 + 有限回退算法	13.46	18.83	15.70
(Yoshimoto et al,2013)	29.10	12.54	17.53

如表 10 所示,经过循环检查,使得最终的准确率和召回率都有提升,最终综合 F 值高于 CoNLL-2013 评测第 1 名.

表 10 综合检查结果

错误类型	准确率/%	召回率/%	F 值/%
冠词	27.12	18.26	21.83
+ 名词	31.32	30.61	30.96
+ 主谓一致 + 动词形式	29.39	33.84	31.46
+ 介词	27.62	36.58	31.47
循环检查	28.01	37.19	**31.96**
(Rozovskaya et al, 2013)	46.45	23.49	31.20

4 结束语

笔者提出了一种基于语料库的规则自动抽取方法进行规则自动获取,在此基础上提出了有限回退算法来对英语文章进行语法错误检查及纠正. 该方法在 2013 年 CoNLL 语法自动检查及纠正评测数据上总体 F_1 为 31.96%,超过第 1 名的 31.20%,在冠词错误的纠正方面 F_1 为 33.45%,超过 2013 年最好成绩 33.40%,在名词错误的纠正方面 F_1 为 45.31%,超过 2013 年最好成绩 44.35%,实验结果表明,该方法对 GEC 任务是有效的.

但仍然有一些问题需要解决:1) 介词缺失的情况,例如“After several years researching and studying”,应该修改为“After several years of researching and studying”;2) 将一个单词修改为多个单词的错误,例如“personal information releasing without knowing”应该修改为“personal information being released without knowing”等.

参考文献:

[1] 刘磊. 面向自动语法检查的依存规则研究[D]. 北京:北京外国语大学, 2014.

[2] Robert D, Adam K. Helping our own: the HOO 2011 pilot shared task[C]//Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation, Association for Computational Linguistics, 2011.

[3] Robert D, Ilya A, George N. HOO 2012: a report on the preposition and determiner error correction shared task [C]//Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, Association for Computational Linguistics, 2012.

[4] Ng H T, Wu Siewmei, Wu Yuanbin, et al. The CoNLL-2013 shared task on grammatical error correction[C]// Proceedings of the Seventeenth Conference on Computational Natural Language Learning. Association for Computational Linguistics, 2013.

[5] Ng H T, Wu Siewmei, Briscoe T, et al. The CoNLL-2014 shared task on grammatical error correction[C]// Proceedings of the Eighteenth Conference on Computational Natural Language Learning, 2014.

[6] Mariano F, Zheng Yuan, Øistein E, et al. Grammatical error correction using hybrid systems and type filtering [C]// Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task. Association for Computational Linguistics, 2014.

[7] Kao Tinghui, Chang Yuwei, Chiu Hsunwen, et al. CoNLL-2013 shared task: grammatical error correction NTHU system description[C]// Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, 2013.

[8] Pratip S, Bidyut B C. A simple real-word error detection and correction using local word bigram and trigram[C]// ROCLING, 2013.

(上接第 44 页)

[3] Lee J, Kim D, Jang M, et al. Proxy-based mobility management scheme in mobile content centric networking environment [C]// Proceedings of 2011 IEEE International Conference on Consumer Electronics (ICCE). Las Vegas: IEEE, 2011: 595-596.

[4] Hermans F, Ngai E, Gunningberg P. Mobile sources in an information-centric network with hierarchical names: an indirection approach [C]// Proceedings of the ACM SIGCOMM 2011 Workshop on Information-Centric Networking (ICN). Sweden: ACM, 2011: 1-6.

[5] Zhenkai Z, Afanasyev A, Lixia Z. A new perspective on mobility support [EB/OL]. 2013. http://named-data.net/ techreports.html.

[6] Lee J, Cho S, Kim D. Device mobility management in content-centric networking [J]. IEEE Communication Magine, 2012, 50(2012): 28-34.

[7] Baumann F V, Niemegeers I G. An evaluation of location management procedures [C]// Proceedings of the 3rd Annual International Conference on Universal Personal Communications. San Diego: IEEE, 1994: 359-364.

[8] Choi Y H, Chung T M. Using correspondent information for route optimization scheme on proxy mobile IPv6 [J]. Journal of Networks, 2010, 5(8): 984-989.