开题报告

林语盈[1)] 徐熔杞[1)] 黄逸轩[1)]

[1)](南开大学 计算机学院, 天津, 中国)

摘 要

关键词

中图法分类号 TP391 DOI 号 10.11897/SP.J.1016.01.2022.00001

Project Proposal

Yuying Lin[1)] Rongqi Xu[1)] Yixuan Huang[1)]

[1)](Department of Computer Science, Nankai University,Tianjin, China)

Abstract

Key words

1 问题定义

语法纠错 (GEC) 是 NLP 领域中的一个重要任务，GEC 任务要求检测一句话中是否有语法错误，并自动将检测出的语法错误进行纠正。

2 目前国内外研究现状

当前主要的模型可分为基于端对端神经网络机器翻译的方法 (包括基于 RNN、CNN、Transformer 的方法)、基于大规模预训练模型的方法、序列标注的方法等。

2.1 基于神经机器翻译的方法

与统计机器学习的方法不同，此类方法无需构建句子的特征，而是直接采用正确句-错误句的平行数据对作为模型的输入，采用编码器-解码器的结构。具体来说，编码器或解码器的选择包括 CNN、RNN 或注意力机制。大规模的预训练模型也被广泛尝试。

基于 RNN 模型 2017 年，Sakaguchi 等人创新性地将强化学习应用于 GEC，以更好地纳入任务特定指标，并克服传统 MLE 导致的暴露偏差。2018 年,[1] 提出了另一项使用基于 RNN 的 NMT 模型的新工作，该模型旨在提高源句子的流利度和可靠性，以纠正语法错误，称为流利度提升学习。这项工作为使用基于 NMT 的方法解决 GEC 提供了一个新的视角。

基于 CNN 模型与基于 RNN 的模型相比，基于 CNN 的模型的优势在于，CNN 在捕捉本地上下文方面更有效，从而纠正更广泛的语法错误。多层体系结构可以捕获长期的上下文信息。此外，无论输入长度如何，只对输入执行恒定数量的非线性计算，而在 RNN 中，非线性的数量为 O（n），从而减少了远距离单词的影响。

基于 transformer 模型。随着 Transformer[2] 的提出，许多基于 NMT 的 GEC 模型用 Transformer 取代了传统的基于 RNN 的编码器-解码器，并取得了有希望的结果。2019 年，复制机制[3] 被创新地应用于 GEC 任务，允许模型将未更改的源单词直接复制到目标端，并实现了最先进的性能。与 RNN 和 CNN 相比，Transformer 通过注意在句子中建立远程依赖的能力更强，并且它能够实现更高效的并行计算。此外,[4] 提出了浅攻击解码 (SAD)，以提高 transformer 在线即时语法纠错 (GEC) 的计算性能。

2.2 大规模预训练模型

随着数据和算力的不断丰富，大规模预训练模型被用于 GEC 任务。2019 年,[5] 首先将大规模的预训练模型与 transformer 用于 GEC 任务。2020 年,[6] 将预训练的模型与编解码模型结合并运用到汉语

收稿日期: 2022-05-04; 修改日期: 2022-05-04 林语盈, 学号: 2012174 E-mail: 2012174@mail.nankai.edu.cn. 徐熔杞, 学号: 1911281 E-mail: 18 5010769@qq.com. 黄逸轩, 学号: 2012067 E-mail: spacetime57@163.com.

语法纠错任务中。[7] 将预先训练好的掩码语言模型（MLM），整合到用于语法错误纠正（GEC）的编码器-解码器（EncDec）模型中。首先使用给定的 GEC 语料库对模型参数进行微调，然后将微调后的模型参数输出作为 GEC 模型中的附加特征。该模型在 BEA-2019 和 CoNLL-2014 中取得了当时最优的效果

2.3 基于序列标注的方法

序列标注模型只是对特定位置的词做特定标注的更改，以[8] 为例，将 GEC 问题转换为局部词语，即 token-level 的特定操作，如 3 所示。这样使得序列标注模型具有更小的解空间，相对于 Seq2seq 而言，效率大幅提升。在[9] 中十分详细地定义了各种类型的操作标签

3 计划复现论文理解阐述

3.1 GECToR −Grammatical Error Correction: Tag, Not Rewrite

本文的动机旨在解决 NMT-based 模型的三个问题：即推理速度慢、需要大量的训练数据以及可解释性问题。本文针对以上三点提出了序列标注方法，取代了原本的序列生成方法本文提出的基于序列标注的方法可概括为以下三个方面：

3.1.1 token-level transformation/edit

该方法扩大了输出序列的解空间，对于大小为 5000 的词表，包含了 4971 个基本变换（包括保持不变、删除和其他 1167 个依赖于 token 的添加词语操作以及 3802 个替换词语操作）以及 29 个与具体 tokne 独立的 g-transformation，下面具体阐述两类 transformation 方法，即 basic transformation 和 g-transformation

basic transformation：基本变换即最常见的 token-level 的操作，例如保持当前的词语不变、删除当前词语、在当前词语后添加词语以及将当前词语替换为其它词语

g-transformation： g-transformation 是基于特定任务的操作，例如转换词语的大小写、将当前词语与后一个词合并、将当前的词分解为两个词、名词的单复数变换、动词的形式变换等

3.1.2 Tagging model architecture

模型结构，本文使用的 GEC 序列标注模型采用基于预训练的 BERT 的 transformer 作为 encoder，其中 transformer 含有两个带有 softmax 的线性层。其中，分词操作依赖于具体地 transformer 的设计。为了在 token-level 传递信息，本文将每个 token 的第一个词经过 encoder 得到的特征传入后续的线性层，两个线性层分别负责错误检测和纠错操作类型的标注

3.1.3 Iterative sequence tagging approach

采用多次迭代进行纠错的方式，逐步的改正错误序列。随着每次迭代，纠正错误的次数逐渐减少，并且大多数错误的改正都发生在前两次迭代中。因此，对于迭代次数进行上限的限制可以在保证生成效果的同时提高效率

3.1.4 实验

预处理

预处理操作分为三个步骤：

- 将训练集/验证集中的序列数据对中的每一个错误的 token 与其相对应的纠正后的 token 建立映射关系
- 对于每个映射，找到其对应的变换类型
- 对于错误序列中的每个 source token，只保留一个变换类型，这样在每个迭代周期中都只对 token 进行一种变换，通过多次迭代以达到最终对原序列进行语法纠错的目的

训练：训练过程分为三个步骤，即首先在合成的错误句子上的预训练，再在在错误句子上进行微调，最后在错误句子和正确句子上进行微调

3.1.5 可以做出的改进

- 设置一个不改变原来 token 的概率阈值，记为 confidence bias，增加模型的灵活度
- 在错误检测层，设置一个句子级别的 minimum error probabilityl 阈值，增加模型的稳定性

3.2 Encode, Tag, Realize: High-Precision Text Editing

LASERTAGGER 是一种序列标记方法，可将文本生成转换为文本编辑任务。通过使用三个主要的编辑操作从输入中重构目标文本：保留字符，删除字符以及在字符之前添加短语。为了预测编辑操作，论文了一个新的模型，该模型将 BERT 编码器与自回归 Transformer 解码器结合在一起。

文本编辑的方法: 我们将其转换为标记问题。其主要组成部分：（1）标记操作，（2）如何将纯文本训练目标转换为标记格式，以及（3）将标记转换为最终输出文本的实现步骤。

- 标记操作: 我们的标记器为每个输入字符分配一个标记。标签由两部分组成：基本标签和添加短语。基本标签是 keep 或 delete，它指示是否在输出中保留字符。添加短语 P（可以为空）强制将 P 添加到相应的字符之前。P 属于词汇 V，词汇 V 定义了一组单词和短语，可以将这些单词和短语插入输入序列以将其转换为输出。基本标签 B 和添加短语 P 的组合被视为单个标签，并用 PB 表示。唯一标签的总数等于基本标签的数量乘以短语词汇量的大小，因此一共有 2|V| 个唯一标签。

- 优化短语词汇: 短语词汇表包含可以在源词之间添加的短语。一方面，我们希望最小化短语的数量，以使输出标签的词汇量保持较小。另一方面，我们希望使用可用的标记操作将可以从源重构的目标文本的百分比最大化

- 将训练目标转换为标签: 确定短语词汇后，我们可以将训练数据中的目标文本转换为标签序列。给定短语词汇，我们无需计算 LCS，但可以利用更有效的方法，该方法遍历输入中的单词，并贪婪地尝试将它们与目标中的单词进行匹配，如果存在无匹配，则对词汇 V 中的短语进行匹配。这可以在 O（|s|*np）时间完成，其中 np 是 V 中最长短语的长度

- 实现：获得预测的标签序列后，我们将其转换为文本。对于保留，删除和添加的基本标记操作，实现是一个简单的过程。此外，我们调整句子边界的大小写。如果我们引入特殊的标记，例如上文中提到的 PRONOMINALIZE，则实现将变得更加复杂。对于此标签，我们需要从知识库中查找被标签实体的性别。拥有单独的实现步骤是有益的，因为我们可以仅在对适当代词有信心的情况下才决定对代词进行名词化，否则可以使实体提及保持不变。

标记器标记器由两个部分组成：一个编码器，它为输入序列中的每个元素生成激活向量；一个解码器，将编码器的激活转换为标签标记：

- 编码器：选择 BERT Transformer 模型作为编码器，使用基于 BERT 的体系结构，该体系结构包含 12 个自我注意力层

- 解码器：在原始 BERT 论文中，简单的解码机制用于序列标记：通过在编码器 logits 上应用 argmax，可在单次前馈过程中生成输出标记。这样，无需对序列中标签之间的依赖关系进行建模，即可独立预测每个输出标签。
  为了更好地建模输出标签之间的依赖关系，本文提出了一种更强大的自回归解码器。具体来说，我们在 BERT 编码器的顶部运行一个单层 Transformer 解码器。在每个步骤中，解码器都会使用先前预测的标签的嵌入以及来自编码器的激活

3.3 Tail-to-Tail Non-Autoregressive Sequence Prediction for Chinese Grammatical Error Correction

模型 TtT[10] 发表在 ACL 2021 上，是腾讯 AI Lab 在中文语法错误纠正上（CGEC）的又一重要成果。

对于 CGEC 任务，语法错误类型和对应的纠正操作可以总结为替换、插入和删除、局部转换三类，其中替换不会改变句子前后的长度，而插入和删除以及局部转化可能会改变句子的长度，属于变长操作。近年来，使用序列标注模型有效解决了 seq2seq 框架因为错误累计而出现幻象（hallucinate），即产生不相关或者反事实的内容的问题，但却因为需要给每个词进行标记而导致计算效率非常低。于此同时，单纯的标记方法需要进行多轮预测只至稳定，这进一步降低了效率。最近许多研究关注于在 CGEC 任务上精调 BERT[11] 等预训练语言模型，但是受限于模型本身，它们大部分都只能处理固定长度的错误纠正，而无法执行变长操作。

针对以上问题，论文提出了一个新的模型框架 TtT（Tail-to-Tail non-autoregressive sequence prediction）。其总体结构如图**??**所示。它仍然使用 Bert 作为骨干模型，可以将字符的信息直接从 bottom tail 传到 up tail，并且学习双向上下文的信息。为了能够同时进行替换、删除、插入、局部转换等各种操作，模型在 up tail 端使用了条件随机场（Conditional Random Fields, CRF）[12] 通过对相邻字符的依赖关系建模来进行非自回归的序列预测。另外，针对任务本身存在的类不平衡问题（句子中大部分字符是正确的，只有很少一部分需要修改），模型还使用了焦点损失惩罚策略[13]。

3.3.1 变长输入

为了使得模型能够执行插入删除等变长操作，即输入 $X = (x_1, x_2, \cdots, x_T)$ 的长度 T 和输出 $Y = (y_1, y_2, \cdots, y_{T'})$ 的长度 T' 不一定相等，模型会对输入输出的格式进行一定的处理。假设输入 $X =$

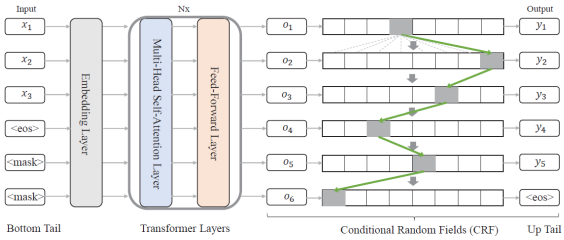

Figure 3: The proposed tail-to-tail non-autoregressive sequence prediction framework (TtT).

图 1 TtT 总体结构

$(x_1, x_2, x_3, <\text{eos}>)$，当 $T = T'$ 时，不用处理；当 $T > T'$，比如 $Y = (y_1, y_2, <\text{eos}>)$，这意味着 X 中的某些字符被删除，那么在训练阶段，会在 Y 的尾部补充 $T - T'$ 个特殊的符号 <pad>，以使得 $T = T'$，即 $Y = (y_1, y_2, <\text{eos}>, <\text{pad}>)$ 当 $T < T'$ 时，比如 $Y = (y_1, y_2, y_3, y_4, y_5, <\text{eos}>)$，这就意味着在原始的输入 X 中插入了额外的信息，那么模型将会在 X 的尾部插入特殊的符号 <mask>，来暗示这些位置可能可以插入一些新的字符，即 $X = (x_1, x_2, x_3, <\text{eos}>, <\text{mask}>, <\text{mask}>)$

3.3.2 双向语义建模

在准备好输入样本后，将会输入由预训练中文 BERT 模型初始化的嵌入层和堆叠的 Transformer 层来对语义信息建模。具体来说，首先将词向量和位置向量相加来获取其表征：

$$\mathbf{H}_t^0 = \mathbf{E}_{w_t} + \mathbf{E}_{p_t} \tag{1}$$

其中 0 是层数标号，t 是状态标号，$\mathbf{E}_{w_t}$ 和 $\mathbf{E}_{p_t}$ 分别是 tokens 的词向量和位置向量。然后嵌入向量 $\mathbf{H}_t^0$ 会被输入多个 Transformer 层，使用多头自注意力机制来学习双向语义表征。

3.3.3 非自回归序列预测

在获取了句子的表征向量 $\mathbf{H}^L$ 后，理论上来说可以直接添加一个 softmax 层来预测结果：

$$\begin{aligned} \mathbf{s}_t &= \mathbf{h}_t^\top \mathbf{W}_s + \mathbf{b}_s \\ P_{\text{dp}}(y_t) &= \text{softmax}(\mathbf{s}_t) \end{aligned} \tag{2}$$

生成对于目标词 V 的概率分布 $P_{\text{dp}}(y_t)$，选取概率最大的那一项作为最终的结果。尽管这种直接预测的方法在固定长度的语法纠错问题上很高效，但是它只能进行相同位置的替换操作，无法进行变长操作。因此论文提出使用 CRF 解决非自回归序列预测的这一问题。

在 CRF 框架下，给定输入序列 X，长度为 T' 的目标序列 Y 的概率为：

$$P_{\text{crf}}(Y \mid X) = \frac{1}{Z(X)} \exp\left(\sum_{t=1}^{T'} s(y_t) + \sum_{t=2}^{T'} t(y_{t-1}, y_t)\right) \tag{3}$$

其中 $Z(X)$ 是归一化因子，$s(y_t)$ 表示 y 在位置 t 的得分，值 $t(y_{t-1}, y_t) = \mathbf{M}_{y_{t-1}, y_t}$ 表示从 token y_{t-1} 到 y_t 的转移得分，$\mathbf{M} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ 是转移矩阵，可以通过两个低维的神经参数矩阵 $\mathbf{E}_1, \mathbf{E}_2 \in \mathbb{R}^{|\mathcal{V}| \times d_m}$ 来估计：

$$\mathbf{M} = \mathbf{E}_1 \mathbf{E}_2^\top \tag{4}$$

4 Final Project 分工

本小组共复现三篇论文，每人负责一篇论文，林语盈负责《Tail-to-Tail Non-Autoregressive Sequence Prediction for Chinese Grammatical Error Correction》、徐熔杞负责《Cross-Sentence Grammatical Error Correction》、黄逸轩负责《GECToR –Grammatical Error Correction: Tag, Not Rewrite》

参考文献

[1] GE T, WEI F, ZHOU M. Fluency boost learning and inference for neural grammatical error correction[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). [S.l.: s.n.], 2018: 1055-1065.

[2] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[3] ZHAO W, WANG L, SHEN K, et al. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data[J]. arXiv preprint arXiv:1903.00138, 2019.

[4] SUN X, GE T, WEI F, et al. Instantaneous grammatical error correction with shallow aggressive decoding[J]. arXiv preprint arXiv:2106.04970, 2021.

[5] ALIKANIOTIS D, RAHEJA V. The unreasonable effectiveness of transformer language models in grammatical error correction[J]. arXiv preprint arXiv:1906.01733, 2019.

[6] WANG H, KUROSAWA M, KATSUMATA S, et al. Chinese grammatical correction using bert-based pre-trained model[J]. arXiv preprint arXiv:2011.02093, 2020.

[7] KANEKO M, MITA M, KIYONO S, et al. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction[J]. arXiv preprint arXiv:2005.00987, 2020.

[8] AWASTHI A, SARAWAGI S, GOYAL R, et al. Parallel iterative edit models for local sequence transduction[J]. arXiv preprint arXiv:1910.02893, 2019.

[9] OMELIANCHUK K, ATRASEVYCH V, CHERNODUB A, et al.

Gector–grammatical error correction: tag, not rewrite[J]. arXiv preprint arXiv:2005.12592, 2020.

[10] LI P, SHI S. Tail-to-tail non-autoregressive sequence prediction for chinese grammatical error correction[C]//Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). [S.l.: s.n.], 2021: 4973-4984.

[11] DEVLIN J, CHANG M W, LEE K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding[C/OL]// Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, 2019: 4171-4186. https://aclanthology.org/N19-1423. DOI: 10.18653/v1/N19-1423.

[12] LAFFERTY J D, MCCALLUM A, PEREIRA F C N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data[C]//ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001: 282–289.

[13] LIN T Y, GOYAL P, GIRSHICK R, et al. Focal loss for dense object detection[J/OL]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2):318-327. DOI: 10.1109/TPAMI.2018.2858826.

[14] WANG D, SONG Y, LI J, et al. A hybrid approach to automatic corpus generation for Chinese spelling check[C/OL]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: Association for Computational Linguistics, 2018: 2517-2527. https://aclanthology.org/D18-1273. DOI: 10.18653/v1/D18-1273.