基于预训练表示模型的英语词语简化方法

强继朋 [1] 钱镇宇 [1] 李 云 [1] 袁运浩 [1] 朱 毅 [1]

摘 要 词语简化是将给定句子中的复杂词替换成意义相等的简单替代词,从而达到简化句子的目的. 已有的词语简化方法只依靠复杂词本身而不考虑其上下文信息来生成候选替换词, 这将不可避免地产生大量的虚假候选词. 为此, 提出了一种基于预语言训练表示模型的词语简化方法, 利用预训练语言表示模进行候选替换词的生成和排序. 基于预语言训练表示模型的词语简化方法在候选词生成过程中, 不仅不需要任何语义词典和平行语料, 而且能够充分考虑复杂词本身和上下文信息产生候选替代词. 在候选替代词排序过程中, 基于预语言训练表示模型的词语简化方法采用了 5 个高效的特征, 除了常用的词频和词语之间相似度特征之外, 还利用了预训练语言表示模的预测排名、基于基于预语言训练表示模型的上、下文产生概率和复述数据库 PPDB 三个新特征. 通过 3 个基准数据集进行验证, 基于预语言训练表示模型的词语简化方法取得了明显的进步, 整体性能平均比最先进的方法准确率高出 29.8%.

关键词 词语简化, 候选词生成, 候选词排序, 预训练语言表示模型

引用格式 强继朋, 钱镇宇, 李云, 袁运浩, 朱毅. 基于预训练表示模型的英语词语简化方法. 自动化学报, 2022, **48**(8): 2075−2087

DOI 10.16383/j.aas.c200723

English Lexical Simplification Based on Pretrained Language Representation Modeling

QIANG Ji-Peng[1] QIAN Zhen-Yu[1] LI Yun[1] YUAN Yun-Hao[1] ZHU Yi[1]

Abstract Lexical simplification (LS) aims to replace complex words in a given sentence with their simpler alternatives of equivalent meaning, so as to simplify the sentence. Recently unsupervised lexical simplification approaches only rely on the complex word itself regardless of the given sentence to generate candidate substitutions, which will inevitably produce a large number of spurious candidates. Therefore, we present a lexical simplification approach BERT-LS based on pretrained representation model BERT, which exploits BERT to generate substitute candidates and rank candidates. In the step of substitute generation, BERT-LS not only does not rely on any linguistic database and parallel corpus, but also fully considers both the given sentence and the complex word during generating candidate substitutions. In the step of substitute ranking, BERT-LS employs five efficient features, including BERT's prediction ranking, BERT-based language model and the paraphrase database PPDB, in addition to the word frequency and word similarity commonly used in other LS methods. Experimental results show that our approach obtains obvious improvement compared with these baselines, outperforming the state-of-the-art by 29.8 Accuracy points on three well-known benchmarks.

Key words Lexical simplification, substitution generation, substitution ranking, bidirectional encoder representations from transformers

Citation Qiang Ji-Peng, Qian Zhen-Yu, Li Yun, Yuan Yun-Hao, Zhu Yi. English lexical simplification based on pretrained language representation modeling. *Acta Automatica Sinica*, 2022, **48**(8): 2075−2087

在阅读资料时, 如果句子中包含不认识的词语, 将直接影响对文本内容的理解, 特别是阅读非母语的文本. Hirsh 等[1] 和 Nation 等[2] 的研究表明, 英语学习者需要熟悉文本中 95% 的词汇才能基本理解其内容, 熟悉 98% 的词汇才能轻易地进行阅读. 词汇简化 (Lexical simplification, LS) 任务要求在不改变文本的语义、不破坏文本语法结构的情况下降低文本的阅读难度, 常采用的方法是用更简单的词语替换句子中的复杂词语. 词语简化有助于降低文本的阅读难度, 针对的人群包括且不限于儿童[3]、非母语人士[4]、有阅读障碍的人[5−6] 等. 词语简化作为文本简化方法的一类, 已经有 20 多年的发展历史.

早期的 LS 系统主要使用人工制定或者自动学

收稿日期 2020-09-05 录用日期 2020-12-23
Manuscript received September 5, 2020; accepted December 23, 2020
国家自然科学基金 (62076217, 61906060, 61703362) 和江苏省自然科学基金 (BK20170513) 资助
Supported by National Natural Science Foundation of China (62076217, 61906060, 61703362) and Natural Science Foundation of Jiangsu Province (BK20170513)
本文责任编委 张民
Recommended by Associate Editor ZHANG Min
1. 扬州大学信息工程学院 扬州 225127
1. School of Information Engineering, Yangzhou University, Yangzhou 225127

习的简化规则来完成词汇简化任务[7]. 例如, 使用 WordNet 生成复杂词的简单同义词[8−10]. 从简单维基百科和普通的维基百科组成的平行语料库中提取复杂词语与简单词语的对应关系[11−13]. 但是这两类方法有很多的局限性. 除了语义词典数据库的制作成本高昂和平行语料库提取困难, 这些规则只能提供有限数量的复杂单词与部分简单同义词的对应关系, 不能够覆盖所有需要简化的单词, 也不能囊括所有合理的简单替换词.

为了解决上述问题, 最近的一些词汇简化方法使用词嵌入模型来获取目标复杂词的简单候选词, 选择在向量空间中与复杂词的词向量余弦相似度最高的一些词语作为候选替代词[14−16]. Glavaš 等[14] 在未注释的文本语料库中训练词嵌入模型, Paetzold 等[15−16] 在带有词性标签的文本上训练语境感知词嵌入模型. 这些方法解决了基于规则方法的局限性. 但是它们生成候选词时没有考虑复杂词的上、下文语境信息, 生成候选替代词集合中不可避免的生成了大量的虚假候选词.

本文提出了一种与已有 LS 系统完全不同的方法, 利用预训练语言表示模型 (Bidirectional encoder representations from transformers, BERT)[17] 获得复杂词的简单替代词. BERT 是无监督的通用语义表示模型, 使用掩码语言模型和下一句预测 2 个任务进行优化. 掩码语言模型通过随机掩码一定比例的输入, 然后根据上、下文对掩码的词进行预测, 这与 LS 任务中为目标复杂词生成符合语境的简单替代词的模式是可关联的. 本文将句子中的目标复杂词进行掩码后输入 BERT 模型进行预测, 从掩码词语的预测中选择高概率的词作为候选词, 并对它们进行排序. 具体方法是将两个原句进行串联, 随机掩盖前一个句子中一定比例的单词, 并对后一个句子的复杂词进行掩盖, 将其输入 BERT 模型, 预测出后一句中掩盖位置的词汇概率分布.

得到生成的候选替代词后, 基于预语言训练表示模型的词语简化方法 (BERT-LS) 使用 5 个特征对所有候选词进行排序, 选择平均值排名最高的词作为最佳替代词. 除了已有方法常使用的词频和候选替代词之间的相似度的特征外, 还结合了 BERT 本身的特色, 利用了 BERT 的预测顺序和基于 BERT 的语言模型作为特征, 还额外采用复述数据库 PPDB 作为特征. 最后, 最佳替代词是否替换原词需要考虑替代词与原词之间的简单程度和替代词与原有上、下文信息之间的流畅性.

图 1 展示了在词汇简化任务中, 两个基线系统 PaetzoldNE[16]、Rec-LS[18] 和 BERT-LS 对句子进行

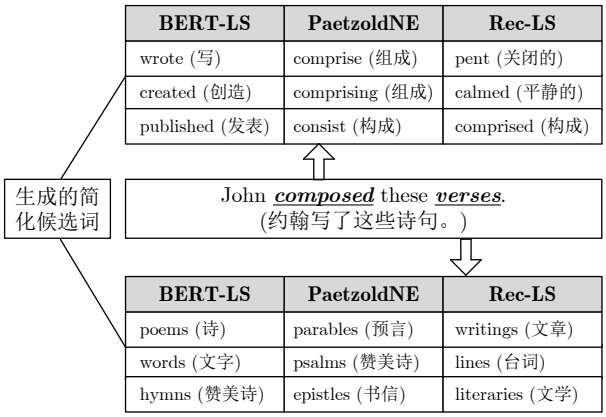


图 1 三种词语简化方法产生的候选替换词进行对比[16, 18]

Fig. 1 The substitution candidates generated by the three lexical simplification methods are compared[16, 18]

简化的实例. 对于句子 “John composed these verses.”中的复杂词 “composed”和 “verses”, 已有的 2 个 LS 系统在生成候选词时只关注复杂词本身, 而没有考虑上、下文语境, 因此这些系统没有能捕获复杂词在句子中的准确词意, 生成的候选词也不符合具体的语境. BERT-LS 生成的候选词不仅与复杂词在句子中的词义一致, 而且非常契合上、下文. 通过对生成的候选词进行排序后, 可以很容易地选择 “wrote”和 “poems”作为 “composed”和 “verses”的最终替代词. 替换后的句子 “John wrote these poems”不仅没有改变句意, 而且保持了句子的语法结构, 达到了句子简化的目的.

本文的主要贡献总结如下:

1) 提出了一种新的基于预语言训练表示模型的词语简化方法 BERT-LS, 充分利用 BERT 的优势来生成候选替代词. 从查阅到的已有文献可知, BERT-LS 是最先尝试利用预训练语言模型进行词语简化的方法. 与现有方法相比, 此方法不仅考虑了复杂词的上、下文信息, 而且生成的候选词无需考虑任何词形的变化.

2) 提出了一种新的候选词排序方法. 最先采用了 BERT 的预测排名和基于 BERT 的上、下文产生概率, 还首次利用复述数据库 PPDB 作为一个特征. 这些特征能够充分地考虑候选词本身的简单性和它们与句子的契合程度.

3)BERT-LS 在实验评估中优于基线算法, 候选词生成过程的性能比较之前最好的的方法的 F 值提升了 41%, 整体系统的效果在准确率上提升了 29.8%. 论文的源代码已公开在 https://github.com 和 qiang2100/BERT-LS.

1 相关工作

文本简化是在保留原有文本信息的情况下, 尽可能简化原有文本的内容, 从而使其更容易被更广泛的读者阅读和理解. 目前研究最多的是英文文本的简化, 本文也针对英文文本简化进行介绍. 文本简化主要针对文本中词汇和句法进行简化. 文本简化的早期方法主要利用基于统计的机器翻译系统学习复杂句子到简化的句子的映射[19−20]. 最近几年, 随着神经网络方法的快速发展, 许多方法采用编码器–解码器模型进行文本简化[21−22]. 以上这些文本简化方法都是有监督的文本简化方法, 其有效性严重依赖获取的平行语料的数量和质量. 从维基百科和简单维基百科中提取的平行语料 WikiSmall 和 WikiLarge 不仅样本数目不多, 还包含了很多不准确的简化 (没有对齐或仅部分对齐) 和不充分的简化 (简化句子不够简化)[23]. 因此, 很多学者只关注文本简化中的词语简化方法的设计.

一般情况下, 词汇简化包括复杂词识别、候选替代词生成、候选替代词选择和候选替代词排序 4 个过程. LS 旨在识别句子中的复杂词, 并为这些复杂词找到合适的替代词[24−25]. 好的替代词不仅要保留句子的句意, 符合句子的语法结构, 还要足够的简单, 因此这是一项非常具有难度的任务.

早期的词汇简化方法设定好一些规则, 然后按照规则对句子中的复杂词进行简化. 这些规则通常将复杂的单词与其简单的同义词进行关联[8, 26−27]. 为了构建规则, 这些系统通常从大型语言数据库中获得复杂单词的同义词, 例如 WordNet. 然后, 根据单词的频率[3, 7] 或单词的长度[28] 等特征在这些同义词中选择最简单的单词作为最佳替代词. 这类方法的缺点在于制作简化规则需要大量的人工注释, 并且手动制作的简化规则相对较为固定. 在实际应用中会出现一些复杂的单词没有被规则定义为 “复杂词”(即需要简化的单词), 在对句子进行简化时保留了这些复杂的单词. 还有一种情况是对包含复杂单词的句子进行简化时, 一些简单替代词被规则遗漏, 从而导致没有为该复杂单词找到合理的简化替代词.

随着 “复杂–简单”平行语言数据库的出现, 如维基百科和简单维基百科, LS 系统的主要模式由知识型简化转变为数据驱动型简化[9, 11, 13]. Yatskar 等[12] 从简单英语维基百科编辑中提取复杂术语的解释作为该复杂术语的简化替代. 这些方法已被验证可以生成很多合理的简化, 例如 “stands for” 简化为 “is the same as”, “indigenous” 简化为 “native”. Biran 等[11] 认为维基百科和简单维基百科中的每一对不同的单词都是可能的简化对, 他们去除了其中词形变化的单词, 并过滤了没有在 WordNet 中标记为同义词或上位词的单词对. Horn 等[13] 也从维基百科和简单维基百科中提取简化规则, 并采用语境感知的二元分类器来判断在特定语境中是否应用简化规则. 除了平行语料, 利用简化的 SimplePPDB 提取候选替代词的集合的方法也被提出, SimplePPDB 是复述数据库 PPDB 的一个子集. 这些基于平行语言数据库的方法存在的缺陷是它们严重地依赖平行语言数据库中 “简单”数据库的质量.

为了减少对语义词典和平行语言数据库的依赖, Glavaš 等[14] 提出了一种无监督的基于词嵌入模型方法. 词嵌入模型在通用的文本语料库中训练, 获得每个单词在分布式语义特征空间中的唯一向量表示. 他们选择与复杂词最相似的 10 个单词作为候选替代词. Paetzold 等[15−16] 提出了另一种方法, 他们在大型语料中训练上、下文感知的词嵌入模型, 即训练过程中使用通用词性标签对每个单词进行标记. 该方法在生成候选词后, 选择在向量空间中与复杂词词向量余弦相似度最高且词性标签相同的一些单词作为候选替代词, 解决了一部分歧义问题, 从而提高了 LS 系统的性能. Gooding 等[18] 联合语言数据库和词嵌入模型获取候选替代词集合. 这类方法提取的候选替代词范围特别大, 因为通过语义相似度得到的词语中, 不仅包含与复杂词相近的词还包含了很多相关的词语.

以上所有的词汇简化方法在产生复杂词的候选词的过程中, 都没有考虑复杂词的上、下文信息, 这不可避免的会产生大量的虚假替代词, 给词语简化后面的步骤带来很大干扰. 本文提出的 BERT-LS 基于预训练语言模型 BERT, 使用模糊的上、下文信息以及原词词义信息来生成候选词. 预训练语言模型[29−30] 已经引起了广泛的关注, 并被证明可以有效地提升很多自然语言处理相关任务的效果.由于 BERT-LS 的基本特性, 该方法可以很容易适用于其他语言.

2 无监督的词语简化

本节将具体对 BERT-LS 的工作原理进行介绍.

2.1 BERT 模型

BERT 是在一个大的文本语料库 (如维基百科) 中训练的通用的 “语言理解”模型, 通过掩码语言建模和下句预测 2 个训练目标进行优化. 不同于传统的语言建模是根据历史记录来预测序列中的下一个词, 掩码语言模型是根据序列中的上、下文语境来预测序列中的缺失词. BERT 通过在每一个句

子上预置一个特殊的分类令牌 [CLS], 并在句子上组合一个特殊的分离令牌 [SEP] 来完成下一句预测任务. 与 [CLS] 令牌相对应的最终隐藏状态被用作总序列表示, 可以从中预测出分类任务的标签, 或者是可能被忽略的标签.

BERT 在下游任务中的表现优于已有方法, 是第一个无监督的深度双向的预训练系统. 无监督意味着 BERT 只需要使用原始文本语料库进行训练.

2.2 候选词生成

给定一个句子 S 和复杂词 w, 候选生成步骤的目的是为词语 w 产生符合上、下文的候选词.

对句子 S 中目标复杂词 w 掩码之后输入到 BERT 的掩码语言模型模型进行预测, 则 BERT 在预测时仅仅从上、下文中获取信息, 而没有考虑到目标词本身的词意. 如果不掩盖目标复杂词, 则 BERT 会获得原词信息, 进而在预测中极大概率的出现原词, 使得系统无法获得更理想的候选词.

考虑到 BERT 模型的擅长处理句子对形式的数据, 主要因为 BERT 的其中一个优化目标下一句预测. 在 BERT-LS 中, 首先随机掩盖其中一定比例的单词 (排除复杂词 w) 后作为句子 S_1, 然后将句子中的目标复杂词进行掩盖后作为句子 S_2, 将 S_1 与 S_2 通过 [CLS] 和 [SEP] 符号进行串联后, 输入 BERT 获取目标复杂词掩码位置的单词概率分布. 考虑到 S_2 中已经包含了复杂词的上、下文信息, 对进行一定比例的掩盖的主要目的是降低上、下文信息的双重影响. 使用这样的方法, 不仅能够获得目标词的上、下文信息, 也获得了复杂词本身的词义信息, 从而提高了生成的候选词的质量. 最后, 从概率分布中选择前 10 个词作为候选词, 并剔除及其形态衍生词.

如图 2 所示, 在句子 “the cat perched on the mat.” (猫栖息在垫子上) (栖息)”, 使用 BERT-LS 可以得到排名前 3 的候选词 “sat (坐), hopped (跳), landed (落)”. 如果采用现有的最先进的基于词嵌入的方法[14] 生成替换词, 前 3 个替换词分别是 “atop (在…上), overlooking (俯瞰), precariously (摇摇晃晃地)”. 很容易发现, BERT-LS 生成的候选词质量更高.

2.3 候选词排序

候选词排序的目的是选择最简单且最符合语境的替代词作为替换词[25]. 本文选用了多个特征对候选替代词进行排序, 除了常用的词义相似度和词频特征以外, 本文还选用了 BERT 输出的预测排名、基于 BERT 的上、下文产生概率和 PPDB 复述数据库 3 个特征. 候选词的排名名次分别是从 1 到 n, 其中 n 表示候选词的数目. 候选词的最终排名是所有排名的平均值, 选择名次最高的候选词作为最佳候选替代词.

1) BERT 输出的预测排名. BERT 输出的预测排序特征本身就考虑到候选词和上、下文之间的连贯性, 还有候选词和复杂词之间的关联性. 根据概率大小对生成的有效候选词进行排名, 概率越高则说明生成的候选词与句子的关联度越高.

2) 基于 BERT 的上、下文产生概率. 该特征主要是验证候选替代词与原有上、下文信息之间的连贯性. 由于 BERT 是利用掩码语言模型进行优化, 无法和传统的语言模型一样直接计算连续的几个词语产生概率. 考虑到 BERT 能够很好地利用上、下文信息预测掩码的词语. 本文采用一种新的方式计算上、下文的产生概率.

首先, 用候选替代词替换原词, 选择一个上、下文窗口. 假设从长度为 n 的原句 S 中选择复杂词 w_j 的前后 m 个词作为上、下文, 组成窗口 $\boldsymbol{WIN} = \{w_{j-1m}, \cdots, w_{j-1}, w_j, w_{j+1}, \cdots, w_{j+m}\}$, 其中 $j-m \geq 0$ 和 $j+m < n$. 用候选替代词 c 替代 $\boldsymbol{WIN}$ 中的复杂词 w, 组成新的序列 $\boldsymbol{WIN'} = \{w_{j-m}, \cdots, w_{j-1}, c, w_{j+1}, \cdots, w_{j+m}\}$.

接着, 从前到后依次掩藏 $\boldsymbol{WIN'}$的每一个字 w_i, 输入 BERT 并用公式计算掩藏后序列的交叉熵

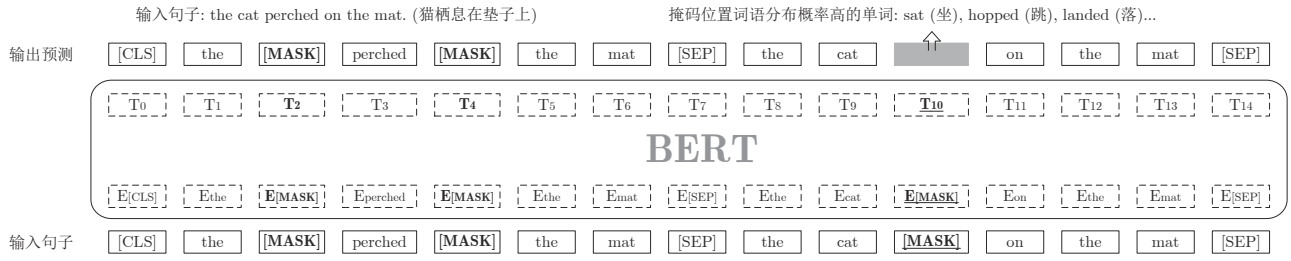


图 2 BERT-LS 使用 BERT 模型生成候选词, 其中输入为 “the cat perched on the mat”

Fig. 2 BERT-LS uses the BERT model to generate candidate words, and the input is “the cat perched on the mat”

损失值:

$$loss\,(w_i) = -\sum_{i=1}^{n} w_i \ln y_i$$

式中, w_i 是词语 w_i 在词汇表中 one-hot 表示, y_i 表示 BERT 在掩藏位置预测的词语的概率分布.

最后, 对整个窗口的所有词的损失值求平均值作为该窗口的损失值. 该值越小代表上、下文之间的连贯性越好. 最后, 对所有的候选替代词的损失值按照从小到大进行排序. 实验中选择的窗口大小为 11, 即原词左右各 5 个词语.

3) 词义相似度. 该特征是考量候选替代词和原词之间相似度. 一般情况下, 相似度越高表示关联度越大. 本文获取词语的向量选用了预训练词向量模型 fastText[31]. 假设候选词 c 和复杂词 w 的向量表示 $\boldsymbol{v}_c$ 和 $\boldsymbol{v}_w$. 通过计算复杂词词向量与候选替代词词向量的余弦相似度对候选替代词进行排名, 计算公式如下;

$$\cos\,(\boldsymbol{v}_c,\,\boldsymbol{v}_w) = \frac{\sum_{j=1}^{g} v_c^h v_w^h}{\sqrt{\sum_{j=1}^{g}\left(v_c^j\right)^2} \times \sqrt{\sum_{j=1}^{g}\left(v_w^j\right)^2}}$$

式中, g 为词嵌入模型中向量的维数.

4) 词频. 词语在大文本语料中的频率是判断词语难易程度简化法最常用的方法之一. 本文使用 SUBTLEXus[32] 的 Zipf 值作为单词的词频特征, 该分值越高表示词频越高. SUBTLEXus 从美国英语电影字幕中提取, 总计 5 100 万个单词, 该语料库中的词频比其他大型语料库中的词频更能反应人们对单词简单性的认知. Zipf 值的范围为 1 ~ 7, 其中 1 ~ 3 表示低频词 (频率为 1/100 万字及以下), 4 ~ 7 表示高频词 (频率为 10/100 万字及以上).

5) PPDB 特征. PPDB[33] 中包含超过 1 亿个英文单词或短语对, 这些词组对的提取是使用了一种二语旋转技术, 即假设两个英语短语翻译成相同的外文短语具有相同的意义. 一些 LS 方法使用 PPDB 或其子集 SimplePPDB[26] 中包含的简化规则生成候选词. 由于 BERT-LS 在候选词生成过程中比 PPDB 以及 SimplePPDB 表现更好, 考虑到 PPDB 中包含有效的简化规则, 因此本文首次尝试使用 PPDB 提供的简化规则对候选词进行排序. 假设上一步产生 n 个候选词, 正常情况下获取的排名是从 1 到 n. 本文采用的一种非常简单的策略. 如果生成的候选替代词和复杂词组成的规则存在于 PPDB 中, 则将该候选词对应的排名值设置为 1; 否则, 则将该词对应的排名值置为候选词数目的 1/3. 本文设置 1/3 的主要原因是不想把不在 PPDB 中的候选词与在 PPDB 的候选词的排名拉开太大. 如果这两种情况下的候选词的排名拉开太大, 就会使得其他排序特征的影响力降低. 如果对这两种情况排名差别很小的话, 该特征的影响就会降低. 在实验过程中, 该方法选择的候选词数目是 10, 对在 PPDB 中候选词, 给出的排名是第 1 名; 对不在 PPDB 中的候选词, 给出的排名是第 3 名. 在未来的工作中, 将致力于在候选词的排序中设计更合适的策略, 如利用 PPDB 中提供的已有的值.

2.4　BERT-LS 方法

词语简化方法第 1 步一般是复杂词识别. 复杂词识别一般作为单独任务出现, 本文没有给出具体的方法. 可以采用传统的词频方法, 利用大语料统计词的频率, 低于指定阈值的作为复杂词. 也可以根据具体的词典设定的等级判断词语的复杂度,如欧洲委员会设置的《欧洲共同语言参考框架》[34] 对英文词汇分了 A1、A2、B1、B2、C1 和 C2 六个标准, 其中 A1 ~ B1 面向语言水平较低的学生和 B2 ~ C2 提供了语言水平较高的词汇可以作为复杂词. 也可以采用一些机器学习的方法[35].

算法 1. BERT-LS

输入. 句子 S, 复杂词 w.

输出. 复杂词 w 的替代词.

1) Randomly mask a certain percentage of words in S excluding w as S_1;

2) Replace word w of S into [MASK] as S_2;

3) Concatenate S_1 and S_2 using [CLS] and [SEP] as SS;

4) p($\cdot|S_1$,S_2,\{w\}))←BERT(SS);

5) scs←$top\_probability$(p($\cdot|S_1$,S_2,\{w\})));

6) scs←excluding_morphological(scs, w);

7) for each feature f do;

8) $scores$← ∅;

9) for each sc ∈ scs do;

10) $scores$←$scores$ ∪ $f(sc)$';

11) end for;

12) $rank$←$rank\_numbers(scores)$;

13) all_ranks←all_ranks∪$rank$;

14) end for;

15) avg_rank←average(all_ranks);

16) $best$←argmax$_{sc}$(avg_rank);

17) if freq($best$)>freq(w) or loss($best$)<$loss$(w) do;

18) return $best$;

19) else return w.

当给定句子 S 和识别出的复杂词 w 以后, 采用算法 1 BERT-LS 方法来执行词语的简化. 首先, 利用 BERT-LS 获取候选替换词生成 (原理在第 2.2 节介绍). 先对句子进行一定的掩码和替换, 符合 BERT 的输入要求, 本文采用了双句输入 (步骤 1) ~ 步骤 3). 然后, 利用 BERT 获取 [MASK] 对应的词语产生概率 (步骤 4)), 选择该高概率的词语作为替换词 (步骤 5)), 并移除复杂词和其形态衍生词 (步骤 6)).

然后进行候选词排序, 具体理论在第 2.3 节中介绍. 分别采用 5 个特征依次对候选词进行排序, 并对排名求和 (步骤 7) ~ 14)). 接着, 对所有候选替代词排序求平均值 (步骤 15)), 选择名次最高的作为最佳替代词 (步骤 16)). 最后,对目标复杂词与最佳候选替代词进行基于简单性的比较, 判断是选择替代词还是保留原词 (步骤 17) ~ 19)). 如果最佳候选替代词的词频高于目标复杂词 (词频特征), 或者最佳候选替代词的损失低于目标复杂词 (基于 BERT 的上、下文产生概率), 则选择最佳候选替代词作为替代词, 否则不进行替代. 本文选用词频特征和基于 BERT 的上、下文产生概率特征, 主要因为词频特征代表着词语的简单性, 上、下文产生概率特征代表着句子的流畅性.

3 实验

本节通过官方的评测数据集进行实验, 主要验证以下问题:

1) 候选词生成的有效性: BERT-LS 与其他候选词生成方法相比是否具有优势?

2) LS 系统的有效性: 在对完整 LS 系统评估的过程中, BERT-LS 是否优于其他的系统?

3) 影响 BERT-LS 系统的因素: 对 BERT-LS 进行消融实验, 验证使用不同参数和模型进行实验对 BERT-LS 系统的影响.

4) 对 BERT-LS 系统进行错误分析: 对 BERT-LS 输出的候选词和简化句子进行错误分析, 探究造成 BERT-LS 产生错误的可能原因.

3.1 实验设置

标注的词语简化数据集通常由一个句子、指定的复杂词和人工标注的多个可选的替代词组成. 本文选用了被广泛使用的 3 个词汇简化评估数据集, 具体细节描述如下:

1) LexMTurk[13]: 从维基百科中选出的 500 个英语实例. 每个实例由句子、目标词和候选词组成, 候选词是由 50 名英语作为母语的亚马逊劳务众包平台 “Amazon Mechanical Turk”的用户 “turker” 进行标注完成. 由于是英语作为母语的人士进行标注, 数据集整体的简化率是非常高的. 唯一不足的是有部分标注存在拼写错误.

2) BenchLS[4]: 由 929 个英语实例组成, 由 LSeval 和 LexMTurk 两个数据集结合而成, 因此 BenchLS 包含最多的目标词. LSeval 包含 429 个实例, 包含 46 位 “turker”和 9 名博士为 SemEval 2007 词汇替换任务的数据集的制作的简单性排序. 创建 LSeval 时使用密集注释过程, 确保标注的简化词比目标词简单. 该数据集也存在一些拼写错误.

3) NNSeval[25]: 由 239 个英语实例组成. 该数据集是 BenchLS 的删减版, 利用母语非英语人士进行以下两种过滤: 过滤实例 (目标复杂词被认为不是复杂词); 过滤替代词 (候选词被认为是复杂词). 相对其他数据集, NNSeval 更准确地满足了母语非英语的人士. 因为使用了过滤技术, 目标复杂词的数量和候选替代词的覆盖范围都比其他数据集小.

实验选择了 9 个算法进行比较. 根据候选替换词生成的策略不同, 对比算法分为以下 5 类:

1) 语义词典: Devlin[7] 使用 WordNet, Kajiwara 等[36] 使用 Merriam 词典, Pavlick 等[26] 使用复述数据库.

2) 平行语料库: Biran 等[11] 和 Horn 等[13] 使用维基百科和简单维基百科提取的平行句子对.

3) 词嵌入模型: Glavaš 等[14] 使用普通的词嵌入模型, Paetzold 等[4] 使用上、下文感知的词嵌入模型.

4) 混合方法: Paetzold 等[16] 使用平行语料和词向量模型, Devlin 等[17] 使用语言数据库和词嵌入模型.

5) BERT 的方法: 本文方法为 BERT-LS. 为了更好地进行对比, 针对单个句子的目标词掩码后作为输入的方法, 称之为 Bert-Single.

REC-LS 方法使用了原论文中提供的代码进行实验. 在实验中, 预训练 BERT 模型采用 BERT-Large, 全词掩码 模型 (网址 https://github.com/google-research/bert). BERT-LS 方法中, 第 1 句中的上、下文信息选用掩盖比例默认为 50%, 生成的候选词的数目是 10, 候选词排序中 BERT 上、下文产生概率中窗口为 5.

3.2 候选替代词生成评估

假设测试集合有 m 为样例, 其中第 i 个样例对应的复杂词为 w_i, 人工标注的替代词集合 p_i, 算法产生的候选替代词集合为 q_i, 用 $\#(p_i)$ 和 $\#(q_i)$ 分别表示 p_i 和 q_i 集合中词的数目.

候选替代词生成的通常使用以下 3 个指标进行

评估:

1) 精确率: 生成的候选替代词中属于人工标注的词占候选替代词总数目的比例:

$$Precision = \frac{\sum_{i=1}^{m} \#(p_i \cap q_i)}{\sum_{i=1}^{m} \#(q_i)}$$

式中, $p_i \cap q_i$ 表示两个集合中共同的词语集合.

2) 召回率: 生成的候选替代词中属于人工标注的词占所有人工标注替代词总数目的比例:

$$Recall = \frac{\sum_{i=1}^{m} \#(p_i \cap q_i)}{\sum_{i=1}^{m} \#(p_i)}$$

3)F 值: 精确率和召回率的调和平均值:

$$F = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

表 1 列出了 BERT-LS 和对比方法的实验结果. 由表 1 可以看出, BERT-LS 在每个数据集上都取得了最高的 F 值, 相较于对比方法中取得最高 F 值的 PaetzoldNE 方法, 分别提升了 37.4%、19.1% 和 41.4%. 在 PaetzoldNE 方法中, 当一个候选词出现在 “标签”中, 则该词的所有形态变换词都计算在内, 因此获得了 2 个最高精确率, 这点从召回率较低上也能够得出.

BERT-LS 在没有增加额外的步骤的情况下, 生成的候选替代词考虑了词的形态问题, 候选替代词中几乎没有不同形态的同一词语. 基于词向量的方法中, 同一词语的不同形态间的向量相似度很高, 因此都可能出现, 导致了召回率较低. 对比 BERT-LS 和 BERT-Single 的结果, BERT-LS 取得了更好的效果, 因为 BERT-Single 在生成候选词时只考虑了句子的上、下文信息而没有考虑目标复杂词的词义, 很多与目标复杂词词义差异很大的词会被作为候选词.

3.3　完整的 LS 系统评估

对 LS 系统进行评估, 使用与候选词生成过程同样的数据集. 假设测试集合有 m 个样例, 其中第 i 个样例对应的复杂词为 w_i, 人工标注的替代词集合 p_i, 算法最后产生的替代词为 t_i.

通常采用的评估指标有以下 2 个:

1) 精确率: 所有样本中最终选择的替代词是目标词或属于人工标注词中的比率;

$$Precision = \frac{\sum_{i=1}^{m} \left(1_{(t_i=w_i)} \| 1_{(t_i \in p_i)}\right)}{m}$$

当 $1_{(t_i=w_i)}$ 成立时, $1_{(t_i==w_i)}$ 为 1, 否则为 0; 当 t_i 属于 p_i 集合时, $1_{(t_i \in p_i)}$ 为 1, 否则为 0.

2) 准确率: 所有样本中最终替代词不是目标词但在人工标注中的比率.

从这 2 个指标可以看出, 如果不进行简化则精确率为 1, 准确率为 0. 如果复杂词全部采用候选词进行替换, 则精确率和准确率具有相同的值.

$$Accuracy = \frac{\sum_{i=1}^{m} 1_{(t_i \in p_i)}}{m}$$

表 2 显示了各个简化系统的结果. BERT-LS 在所有数据集上都获得了最高的准确率, 相较于 PaetzoldNE, 分别提升了 17.2%、41.9% 和 30.1% 的性能. Rec-LS 取得了较高的精确率和较低的准确率, 这是因为该方法在大部分简化中使用原词作为

表 1　候选词生成过程评估结果

Table 1　Evaluation results of candidate word generation process

方法	LexMTurk			BenchLS			NNSeval		
	精确率	召回率	F值	精确率	召回率	F值	精确率	召回率	F值
Yamamoto	0.056	0.079	0.065	0.032	0.087	0.047	0.026	0.061	0.037
Biran	0.153	0.098	0.119	0.130	0.144	0.136	0.084	0.079	0.081
Devlin	0.164	0.092	0.118	0.133	0.153	0.143	0.092	0.093	0.092
Horn	0.153	0.134	0.143	0.235	0.131	0.168	0.134	0.088	0.106
Glavaš	0.151	0.122	0.135	0.142	0.191	0.163	0.105	0.141	0.121
PaetzoldCA	0.177	0.140	0.156	0.180	0.252	0.210	0.118	0.161	0.136
PaetzoldNE	**0.310**	0.142	0.195	**0.270**	0.209	0.236	0.186	0.136	0.157
Rec-LS	0.151	0.154	0.152	0.129	0.246	0.170	0.103	0.155	0.124
BERT-Single	0.253	0.197	0.221	0.176	0.239	0.203	0.138	0.185	0.158
BERT-LS	0.306	**0.238**	**0.268**	0.244	**0.331**	**0.281**	**0.194**	**0.260**	**0.222**

最佳替换词, 并没有实现有效的简化. 整体而言, 整个简化系统的结果与候选词生成的结果类似, 证明了本文提出的排序方法是有效的.

表 2 整个简化系统评估结果
Table 2 Evaluation results of the whole simplified system

方法	LexMTurk		BenchLS		NNSeval	
	精确率	准确率	精确率	准确率	精确率	准确率
Yamamoto	0.066	0.066	0.044	0.041	0.444	0.025
Biran	0.714	0.034	0.124	0.123	0.121	0.121
Devlin	0.368	0.366	0.309	0.307	0.335	0.117
PaetzoldCA	0.578	0.396	0.423	0.423	0.297	0.297
Horn	0.761	0.663	0.546	0.341	0.364	0.172
Glavaš	0.710	0.682	0.480	0.252	0.456	0.197
PaetzoldNE	0.676	0.676	0.642	0.434	0.544	0.335
Rec-LS	0.784	0.256	**0.734**	0.335	**0.665**	0.218
BERT-Single	0.694	0.652	0.495	0.461	0.314	**0.285**
BERT-LS	**0.864**	**0.792**	0.697	**0.616**	0.526	**0.436**

3.4 影响 BERT-LS 系统的因素

1) 排序特征对系统整体性能的影响

为了确定每一组特征的作用, 本文进行了消融实验, 验证 BERT-LS 使用的 5 个特征对候选词排序的影响, 结果如表 3 所示.

由表 3 可知, 使用全部 5 个特征, BERT-LS 在 3 个数据集的平均精确率和准确率分别为 0.696 和 0.615, 兼顾了精确率与准确率. 词义相似度与词频特征在之前的简化系统中也常被使用. 从结果可以看出, 如果减少任何 1 个特征会降低方法的精确率和准确率, 而且这 2 个特征对候选词的排序影响最大. BERT 输出的预测排名、基于 BERT 的上、下文产生概率和 PPDB 特征是该方法首次使用. 通过观察 3 个数据集的平均结果, 如果不使用 BERT 输出的预测排名, BERT-LS 的精确率和准确率降为 0.662 和 0.608; 如果不使用基于 BERT 的上、下文产生概率, 系统的精确率和准确率降为 0.686 和 0.593; 如果不使用 PPDB 特征, 系统的精确率和准确率降为 0.679 和 0.606. 可以看出, 这 3 个首次使用的特征对候选词的排序都是有效的. 同时, PPDB 对 BERT-LS 的影响最小, 可能是由于采用的 PPDB 策略简单, 未来研究将尝试使用不同的策略引入 PPDB.

总之, 5 个特征都为模型的性能带来了提升, 都能有助于提高方法的整体性能. 由于这些特征对排序过程带来的影响不同, 未来研究可通过加权的方法获得最佳排名效果, 而不是使用平均排名的方式.

2) 不同 BERT 模型对系统的影响

为了探究不同 BERT 模型对系统的影响, 本文在在 LexMturk 数据集上使用 3 种不同的 BERT 模型进行实验:

a) BERT-base (Base): 12 层模型, 参数量为 110 M, 其中 M 表示 1 百万;

b) BERT-large (Large): 24 层模型, 参数量为 340 M;

c) BERT-large (全词掩码): 24 层模型, 参数量为 340M. 本文使用的全词掩码是一种特殊处理策略, 对词进行掩码时, 掩盖整个词语, 而不是一个部分. 已公开的工作中[29−30] 展示了使用全词掩码的方式预训练模型能够提升模型在 NLP 下游任务中的表现.

表 4 显示了使用不同 BERT 模型在三个数据集上进行实验的结果. 由表 4 可以看出, 在完整系统评估中, 全词掩码模型的系统获得了最高的精确率和准确率, 相对 Large 模型的提高还是显而易见的. 此外, Large 模型的性能优于 Base 模型. 由此可以简单得出, 更好的 BERT 的模型有助于提供简化系统的性能. 如果有更好的 BERT 模型, 可以尝

表 3 不同特征对候选词排序的影响
Table 3 The influence of different features on the ranking of candidates

方法	LexMTurk		BenchLS		NNSeval		平均值	
	精确率	准确率	精确率	准确率	精确率	准确率	精确率	准确率
BERT-LS	0.864	0.792	0.697	0.616	0.526	0.436	0.696	0.615
仅用 BERT 预测排名	0.772	0.608	0.695	0.502	0.531	0.343	0.666	0.484
去除 BERT 预测排名	0.834	0.778	0.678	0.623	0.473	0.423	0.662	0.608
去除上下文产生概率	0.838	0.760	0.706	0.614	0.515	0.406	0.686	0.593
去除相似度	0.818	0.766	0.651	0.604	0.473	0.418	0.647	0.596
去除词频	0.806	0.670	0.709	0.550	0.556	0.397	0.691	0.539
去除 PPDB	0.840	0.774	0.682	0.612	0.515	0.431	0.679	0.606

表 4　使用不同的 BERT 模型的评估结果
Table 4　Evaluation results using different BERT models

数据集	模型	候选词生成评估			完整系统评估	
		精确率	召回率	*F*值	精确率	准确率
LexMTurk	Base	0.317	0.246	0.277	0.746	0.700
	Large	**0.334**	**0.259**	**0.292**	0.786	0.742
	全词掩码	0.306	0.238	0.268	**0.864**	**0.792**
BenchLS	Base	0.233	0.317	0.269	0.586	0.537
	Large	**0.252**	**0.342**	**0.290**	0.636	0.589
	全词掩码	0.244	0.331	0.281	**0.697**	**0.616**
NNSeval	Base	0.172	0.230	0.197	0.393	0.347
	Large	0.185	0.247	0.211	0.402	0.360
	全词掩码	**0.194**	**0.260**	**0.222**	**0.526**	**0.436**

试替换掉本文使用的 BERT 模型, 达到进一步提高系统性能的目的.

3) 掩码比例对系统整体性能的影响

为了探究不同掩码比例对系统的影响, 本文将掩码比例分别设置为 0% ~ 90%, 在 LexMturk 数据集上进行实验, 并使用 5 次实验的平均值作为结果. 由于输入 2 个相同的句子到 BERT 中, 第 2 句已经包含了复杂词的上、下文信息, 第 1 句除了包含复杂词还包含上、下文信息. 为了降低上、下文信息的重复使用, 采用了对第 1 句的上、下文进行掩码.

由图 3 可以看出, 随着掩码比例的提升, 算法的性能有略微的变化, 候选词生成过程大概在 50% ~ 80% 时取得了最高的精确率、召回率和 F 值. 由图 4 可以看出, 整个系统的准确率在掩码比例为 50% 左右时获得了最高值.

4) 生成候选词的数量对系统性能的影响

为了探究生成候选词的数量对系统的影响, 本文在 LexMturk 数据集上进行实验, 将生成候选词的数量依次设定为 5 的 1 到 5 倍. 图 5 显示了生成候选词的数量对系统带来的影响.

由于人工标注的 “标签” 中提供的简化词的数量是固定的, 随着生成候选词数量的增加, 精确率必定会降低, 召回率一定会升高. 从候选替代词的生成结果中, 在生成词数量为 10 时, F 值取得了最大值. 同时在对系统的评估中, 精确率与准确率在候选词数量为 20 时取得最好的结果, 之后已逐步稳定. 由此可知, BERT-LS 系统拥有良好的稳定性与鲁棒性.

3.5　错误分析

1) 候选替代词生成的错误分析

第 3.2 ~ 3.4 节实验都是定量的对 BERT-LS

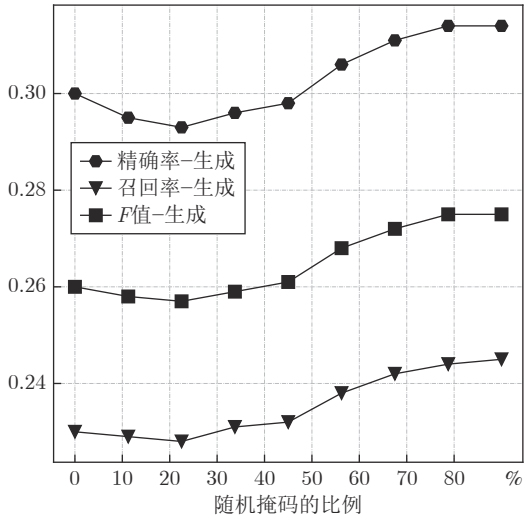


图 3　不同的掩码比例对系统的影响
Fig. 3　The influence of different mask proportion on the system

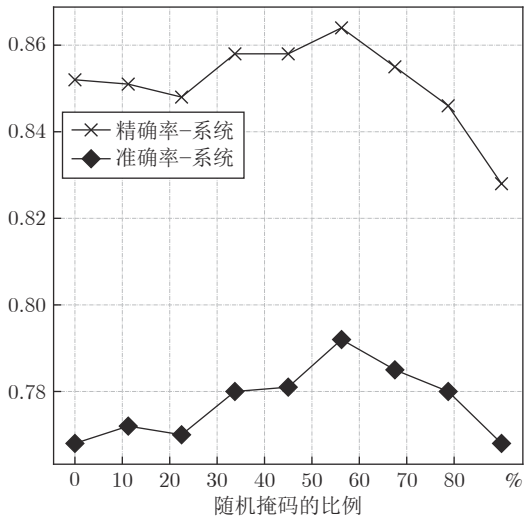


图 4　不同的掩码比例对系统的影响
Fig. 4　The influence of different mask proportion on the system

进行分析, 本文将定性的分析 BERT-LS 的结果. 当使用 LexMturk 数据集进行实验的过程中, 当生成候选词的数量为 10 时, 生成的候选词没有一个是 “标签” 中的合理简化替代词的比例仅为 1.6%, 即 BERT-LS 只在 8 个句例中没有产生任何一个有效的替代词. 当生成的候选词数量为 15 时, 该比例降为 0.8%; 当生成的候选词数量为 30 时, 该比例降为 0.2%.

在本节中, 对生成候选词数量为 10 时, 获得的候选词没有一个出现在 “标签” 中的原因进行具体分析. 表 5 展示了 8 个句例, 给出了原句、人工标注

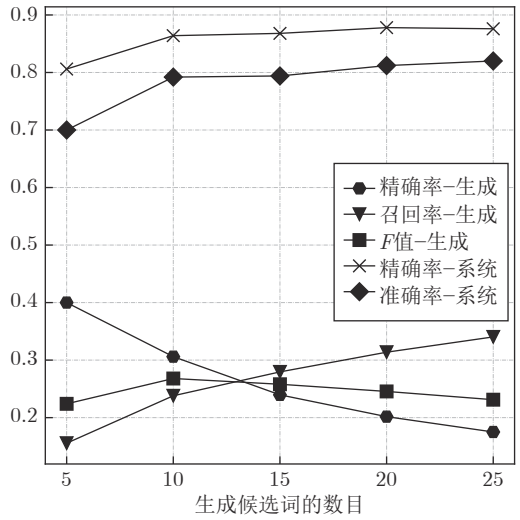


图 5 不同生成候选词数量的评估结果

Fig. 5 Evaluation results of different number of candidate words generated

的 “标签”、排序后的候选替代词和最终替代词, 复杂词用加粗和下划线标记.

由表 5 可以看出, 句子 5 中的目标复杂词 “demonstrated”在语境中有 “证明、证实、示范、演示”的意思, 但人工标注给出的注释都是 “展示”的含义, 这显然是不合理的, 而生成词中的 “proved (证明)”是合理的简化替代词; 句子 7 中, “dynamic energy”也被翻译为 “动能”, 因此 “dynamic (动力的)”也是合理的简化替代词; 句子 8 中, 生成的 “altered (修改)”, “modified (调整)”也是合理的简化替代词. 这 3 个句例是由于人工注释不完备导致的.

而其余的 5 个句例 (句 1、句 2、句 3、句 4 和句 6) 没有生成任何一个合适的候选词. 如果生成的 “最终替代词”不是原词的话, 该词语简化就可能改变句子原有的意思. 从整体上看, BERT-LS 为这些句子产生的候选词放在原有的上、下文环境中都是非常流畅的. 可能是因为 BERT-LS 生成候选词时是考虑在该位置最有可能出现的词, 从而生成和原词意思无关甚至相悖的候选词. 也有少数产生的候选词出现异常, 如在第 7 句中生成的候选词有 “the, momentum, velocity”. 这样的情况是非常少的, 但是这个例子前几个候选词还是符合上、下文的, 可能的原因就是训练 BERT 的语料缺乏大量的针对该句子的上、下文情况. 在未来的工作中, 计划针对以上出现问题, 从以下 2 个方面展开工作: 1) 在 BERT 生成 “[MASK]”位置的词语中引入适当的同义词的语义知识, 使其更多关注生成词与原有词之间的相似度; 2) 获取词语简化的平行数据, 利用平行数据对 BERT 进行微调, 然后利用微调后的 BERT 模型进行词语简化的候选词生成.

2) 候选词排序的错误分析

BERT-LS 几乎所有的样本都能找到一个或者多个合适的替代词, 但是最终系统结果并不能总选择最合适的候选替代词作为最终替代词, 本节将定

表 5 LexMTurk 数据集中的简化句例

Table 5 Simplified sentences in LexMTurk

句子	原句; 标签; 生成词; 最终
句 1	Much of the water carried by these streams is **diverted;** Changed, turned, moved, rerouted, separated, split, altered, veered, …; transferred, directed, discarded, converted, derived; transferred
句 2	Following the death of Schidlof from a heart attack in 1987, the Amadeus Quartet **disbanded;** dissolved, scattered, quit, separated, died, ended, stopped, split; formed, retired, ceased, folded, reformed, resigned, collapsed, closed, terminated; formed
句 3	…, apart from the **efficacious** or prevenient grace of God, is utterly unable to…; ever, present, showy, useful, effective, capable, strong, valuable, powerful, active, efficient, …; irresistible, inspired, inspiring, extraordinary, energetic, inspirational; irresistible
句 4	…, **resembles** the mid-19th century Crystal Palace in London; mimics, represents, matches, shows, mirrors, echos, favors, match; suggests, appears, follows, echoes, references, features, reflects, approaches; suggests
句 5	…who first **demonstrated** the practical application of electromagnetic waves,…; showed, shown, performed, displayed; suggested, realized, discovered, observed, proved, witnessed, sustained; suggested
句 6	…a well-defined low and strong wind gusts in squalls as the system **tracked** into…; followed, traveled, looked, moved, entered, steered, went, directed, trailed, traced…; rolled, ran, continued, fed, raced, stalked, slid, approached, slowed; rolled
句 7	…is one in which part of the **kinetic** energy is changed to some other form of energy…; active, moving, movement, motion, static, motive, innate, kinetic, real, strong, driving…; mechanical, total, dynamic, physical, the, momentum, velocity, ballistic; mechanical
句 8	None of your watched items were **edited** in the time period displayed; changed, refined, revise, finished, fixed, revised, revised, scanned, shortened; altered, modified, organized, incorporated, appropriate; altered

表 6 LexMTurk 数据集中的简化句例
Table 6 Simplified sentences in LexMTurk

句子	原句; 标签; 生成词; 最终
句 1	Triangles can also be **classified** according to their internal angles, measured here in degrees; grouped, categorized, arranged, labeled, divided, organized, separated, defined, described ...; **divided**, **described**, separated, designated; classified
句 2	...; he **retained** the conductorship of the Vienna Philharmonic until 1927; kept, held, had, got maintained, **held**, **kept**, remained, continued, shared; maintained
句 3	..., and a Venetian in Paris in 1528 also **reported** that she was said to be beautiful; said, told, stated, wrote, declared, indicated, noted, claimed, announced, mentioned; **noted**, confirmed, described, claimed, recorded, **said;** reported
句 4	..., the king will **rarely** play an active role in the development of an offensive or ....; infrequently, hardly, uncommonly, barely, seldom, unlikely, sometimes, not, seldomly...; never, usually, **seldom**, **not**, **barely**, **hardly;** never

性的分析可能的原因. 表 6 为未能达成有效简化的句例. 主要有两种情况导致找的最终替代词不能满足要求. 第 1 种情况是排序策略没有选择最佳替代词, 而是选择了原词. 由表 6 可知, 句 1 与句 3 生成的最佳替代词是合理的简化替代词, 同时也在标签中, 但是系统经过比较词频与交叉熵损失后并没有使用最佳替代词而是使用原词作为最终替代词. 这是因为句 1 中的生成词 “divided”的词频 Zipf 值为 3.649769, 而原词 “classified”的词频 Zipf 值为 3.83091, 系统认为 “classified”比 “divided”更加简单; 句 3 也是这样, 生成词 “noted”的 Zipf 值为 3.682769, 而原词 “reported”的 Zipf 值为 4.179451. 因此在句 1 和句 3 中未能使用最佳替代词作为最终替代词. 第 2 种情况是排序策略没有选择最合适的替代词. 在句 2 和句 4 中, 生成的最佳候选词并没有出现在标签中, 但是排名更靠后的候选替代词在标签中出现. 经检查发现, 句 2 和句 4 中的最佳候选词也是合理的简化候选词, 但是在标签中没有出现. 对更多输出的结果进行分析发现, BERT-LS 在很多样例中产生了正确的结果, 由于标注数据集的原因导致没有识别出来. 因此, BERT-LS 在候选词生成和排序中比在官方数据集上展示的结果有着更好的性能.

4 结束语

本文提出了一种基于预语言训练表示模型的词语简化方法 BERT-LS, 利用 BERT 的掩码语言模型进行候选词的生成和排序. 在不依赖平行语料库或语言数据库的情况下, BERT-LS 在生成候选替换过程中既考虑了复杂词又考虑了复杂词的上、下文. 在 3 个的基准数据集上进行实验, 实验结果验证了 BERT-LS 取得了最好的性能. 由于 BERT 只利用了原始文本上进行训练, 针对不同语言的 BERT 模型 (如中文、德语、法语和日语等) 也被提出来, 因此该方法可以应用到对应语言中进行词语简化.

BERT-LS 的一个限制是只能生成一个词而不是多个词来替换复杂的词. 下一步计划扩展 BERT-LS 支持多个词的替代, 进一步提高模型的实用性.

References

1 Hirsh D, Nation P. What vocabulary size is needed to read unsimplified texts for pleasure? *Reading in a Foreign Language*, 1992, **8**(2): 689−696

2 Nation I S P. *Learning Vocabulary in Another Language*. Cambridge: Cambridge University Press, 2001.

3 De Belder J, Moens M F. Text simplification for children. In: Proceedings of the SIGIR Workshop on Accessible Search Systems. Geneva, Switzerland: ACM, 2010. 19−26

4 Paetzold G H, Specia L. Unsupervised lexical simplification for non-native speakers. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence. Phoenix, USA: AAAI, 2016. 3761−3767

5 Feng L J. Automatic readability assessment for people with intellectual disabilities. *ACM SIGACCESS Accessibility and Computing*, 2009, (93): 84−91

6 Saggion H. Automatic text simplification. *Synthesis Lectures on Human Language Technologies*, 2017, **10**(1): 1−137

7 Devlin S. The use of a psycholinguistic database in the simplification of text for aphasic readers. *Linguistic Databases*, 1998

8 Lesk M. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In: Proceedings of the 5th Annual International Conference on Systems Documentation. New York, USA: ACM, 1986. 24−26

9 Sinha R. UNT-SimpRank: Systems for lexical simplification ranking. In: Proceedings of the 1st Joint Conference on Lexical and Computational Semantics. Montreal, Canada: ACL, 2012. 493−496

10 Leroy G, Endicott J E, Kauchak D, Mouradi O, Just M. User evaluation of the effects of a text simplification algorithm using term familiarity on perception, understanding, learning, and information retention. *Journal of Medical Internet Research*, 2013, **15**(7): 144

11 Biran O, Brody S, Elhadad N. Putting it simply: A context-aware approach to lexical simplification. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland, USA: ACL, 2011. 496−501

12 Yatskar M, Pang B, Danescu-Niculescu-Mizil C, Lee L. For the

sake of simplicity: Unsupervised extraction of lexical simplifications from Wikipedia. In: Proceedings of the 2010 Annual Conference of the North American Chapter of the ACL. Los Angeles, USA: ACL, 2010. 365−368

13 Horn C, Manduca C, Kauchak D. Learning a lexical simplifier using Wikipedia. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore, USA: ACL, 2014. 458−463

14 Glavaš G, Štajner S. Simplifying lexical simplification: Do we need simplified corpora. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. Beijing, China: ACL, 2015. 63−68

15 Paetzold G. Reliable lexical simplification for non-native speakers. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics, Student Research Workshop. Denver, UAS: ACL, 2015. 9−16

16 Paetzold G, Specia L. Lexical simplification with neural ranking. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. Valencia, Spain: ACL, 2017. 34−40

17 Devlin J, Chang M W, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis, USA: ACL, 2019. 4171−4186

18 Gooding S, Kochmar E. Recursive context-aware lexical simplification. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: ACL, 2019. 4853−4863

19 Coster W, Kauchak D. Simple English Wikipedia: A new text simplification task. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland, USA: ACL, 2011. 665−669

20 Xu W, Napoles C, Pavlick E, Chen Q Z, Callison-Burch C. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 2016, **4**: 401−415

21 Nisioi S, Štajner S, Ponzetto S P, Dinu L P. Exploring neural text simplification models. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver, Canada: ACL, 2017. 85−91

22 Dong Y, Li Z C, Rezagholizadeh M, Cheung J C K. EditNTS: An neural programmer-interpreter model for sentence simplification through explicit editing. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: ACL, 2019. 3393−3402

23 Xu W, Callison-Burch C, Napoles C. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 2015, **3**: 283−297

24 Shardlow M. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 2014, **4**(1): 58−70

25 Paetzold G H, Specia L. A survey on lexical simplification. *Journal of Artificial Intelligence Research*, 2017, **60**(1): 549−593

26 Pavlick E, Callison-Burch C. Simple PPDB: A paraphrase database for simplification. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin, Germany: ACL, 2016. 143−148

27 Maddela M, Xu W. A word-complexity lexicon and a neural readability ranking model for lexical simplification. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: ACM, 2018. 3749−3760

28 Bautista S, León C, Hervás R, Gervás P. Empirical identification of text simplification strategies for reading-impaired people. In: Proceedings of the European Conference for the Advancement of Assistive Technology. Maastricht, Netherland, 2011. 567−574

29 Lee J, Yoon W, Kim S, Kim D, Kim S, So C H, et al. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 2020, **36**(4): 1234−1240

30 Conneau A, Lample G. Cross-lingual language model pretraining. In: Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Vancouver, Canada: NIPC, 2019.

31 Mikolov T, Grave E, Bojanowski P, Puhrsch C, Joulin A. Advances in pre-training distributed word representations. In: Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018). Miyazaki, Japan: LREC, 2018.

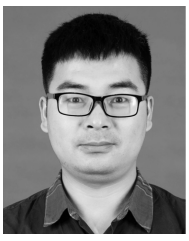
32 Brysbaert M, New B. Moving beyond Kučera and Francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for American English. *Behavior Research Methods*, 2009, **41**(4): 977−990

33 Ganitkevitch J, Van Durme B, Callison-Burch C. PPDB: The paraphrase database. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics. Atlanta, USA: ACL, 2013. 758−764

34 Little D. The Common european framework of reference for languages: Content, purpose, origin, reception and impact. *Language Teaching*, 2006, **39**(3): 167−90

35 Gooding S, Kochmar E. Complex word identification as a sequence labelling task. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: ACL, 2019. 1148−1153

36 Kajiwara T, Matsumoto H, Yamamoto K. Selecting proper lexical paraphrase for children. In: Proceedings of the 25th Conference on Computational Linguistics and Speech Processing (ROCLING 2013). Kaohsiung, China, 2013. 59−73



强继朋 扬州大学信息工程学院副教授. 2016 年获合肥工业大学计算机博士学位. 主要研究方向为数据挖掘和自然语言处理.
E-mail: jpqiang@yzu.edu.cn
(**QIANG Ji-Peng** Associate professor at the School of Information Engineering, Yangzhou University. He received his Ph.D. degree in computer science and technology from Hefei University of Technology in 2016. His research interest covers data mining and natural language processing.)

钱镇宇 扬州大学信息工程学院硕士研究生. 主要研究方向为主题建模和数据挖掘.
E-mail: qzyjnwss@126.com
(**QIAN Zhen-Yu** Master student at the School of Information Engineering, Yangzhou University. His research interest covers topic modeling and data mining.)

李 云 中国扬州大学信息工程学院教授. 主要研究方向为数据挖掘和云计算. 本文通信作者.
E-mail: liyun@yzu.edu.cn
(**LI Yun** Professor at the School of Information Engineering, Yangzhou University. His research interest covers data mining and cloud computing. Corresponding author of this paper.)

袁运浩 扬州大学信息工程学院副教授. 2013 年获南京理工大学模式识别与智能系统博士学位. 主要研究方向为模式识别, 数据挖掘和图像处理.
E-mail: yhyuan@yzu.edu.cn
(**YUAN Yun-Hao** Associate professor at the School of Information Engineering, Yangzhou University. He received his Ph.D. degree in pattern recognition and intelligence system from Nanjing University of Science and Technology in 2013. His research interest covers pattern recognition, data mining, and image processing.)

朱 毅 扬州大学信息工程学院讲师. 2018 年获合肥工业大学软件工程博士学位. 主要研究方向为数据挖掘和知识图谱.
E-mail: zhuyi@yzu.edu.cn
(**ZHU Yi** Lecturer at the School of Information Engineering, Yangzhou University. He received his Ph.D. degree in software engineering from Hefei University of Technology in 2018. His research interest covers data mining and knowledge graph.)