计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
[doi: 10.15888/j.cnki.csa.010100] [CSTR: 32024.14.csa.010100]
©中国科学院软件研究所版权所有.
E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

融合知识增强与 SimCLS 框架的生成式文本摘要模型①

刘佳旭[1], 吴　石[1], 张艳菊[2]

[1](辽宁工程技术大学 软件学院, 葫芦岛 125105)
[2](辽宁工程技术大学 工商管理学院, 葫芦岛 125105)
通信作者: 刘佳旭, E-mail: liujiaxu@buaa.edu.cn

摘　要: 生成式文本摘要旨在通过理解输入的原始文本生成简洁、易读的摘要. 然而, 现有模型输出的摘要中仍存在语义冗余、事实性错误以及暴露偏差的问题. 解决这些问题对于提升模型性能和摘要质量具有重要的研究意义. 因此, 提出一种融合知识增强与 SimCLS 框架的生成式文本摘要模型. 首先, 设计了知识增强编码器获取源文本的结构化知识信息以维护全局上下文的结构信息, 并结合文本编码器充分编码全文语义信息. 其次, 在解码器部分利用 Copy 机制, 更加准确地再现原始文本中的信息. 最后, 使用 SimCLS 两阶段的对比学习训练框架, 对模型生成的摘要进行评分以驱动模型生成高质量摘要. 实验结果表明, 相较于表现较为优异的 SeqCo 模型, 所提模型在 CNN/Daily Mail 数据集上的 ROUGE-1/2/L 及 BERTScore 分别提升了 1.84、0.65、2.04 和 0.21 个百分点, 在 XSum 数据集上分别提升了 1.78、2.16、2.36 和 0.13 个百分点, 验证了模型的有效性.

关键词: 知识增强; 生成式文本摘要; 对比学习; 语义知识; 图注意力网络

引用格式: 刘佳旭,吴石,张艳菊.融合知识增强与 SimCLS 框架的生成式文本摘要模型.计算机系统应用. http://www.c-s-a.org.cn/1003-3254/10100.html

Abstractive Text Summarization Model Combining Knowledge-enhancement and SimCLS Framework

LIU Jia-Xu[1], WU Shi[1], ZHANG Yan-Ju[2]

[1](College of Software, Liaoning Technical University, Huludao 125105, China)
[2](College of Business Management, Liaoning Technical University, Huludao 125105, China)

Abstract: Abstractive text summarization aims to generate concise and readable summaries by understanding the original input text. However, the summaries produced by existing models still face issues such as semantic redundancy, factual errors, and exposure bias. Addressing these problems is crucial for improving model performance and summary quality. Therefore, an abstractive text summarization model that integrates knowledge enhancement with the SimCLS framework is proposed. First, a knowledge-enhanced encoder is designed to obtain structured knowledge information from the source text to preserve the structural information of the global context, and it is combined with a text encoder to fully capture the semantic information of the entire text. Then, the copy mechanism is utilized in the decoder to more accurately reproduce the information from the original text. Finally, the summaries generated by the model are scored using the SimCLS two-stage contrastive learning framework to guide the generation of high-quality summaries. The experimental results show that, when compared with the higher-performing SeqCo model, the proposed model improves ROUGE-1/2/L and BERTScore on the CNN/Daily Mail dataset by 1.84, 0.65, 2.04, and 0.21 percentage points, respectively, and on the

① 基金项目: 辽宁省社会科学规划基金 (L22BJY034)
收稿时间: 2025-09-01; 修改时间: 2025-09-29; 采用时间: 2025-10-14; csa 在线出版时间: 2026-01-15

XSum dataset by 1.78, 2.16, 2.36, and 0.13 percentage points, confirming the model's effectiveness.

Key words: knowledge-enhancement; abstractive text summarization; contrastive learning; semantic knowledge; graph attention network (GAT)

自动文本摘要 (automatic text summarization, ATS), 简称自动文摘, 是自然语言处理领域的一个重要研究方向, 旨在利用算法, 将输入文本信息自动地凝练成一段言简意赅的摘要. 自动文摘从实现方式上可以分为: 抽取式文摘 (extractive summarization) 和生成式文摘 (abstractive summarization)[1,2]. 传统的自动文摘通常采用抽取式文摘方法, 将输入文档中最重要的前 n 个句子进行排序, 然后组合形成摘要[3]. 潜在语义分析 (latent sematic analysis, LSA) 是最早用于抽取式文摘的技术之一. 该技术的核心思想是将文档中每个句子每个单词的频率建模为一个词句矩阵, 然后应用奇异值分解提取文档中的重要语义特征, 从而对重要句子进行排序和提取[4]. 另一种较为突出的抽取式文摘方法是基于图的 TextRank 算法[5], 因为它可以执行快速且易于扩展的摘要, 从而解决了先前方法的一些局限性. 该方法首先将文档表示为句子的加权图, 其中句子表示为节点, 句子之间的关系表示为边. 然后应用 PageRank 中心性算法根据节点权重对其进行排序, 最后选择靠前的句子作为输入文档的摘要. 抽取式文摘方法的优势在于摘要全部来源于原文, 生成的摘要较为连贯, 并且具备良好的可读性. 但是也存在抽取句子重复, 摘要不能完整概括全文等缺陷. 相比之下, 生成式文摘更具有研究意义, 因为它不局限于简单地从原文中进行选择, 而更接近人类撰写摘要的过程, 在理解全文语义的基础上, 生成内容丰富且人类可读的摘要.

随着神经语言模型的使用, 生成式文摘方法逐渐受到学者们的更多关注. 该方法通常采用 Seq2Seq 模型进行生成式文摘任务. 具体来说, 该方法首先使用编码器将输入序列转换为固定长度的向量表示, 然后使用解码器从向量表示生成输出序列[6]. 在此基础上结合注意力机制[7]与指针网络[8]等技术, 提高模型生成摘要的质量.

这些技术的应用提升了模型的表现, 但是仍存在许多问题: 由于模型结构与学习目标的局限性, 模型生成摘要中包含的事实与源文本中描述的事实不一致, 偏离原文表达含义造成事实性错误[9]; 受限于词汇表大小及模型对于上下文的理解能力, 摘要的内容存在语句冗余与词典外词汇 (out of vocabulary, OOV) 问题[8,10]; 由于模型在训练与测试时输入数据的差异, 导致误差的积累, 依据先前错误的步骤生成输出, 损害模型性能造成暴露偏差问题[11].

针对以上问题, 本文提出了一个融合知识增强与 SimCLS 框架的生成式文本摘要模型 (knowledge-enhancement and SimCLS framework fusion model for abstractive text summarization, KESCSum). 如图 1 所示是一个知识图谱构建示例, 图中红色加粗字体表示文本中代表主语的实体, 蓝色、粉色与绿色字体表示与主语相关联的其他实体, 黑色下划线部分表示事件.

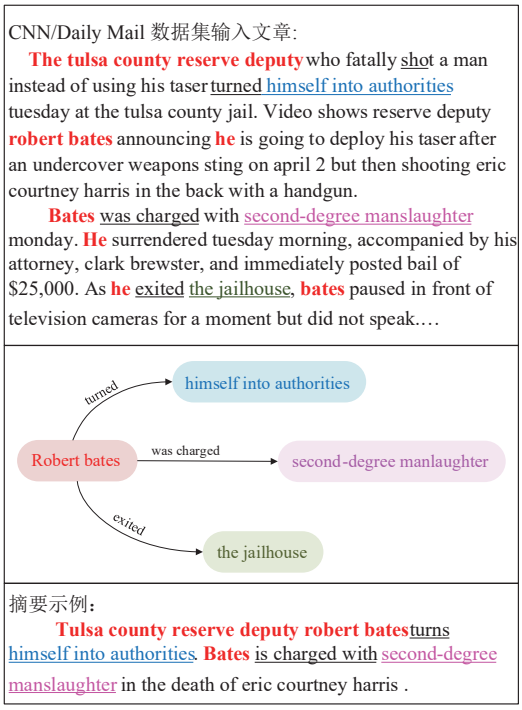

图 1 知识图谱构建示例

由于文本中具有关联的不同实体可能会横跨多个句子, 使得模型难以捕获实体间的交互信息. 因此, 该模型通过提取文本中的结构化知识以构建线性化知识图谱, 增强了模型对实体信息的表达能力. 此外, 相较于连接外部知识, 使用内部的知识图谱能够减少外部

噪声对模型的干扰, 从而进一步增强模型对上下文语义感知能力. 本文在解码器部分采用 Copy 机制以解决生成摘要中存在的语句冗余及 OOV 问题, 确保准确再现原文信息. 最后, 利用 SimCLS 两阶段的对比学习训练框架弥补训练目标和评价指标之间的差距, 缓解模型存在的暴露偏差问题, 从而生成信息丰富且流畅的摘要. 在 CNN/Daily Mail、XSum 公开数据集上的实验结果表明, 本文提出的模型在 ROUGE-1、ROUGE-2、ROUGE-L、BERTScore 指标上优于先前的生成式文摘模型.

1 相关工作

大数据时代的到来, 使得文本信息过载过量的问题日益严重, 人们对于高效、快速阅读的需求正日益上升. 因此, 自动文本摘要任务引起了大量学者的关注. 如今, 伴随着深度学习的兴起, 生成式文摘进入了一个新的发展阶段. Rush 等人[7]受到神经机器翻译 (neural machine translation, NMT) 任务的启发, 首次将注意力机制应用在生成式文摘任务中, 与过去传统的方法相比, 生成摘要的质量得到了显著提升. 此后, Chopra 等人[12]在前者的基础上进行了改进, 引入一种带有条件的循环神经网络 (recurrent neural network, RNN), 该条件由基于注意力的卷积编码器提供. 在 Gigaword 数据集上的效果优于当时最先进的模型. Gu 等人[10]提出一种新型的 CopyNet 模型, 将 Copy 机制引入解码过程中, 有选择的复制输入序列中的子序列并将其融入输出序列中, 缓解 OOV 词的问题. Nallapati 等人[13]将编码器改为 RNN 编码器, 并且通过加入文本特征、生成器指针、分层注意力机制的方式解决了生成式摘要中容易发生的问题. Gehrmann 等人[14]采用一种自下而上 (bottom-up) 的方法, 利用内容选择器过度先确定源文档中需要成为摘要一部分的短语, 利用该选择器作为 bottom-up attention 步骤, 将模型约束为可能的短语, 得到更加简洁流畅的摘要. Chen 等人[15]提出使用基于策略的强化学习算法作为生成模型中的句子抽取模块, 选取重要的句子, 联合训练抽取模块和生成模块进行摘要生成任务. 吴仁守等人[16]引入一种全局自匹配机制, 根据单词语义和文本整体语义动态收集相关词信息, 通过进一步编码获取全局信息的隐层表示, 同时利用全局门控单元筛选出原文本的核心内容.

大型预训练语言模型 (pre-trained language model, PLM) 的出现, 为生成式文摘带来了新的突破. Brown 等人[17]提出 GPT 模型, 基于 Transformer[18]的解码器架构, 能够捕捉更长的记忆信息; 同时通过两阶段的模型, 即“预训练-微调”的方式, 降低了针对不同任务的繁琐性. Devlin 等人[19]提出的 BERT 模型, 采用双向编码器表示句子, 通过预训练阶段的两个任务, 大大提升了文本理解能力. Raffel 等人[20]提出了 T5 (text-to-text transfer Transformer) 模型, 将所有文本任务统一为文本到文本的格式, 大大提升了模型的泛化能力, 同时在自动文摘任务中也表现出了优异的性能. Liu 等人[21]提出了一种 BERT 的变体模型——BERTSUM, 可以同时用于提取式摘要和生成式摘要, 3 个数据集上的实验证明其在自动文摘任务上的先进性. Lewis 等人[22]提出了 BART 模型, 该模型是建立在标准的 Transformer 结构之上, 吸收了 BERT 与 GPT 的特点, 使得其在摘要生成任务取得进步.

现有的基于知识的摘要主要通过利用知识图谱[23]提取所需的结构化知识并应用于摘要生成. Fernandes 等人[24]利用图神经网络增强了基于序列的编码器, 较为全面地考虑了实体类型, 但是在实体交互方面仍有所欠缺. Fan 等人[25]利用 OpenIE 工具输出线性化知识图谱以进行多文档摘要, 展现了知识信息的实用性. Koncel-Kedziorski 等人[26]将知识图谱作为输入表示, 提出了一种新的图编码器, 实现了从图到文本的端到端生成系统, 丰富了文本的语义知识表达的同时, 也证明了知识图谱对于文本生成的有效性.

事实一致性问题[9]是指自动文摘模型生成的摘要所包含的事实, 与给定的输入文本所描述的事实不一致, 从而导致摘要的可读性差, 甚至偏离原文表达含义从而造成事实性错误. 为此, 学者们探索了多种方法以改善这一问题. Li 等人[27]提出对摘要和词义进行联合训练, 同时采用词义感知的编码器来增强事实一致性. Cao 等人[28]利用开放信息提取与依赖解析技术提取文章中的实际事实内容, 并提出 Dual-Attention Seq2Seq 训练框架进一步提升摘要的事实一致性.

2 融合知识增强与 SimCLS 框架的生成式文本摘要模型

生成式摘要旨在将自动文本摘要任务概念化为序列到序列的问题[7], 具体来说, 给模型输入文档D, 将其视为含有n个单词的标记序列$D=\{w_1,w_2,\cdots,w_n\}$, 是为了得到由单词y组成包含m个句子的摘要$Y=\{y_1,y_2,$

$\cdots, y_m\}$, 其中输入文档的序列长度远大于生成摘要的序列长度.

本文提出了一个融合知识增强与 SimCLS 框架的生成式文本摘要模型, 模型整体架构如图 2 所示.

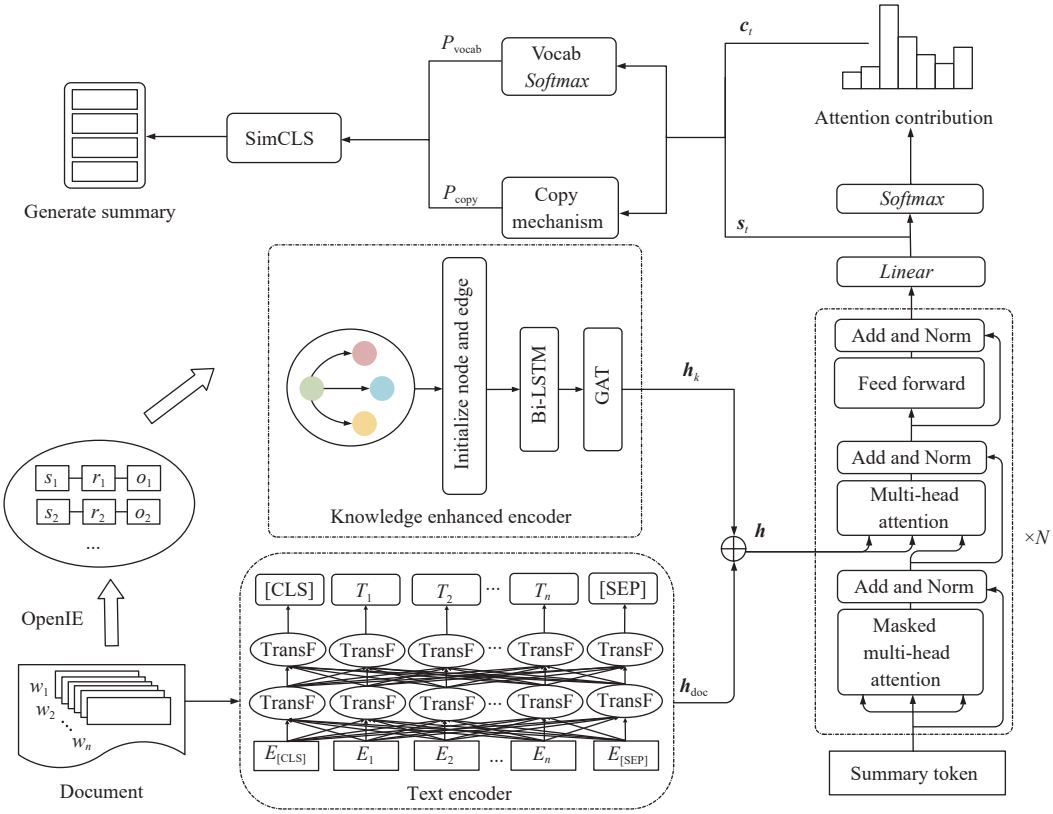

图 2 模型整体结构图

该模型集成了两个编码器综合考虑文章的原文及生成摘要所需的外部知识结构, 使模型能够充分理解文章内容, 进一步提升摘要质量. 首先, 该模型的文本编码器部分采用 BERT 模型获取每个词汇的上下文表示. 然后, 利用开放信息提取模型 OpenIE[29]从文本中提取三元组, 将主语和宾语视为有向边的连接节点, 将谓语视为属性, 初始化节点与边的表示, 以此构建知识图谱. 将生成的知识图谱序列输入到知识增强编码器获取知识图谱表示$\boldsymbol{h}_k$. 利用先前获取到的文本与知识的向量表示进行拼接, 然后输入到解码器中进行解码, 得到上下文向量$\boldsymbol{c}_t$和解码器的输出$\boldsymbol{s}_t$. 最后, 利用 Copy 机制并结合 SimCLS 两阶段的对比学习训练框架生成高质量摘要.

2.1 文本编码器

双编码器之一的文本编码器用于表达文档的语义并获取文档中每个句子的表示. 该编码器的结构与 BERT 预训练模型类似, 由多个 Transformer 的 Encoder 层堆叠而成, 从而更加高效地捕捉文章语句之间的双向关系及语义信息. 每一层都由自注意力 (self-attention) 层、前向反馈 (feed forward) 层与输出 (output) 层组成, 同时利用层归一化和 Dropout 正则化方法提升模型的泛化能力与稳定性. 在文本输入模型前, 需要对每句话的开头和结尾添加[CLS]和[SEP]标记以间隔句子, 最后将其转换为字向量 (token embedding)、文本向量 (segment embedding) 与位置向量 (position embedding) 并进行累加得到y_i输入模型中, 文本编码器最后一层的输出作为整个文本的向量表示$\boldsymbol{h}_{\text{doc}}$, 计算公式如下:

$$\boldsymbol{h}_{\text{doc}} = BERT(y_i), \quad i \in [1, n] \tag{1}$$

2.2 知识增强编码器

为了缓解模型生成摘要中存在事实一致性的问题, 本文构建了一个知识增强编码器, 通过知识图谱能够将文本蕴含的知识引入到摘要生成过程中. 对于知识图谱的构建, 首先利用开源工具 Stanford CoreNLP[30]获取文本内容中的实体并运用共指解析进一步区分不同

实体间的关系. 然后, 利用开放信息提取模型 OpenIE 提取文本中的知识三元组以构建文本的知识图谱. 知识图谱本质上是为了描述各种实体或概念及其关系的一张语义网络图, 节点用来表示实体或概念, 而边则由属性或关系构成, 通常用三元组 SPO 来符号性描述实体之间的关系, 即<Subject, Predicate, Object>. 具体来说, 提取到的知识三元组有 3 个元素, 分别是e_1、r和e_2. e_1和e_2为实体节点, r看作是实体之间的关系或属性, 以此构建文本的知识图谱$G=(V,E)$, 其中V表示节点的集合, E表示边的集合. 对于每个三元组, 我们可以用节点与边的关系来表示, 即$e_1 \xrightarrow{r} e_2$.

本文提出模型通过引入图注意力网络 (graph attention network, GAT)[31]来更新节点的表示. 具体来说, 图注意力网络利用多头注意力机制聚合来自其邻居节点的信息及注意力系数, 迭代更新节点的表示. 由于知识图谱中的每个节点通常指代不同的实体, 因此, 通过平均嵌入来初始化节点表示$\boldsymbol{v}_{i'}$, 同一实体被提及的次数将作为额外编码添加至节点中. 考虑到上下文节点特征, 通过在图注意力网络层中增加残差连接改进了注意力系数, 利用 *Softmax* 函数计算得到知识图谱中实体节点$\boldsymbol{v}_{i'}$与其相邻的节点$\boldsymbol{v}_{j'}$之间的注意力系数$\alpha^n_{i',j'}$, 将该权重系数与上下文节点特征进行拼接, 从而获取最终的节点表示$\hat{\boldsymbol{v}}_{i'}$. 具体计算方式如下:

$$\hat{\boldsymbol{v}}_{i'} = \boldsymbol{v}_{i'} + \left(\Big\|_{n=1}^{N} \sum_{\boldsymbol{v}_{j'} \in \mathcal{N}(\boldsymbol{v}_{i'})} \alpha^n_{i',j'} W_{0,n} \boldsymbol{v}_{j'} \right) \omega_O \tag{2}$$

$$\alpha^n_{i',j'} = Softmax((W_{1,n}\boldsymbol{v}_{i'})^{\top}(W_{2,n}\boldsymbol{v}_{j'})) \tag{3}$$

其中, $\|$表示注意力头的拼接; N表示图注意力头的数量, 每个头可以产生与$v_{i'}$相同维度的向量; W_*为模型训练得到的参数; $\mathcal{N}(\boldsymbol{v}_{i'})$表示图中$\boldsymbol{v}_{i'}$的相邻节点; ω_O为输出投影矩阵, 用于将拼接后的多头特征向量进行线性映射与融合.

最后, 将获取到的节点表示应用平均池化, 以获取最终知识表示$\boldsymbol{h}_k$. 将提取到的知识信息与文本的语义信息进行拼接, 得到最终的编码器特征表示$\boldsymbol{h}$, 计算公式如下:

$$\boldsymbol{h}_k = \frac{1}{\mathcal{V}} \sum_{i' \in \mathcal{V}} \hat{\boldsymbol{v}}_{i'} \tag{4}$$

$$\boldsymbol{h} = \boldsymbol{h}_{\text{doc}} \oplus \boldsymbol{h}_k \tag{5}$$

针对模型构建知识图谱的过程中可能引入的噪声知识, 本文提出模型依据图结构的特性以及图注意力机制实现对噪声的隐性抑制. 具体来说, 模型使用 GAT 更新节点表示时, 通过注意力机制计算不同邻居节点的重要性权重, 语义关联度较高的节点将能够获得更高的权重, 而语义无关或存在噪声的节点则权重较低, 从而削弱噪声知识对模型的影响. 因此模型能够通过注意力的加权聚合实现“噪声过滤”, 从而降低噪声知识对模型性能的影响.

与噪声知识不同, 实体歧义问题是由于一词多义或指代不明确导致同一词汇或实体在不同语境中可能指向不同概念. 针对该问题, 本文提出模型通过语境区分与语义聚合实现消歧. 具体来说, 模型通过图注意力网络将实体节点表示与上下文节点特征相结合, 保留了实体节点在语义环境下的信息, 使模型能够区分不同语境下的同名实体. 此外, 利用多头注意力机制能够进一步整合全局上下文语义, 在整体上实现语义区分, 确保模型输出节点表示的准确性.

2.3 融入知识的摘要生成

在完成了对知识的编码后, 需要将其融入摘要生成的过程中. 当解码时间步为t时, 摘要部分嵌入矩阵通过带有掩码的多头注意力机制并进行归一化操作得到输出$\boldsymbol{p}_t$. 将先前获取到的编码器特征表示$\boldsymbol{h}$与$\boldsymbol{p}_t$输入到多头注意力中, 对于第i个头注意力, 转换方式如下:

$$Att_i(\boldsymbol{h}, \boldsymbol{p}_t) = Softmax\left(\frac{[W_{Q_i}\boldsymbol{p}_t]^{\top}[W_{K_i}\boldsymbol{h}]}{\sqrt{d}} \right)(W_{V_i}\boldsymbol{h})^{\top} \tag{6}$$

在得到注意力的输出后, 模型通过归一化层、前馈网络层与线性回归得到解码器的输出$\boldsymbol{s}_t$, 计算公式如下:

$$\boldsymbol{s}_t = Linear(Ln(MLP(Ln(Att(\boldsymbol{h}, \boldsymbol{p}_t))))) \tag{7}$$

其中, $Ln(\cdot)$表示归一化函数; $Linear(\cdot)$表示线性回归; $MLP(\cdot)$表示标准的前馈神经网络.

随后, 本文模型采用基于 Transformer 的 Decoder 层结构同时结合 Copy 机制以解决生成摘要中存在的 OOV 问题与重复问题. 具体来说, 模型针对知识图谱与文本分别进行解码. 根据知识增强编码器输出的节点表示$\hat{\boldsymbol{v}}_{i'}$与解码器的隐状态$\boldsymbol{s}_t$计算二者的相关程度, 以获取知识图谱部分的权重分布$a^v_{k,t}$及上下文向量$\boldsymbol{c}^v_t$:

$$a_{k,t}^{v} = Softmax(u_0^{\top}\tanh(W_3\boldsymbol{s}_t + W_4\hat{\boldsymbol{v}}_{i'})) \tag{8}$$

$$\boldsymbol{c}_t^{v} = \sum_k a_{k,t}^{v}\hat{\boldsymbol{v}}_{i'} \tag{9}$$

类似地, 针对文本部分的权重分布$a_{\text{doc},t}$以及上下文向量$\boldsymbol{c}_t$时需考虑先前的输入文本向量表示$\boldsymbol{h}_{\text{doc}}$, 计算方式如下:

$$a_{\text{doc},t} = Softmax(u_1^{\top}\tanh(W_5\boldsymbol{s}_t + W_6\boldsymbol{h}_{\text{doc}} + W_7\boldsymbol{c}_t^{v})) \tag{10}$$

$$\boldsymbol{c}_t = \sum_{\text{doc}} a_{\text{doc},t}\boldsymbol{h}_{\text{doc}} \tag{11}$$

其中, tanh为激活函数, u_*为模型训练得到的参数.

此时, 先前两个上下文向量表示可以被视为从知识层面以及文本层面上的重点内容, 用以提示解码器需要着重关注的内容. 将两个上下文向量表示与解码器的隐藏状态连接起来, 得到生成摘要所需的词汇分布P_{vocab}:

$$P_{\text{vocab}} = Softmax(\boldsymbol{M}_{\text{out}}[\boldsymbol{s}_t|\boldsymbol{c}_t|\boldsymbol{c}_t^{v}]) \tag{12}$$

其中, $\boldsymbol{M}_{\text{out}}$表示输出嵌入矩阵, 为了让模型学习到文本中的知识信息, 在输入嵌入矩阵与输出嵌入矩阵之间使用权重共享.

模型还进一步引入了类似 See 等人[8]提出的 Copy 机制以应对生成摘要中存在的语句冗余问题, 通过综合考虑文本中的知识及上下文信息, 使模型倾向于生成信息量更高的词汇, 从而降低生成摘要的冗余程度. 最终得到生成摘要的复制概率为:

$$P_{\text{copy}} = \sigma(\boldsymbol{M}_{\text{copy}}[\boldsymbol{s}_t|\boldsymbol{c}_t|\boldsymbol{c}_t^{v}|\boldsymbol{y}_{t-1}]) \tag{13}$$

其中, $\boldsymbol{y}_{t-1}$表示在步骤$t-1$预测的标记嵌入.

2.4 SimCLS 对比学习训练框架

Seq2Seq 模型在训练阶段通常是在最大似然估计 (maximum likelihood estimation, MLE) 框架下进行训练. 然而, 模型仍存在暴露偏差[11]的问题. 暴露偏差是指模型在训练时采用 Teacher-Forcing 的方式, 解码端上一时刻输入的单词是来自训练集的标准答案, 而在测试时的输入是模型生成的单词, 造成误差的积累, 从而导致模型在训练阶段精度较高, 测试阶段的精度显著下降. 因此, 本文提出的模型使用 SimCLS[32]两阶段的对比学习训练框架, 通过"生成-评估"两阶段框架来生成摘要, 缓解了模型中存在的上述问题.

具体来说, 在"生成"阶段, 利用已有模型生成一批候选摘要; 然后在"评估"阶段, 通过评估模型为候选摘要评分, 最终输出得分最高的候选摘要. 图 3 展示了该方法的训练过程.

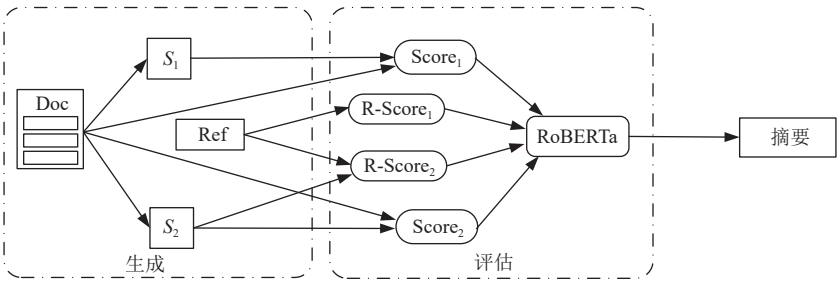

图 3 SimCLS 两阶段对比学习训练框架

该方法的核心在于"评估", 对于更好的候选摘要应当获得相较于源文档更高的分数, 因此定义评估函数$f(\cdot)$并依据源文档D与候选摘要S_i分配不同的相似度分数r_i, 实现对候选摘要的评估. 最终输出得分最高的候选摘要S:

$$S = \arg\max_{S_i} f(S_i, D) \tag{14}$$

对于评估函数将其实例化到 RoBERTa 模型[33], 分别对候选摘要与源文档进行编码, 将编码后的余弦相似度作为源文档与候选摘要的相似度分数, 以此进一步计算损失函数.

在损失函数方面, 与普通对比学习构建的正负样本方式不同, 通过定义一个排序损失函数L_{ctr}实现了对比, 计算公式如下:

$$L_{\text{ctr}} = \sum_i \max(0, f(D,\tilde{S}_i) - f(D,\hat{S})) + \sum_i\sum_{j>i}\max(0, f(D,\tilde{S}_j) - f(D,\tilde{S}_i) + \lambda_{ij}) \tag{15}$$

$$\lambda_{ij} = (j-i)\times\lambda \tag{16}$$

其中, $\hat{S}$ 表示参考摘要; $\{\tilde{S}_1,\cdots,\tilde{S}_n\}$是依据$M(\tilde{S}_i,\hat{S})$进行降序排列的候选摘要, $\tilde{S}_i \in \{\tilde{S}_1,\cdots,\tilde{S}_n\}$, 排名越靠后代表

摘要的指标越高, 候选摘要的质量越好; $M(\cdot)$表示任意的评估指标, 本文使用的是 ROUGE (recall-oriented understudy for gisting evaluation)[34]; λ_{ij}为调整惩罚边距, 由超参数λ计算.

3 实验与结果分析

3.1 数据集描述

CNN/Daily Mail 数据集是由 Hermann 等人[35]从美国有线新闻网 (CNN) 和每日邮报 (Daily Mail) 网中收集的大约一百万条新闻数据, 包含新闻文章与简要概述文章的相关要点. 该数据集中包含 287 227 个数据对用于训练, 13 368 个数据对用于验证和 11 490 个数据对用于测试. 由于其语料量大、文本长, 因此被广泛应用与自动文本摘要任务中. 利用 Stanford CoreNLP 工具包分割句子, 按照 See 等人[8]的方法对数据集进行预处理.

XSum 数据集是由英国广播公司在 2010–2017 年的 22.7 万篇文章组成, 涵盖各种主题, 并附有专业撰写的单句摘要, 回答了“这篇文章是关于什么的?”这一问题. 利用 Narayan 等人[36]的拆分方式, 将数据集划分为 204 045 个数据对用于训练, 11 332 个数据对用于验证和 11 334 个数据对用于测试. 数据集详细信息如表 1 所示.

表 1 数据集信息

数据集名称	训练集	验证集	测试集
CNN/Daily Mail	287 227	13 368	13 368
XSum	204 045	11 332	11 334

3.2 评价指标

实验采用的两类评价指标以检验模型的效果. 在衡量摘要的文本质量方面采用 ROUGE 和 BERTScore[37]指标.

ROUGE 通过计算模型生成摘要与参考摘要之间的重叠的基本单元 (n 元语法、词序列与词对) 的数目来评价生成的摘要质量. 该方法目前已成为摘要评价技术的通用标准之一. BERTScore 是一种用于文本生成的常见评估指标, 通过在 token 级别对齐生成摘要与参考摘要以计算相似度得分, 相比较于传统的评估方法, BERTScore 能更好地捕捉语义信息.

具体来看, 本文采用 ROUGE 标准中的 ROUGE-1、ROUGE-2 及 ROUGE-L 来评测实验的结果. ROUGE-1 和 ROUGE-2 用于衡量文档中一元词和二元词的召回率, 而 ROUGE-L 则是用于衡量摘要和文档之间的最长公共子序列. 计算公式如下:

$$\text{ROUGE-}N = \frac{\sum\limits_{S\in\{Ref\}}\sum\limits_{gram_n\in S} Count_{\text{match}}(n\text{-}gram)}{\sum\limits_{S\in\{Ref\}}\sum\limits_{gram_n\in S} Count(n\text{-}gram)} \tag{17}$$

其中, $Count_{\text{match}}(n\text{-}gram)$表示同时出现在模型生成摘要与参考摘要中的$n\text{-}gram$个数, $Count(n\text{-}gram)$表示参考摘要中出现的$n\text{-}gram$个数, $n\text{-}gram$ 表示n元词, $\{Ref\}$表示参考摘要, $N\in\{1,2,L\}$.

BERTScore 利用 BERT 模型对候选句子与参考句子进行编码, 获取每个词的上下文嵌入, 然后计算候选句子与参考句子中每个词的余弦相似度, 利用贪心匹配算法为候选句子选取与参考句子中最相似的词, 最后根据匹配结果计算召回率、准确率与 $F1$ 分数. 计算公式如下:

$$\begin{cases} R_{\text{BERT}} = \dfrac{1}{|R|}\sum\limits_{x_i\in R}\max\limits_{\hat{x}_j\in l}\boldsymbol{x}_{\text{ref}}^{\top}\boldsymbol{x}_{\text{can}} \\ P_{\text{BERT}} = \dfrac{1}{|l|}\sum\limits_{\hat{x}_j\in l}\max\limits_{x_i\in R}\boldsymbol{x}_{\text{ref}}^{\top}\boldsymbol{x}_{\text{can}} \\ F_{\text{BERT}} = 2\dfrac{P_{\text{BERT}}\cdot R_{\text{BERT}}}{P_{\text{BERT}}+R_{\text{BERT}}} \end{cases} \tag{18}$$

其中, R表示参考摘要的词集合$<x_1,\cdots,x_i>$, l表示候选摘要的词集合$<\hat{x}_1,\cdots,\hat{x}_j>$, $\boldsymbol{x}_{\text{ref}}$与$\boldsymbol{x}_{\text{can}}$分别表示参考句子与候选句子的嵌入向量, R_{BERT}表示召回率, P_{BERT}表示精确率, F_{BERT}表示 $F1$ 分数. 本文中 BERTScore 取值为F_{BERT}.

在衡量摘要的事实一致性方面采用 FactCC[38]指标. FactCC 是一种用于评测摘要事实一致性的流行性指标, 其利用微调后的 BERT 模型来检测模型输出摘要与源文本是否保持事实一致性. 具体来说, FactCC 将每个句子看作一个单独的例子进行处理, 利用模型预测该句子是否与源文档的内容相符.

3.3 实验环境与超参数设置

本文的实验环境为 Ubuntu 20.04 操作系统, 使用 PyTorch 学习框架构建实验平台, 在 NVIDIA 3090 GPU 上进行实验.

在超参数设置上, 针对摘要生成阶段, 文本编码器所使用的模型词语向量维度为 768. 知识增强编码器中 GAT 的层数为 2, 同时为避免出现过拟合现象在 GAT 中添加 Dropout 机制, GAT 中的 Dropout 率设置为 0.6, 其他地方均为 0.1. 编码器与解码器的层数相

同, 均为 6 层. 针对摘要评估阶段, 评分模型由 12 层 Transformer 的 Encoder 层组成, 同时使用 Adam 作为优化器, 其中$\beta_1 = 0.9$, $\beta_2 = 0.999$. 学习率使用 warming-up 策略, 步数为 10000.

3.4 实验结果与分析

3.4.1 对比实验分析

为了评估本文提出的模型在生成式摘要任务中的表现, 将本文模型与以下模型进行对比.

Lead-3[13]: 一种抽取式摘要算法, 依据句子在文章中的位置抽取摘要. 通过选取文本的前 3 句作为摘要, 虽然算法较为简单, 但通常可以涵盖文本的主要内容, 效果甚至超越了部分深度学习模型.

POINTGEN 以及 POINTGEN+COV[8]: 此类模型提出了指针-生成器网络与覆盖机制, 通过利用指针从原文中复制词的同时, 也保留了生成器生成新词的能力. 利用覆盖机制追踪摘要中已存在的信息, 有效缓解了 OOV 词与重复问题.

FASum[39]: 该模型提出将知识图谱中所表示的事实关系提取并整合至摘要生成过程中, 旨在校正文本摘要中的事实性错误.

BERTSUM[21]: 该模型提出一种基于 BERT 的新型文档编码器, 通过应用简单分类器以及句子间的 Transformer 层有效提升了文档语义特征的提取效果, 同时提出一种新的微调方案, 提高了生成摘要的质量.

GSum[40]: 该模型提出一种基于编码器-解码器架构的通用可扩展引导摘要框架, 能够有效接收不同类型的外部指导, 从而让模型在关注源文档的同时也能够关注不同的引导信号.

T5[20]: 该模型采用大型语料库与多层次预训练目标对模型进行预训练, 提出一种统一的模型框架, 将不同形式任务都视为“文本到文本”的生成任务, 从而简化了模型训练与测试的过程.

T-BERTSum[41]: 该模型在 BERTSUM 模型基础上进行了改进, 采用神经主题模型编码潜在的主题表示并与嵌入表示进行匹配, 引导模型生成主题. 此外, 该模型还通过叠加 LSTM 网络层与门控网络进一步过滤有效信息.

CLIFF[42]: 该模型将对比学习引入文本摘要, 旨在推动摘要模型扩大事实一致的摘要与错误摘要之间的差距, 提升模型事实一致摘要的能力.

BART[22]: 一种序列到序列的去噪预训练语言模型, 可看作 BERT 与 GPT 的泛化. 微调后, 可以在文本摘要任务中实现较为优异的性能.

PEGASUS[43]: 一种基于 Transformer 的编码器-解码器生成模型, 采用多种方法抽取重要句子并提出新的预训练任务 GSG (gap sentences generation), 在多个数据集上实现良好的效果.

CaPE[44]: 该模型提出一种对比参数集成的方法, 该方法基于有噪声和无噪声两个数据样本上训练模型权重并调整参数, 从而减少幻觉现象的发生.

SeqCo[45]: 该模型提出一种监督性对比学习模型, 通过最小化源文档、标准摘要与生成摘要之间的相似度, 从而提升模型性能.

KESCSum 模型在 CNN/Daily Mail 数据集上的对比实验结果如表 2 所示, 加粗数值为最优结果. 图 4 为结果对比柱状图.

表 2 CNN/Daily Mail 数据集的实验结果 (%)

模型	Quality				Factuality
	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	FactCC
Lead-3	40.42	17.62	36.67	80.29	**63.21**
POINTGEN	36.44	15.66	33.42	79.98	45.73
POINTGEN+COV	39.53	17.28	36.38	80.11	46.21
FASum	40.53	17.84	37.40	87.86	48.75
BERTSUM	41.72	19.39	38.76	80.04	40.24
GSum	43.78	20.66	40.66	85.73	42.65
T-BERTSum	43.58	20.43	39.80	83.42	41.13
CLIFF	43.92	20.95	40.60	88.68	52.09
BART	44.16	21.28	40.90	87.21	50.61
PEGASUS	44.17	21.47	41.11	88.70	51.58
CaPE	44.28	21.23	40.88	88.31	51.36
SeqCo	44.66	21.57	41.38	88.56	51.74
KESCSum	**46.50**	**22.22**	**43.42**	**88.77**	51.92

通过对表 2 中的数据以及图 4 中的柱形图分析可知, 本文提出模型在 ROUGE 和 BERTScore 评估指标得分上均优于对比模型, 与表现较为优异的 SeqCo 模型相比, 本文提出的 KESCSum 模型在 ROUGE 与 BERTScore 得分分别提升了 1.84、0.65、2.04 和 0.21 个百分点, 证明了本文模型的有效性. 进一步分析, 在衡量文本质量方面, Lead-3 方法虽然简单, 但十分有效, 并且由于 CNN/Daily Mail 数据集的摘要类型更倾向于抽取式文摘, 因此在文本质量上的表现较为良好. CLIFF 模型引入对比学习训练框架, 虽然一定程度上提升了模型生成摘要的事实一致性, 但未能充分利用文本中蕴含的知识信息, 因此摘要的文本质量提升相对有限.

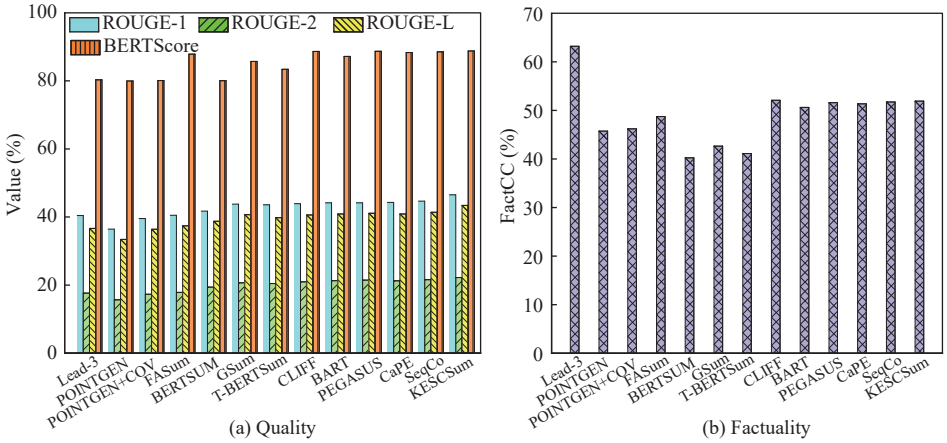

图 4 CNN/Daily Mail 实验结果对比柱状图

此外, 本文提出的模型相较于大型预训练模型 BART 和 PEGASUS 均表现出更好的性能. 这种差异主要归因于以下几个因素: 首先, BART 和 PEGASUS 模型利用传统的注意力机制捕获源文本信息, 难以捕捉文本中存在的实体信息. 而 KESCSum 模型通过构建文本内部的知识图谱, 将文本中的实体信息提炼成一种结构化的知识表示, 提升了模型的信息表达能力. 其次, 这两种模型采用 MLE 方式训练, 难以判断参考摘要中存在的质量差异. KESCSum 模型通过两阶段对比学习训练框架的方式, 增强了模型对语义信息的理解能力, 使模型能够区分这种差异, 从而输出更加流畅、信息量更高的摘要. 本文模型与较优秀的 CaPE、SeqCo 模型相比评估得分更高, 原因在于, CaPE 模型利用两种数据对模型进行训练, 旨在利用参数层面的对比集成方式对模型进行修正, 更加侧重于减少输出摘要的幻觉现象. 而 KESCSum 模型通过知识增强模块获取文本中的结构化知识, 构建与输入文本语义关联程度更高的知识图谱, 捕获全文的知识结构, 从而为模型提供更加精确知识信息. 此外, 采用两阶段的对比学习训练框架弥补了模型训练与测试之间的差距, 帮助模型捕捉到更有效的语义特征, 更为全面地提升了摘要质量. SeqCo 模型旨在将源文本与摘要视为同一语义的不同视图, 优化了模型的语义对齐方式, 但是由于缺乏对文本中实体信息的编码, 导致输出的摘要缺乏广度. KESCSum 模型则将结构化知识融入文本编码, 能够为模型提供更加全面的语义信息. 同时, 对比学习训练框架的引入提升了模型对摘要质量的判别能力, 使其能够生成更全面且流畅的高质量摘要.

而在衡量摘要的事实一致性上, Lead-3 方法倾向于直接从源文本中提取内容, 因此事实一致性得分最高. 除此之外, KESCSum 在事实性评估得分上仅次于侧重提升事实一致性的 FASum 模型, 原因在于 FASum 模型引入了事实校正过程, 对具有事实性错误的摘要进行校正, 提高了摘要的事实性. 虽然 CaPE 模型与 SeqCo 模型在文本质量方面较为优异, 但与本文提出的 KESCSum 模型相比, 在事实一致性方面的表现较差, 说明模型能够通过知识增强模块对文本进行结构化知识编码, 增强模型对关键实体的感知能力, 降低模型输出事实性错误摘要的概率.

KESCSum 模型在 XSum 数据集上的对比实验结果如表 3 所示, 对比实验结果柱状图如图 5 所示.

表 3 XSum 数据集的实验结果 (%)

模型	Quality				Factuality
	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	FactCC
Lead-3	16.30	1.60	11.95	80.39	**30.83**
POINTGEN	29.70	9.21	23.24	81.32	18.26
POINTGEN+COV	28.10	8.02	21.72	82.50	19.05
FASum	29.72	10.03	23.29	88.03	22.85
BERTSUM	38.76	16.33	31.15	82.51	22.35
CLIFF	45.17	23.32	37.61	88.96	23.77
BART	45.14	22.27	37.25	88.26	24.06
PEGASUS	47.21	24.56	39.25	89.19	24.14
SeqCo	45.65	22.41	37.04	91.41	22.48
CaPE	45.35	22.25	37.17	89.73	21.13
KESCSum	**47.43**	**24.57**	**39.40**	**91.54**	25.42

表 3 及图 5 分别展示了 KESCSum 模型与对比模型在 XSum 数据集上的对比实验结果. 在文本质量方面, Lead-3 方法在该数据集上的表现较差, 因为 Lead-3

方法由于偏向于抽取式文摘任务, 而 XSum 数据集的摘要风格与类型更加偏向生成式文摘, 因此该方法在 XSum 数据集上的效果较差. 与 POINTGEN、POINTGEN+COV、FASum、BERTSUM 这些传统的基线模型相比, KESCSum 模型在文本质量指标 ROUGE 和 BERTScore 上均有所提升, 验证了模型的有效性. 此外, 本文模型与表现较为优异的 BART 模型和 SeqCo 模型相比, 在 ROUGE 得分上分别提升了 2.29、2.3、2.15 和 1.78、2.16、2.36 个百分点, 在 BERTScore 得分上分别提升了 3.28 和 0.13 个百分点, 说明本文提出模型能够有效利用知识增强模块理解文中蕴含的关键实体信息, 此外, 对比学习训练框架的使用能够引导模型辨别输出摘要质量的优劣, 增强了模型生成语义连贯且信息完整摘要的能力, 进而提升模型性能. 相较于针对摘要任务优化的 PEGASUS 模型, KESCSum 模型在摘要的文本质量评估上提升幅度较小, 证明知识增强模块在短文本上的表现相对较差. 分析其原因主要体现在短文本中的知识关系数量较少, 导致模型构建的知识图谱稀疏, 文本内部的语义关联性较差, 从而限制了模型对知识信息的利用. 此外, 短文本的语义重叠度较高, 使得知识增强模块难以发挥对全局语义信息的编码能力, 因此知识增强模块在短文本上的表现较差.

而在事实一致性评估上, 除 Lead-3 方法外, KESCSum 模型的事实一致性得分均高于其他对比模型. 验证了本文提出的知识增强模块能够充分利用文本中的结构化知识, 为模型提供准确的知识信息, 从而提升了摘要的事实一致性. 此外, 模型通过两阶段的对比学习训练框架模块能够学到高质量摘要蕴含的特征, 在提升摘要事实一致性的同时, 也保留了较高文本质量.

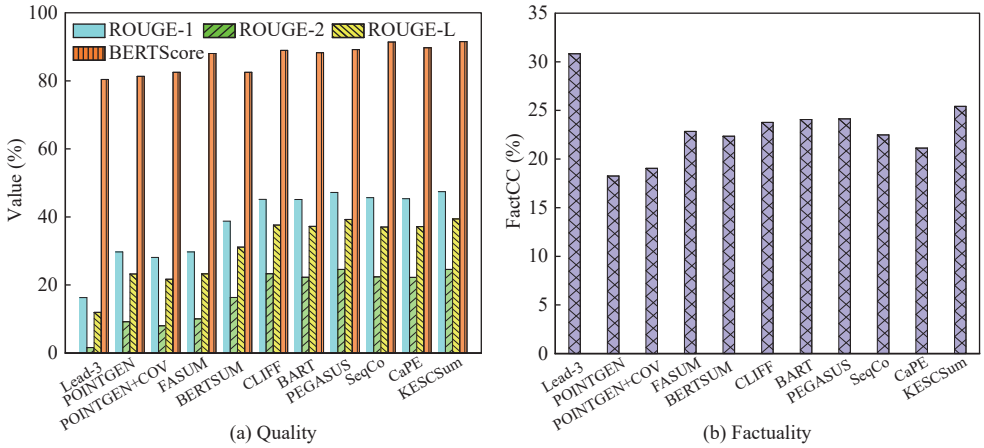

图 5 XSum 实验结果对比柱状图

3.4.2 消融实验分析

为了深入理解不同模块对于本文提出模型的性能贡献, 本节通过逐一掩盖不同模块在 CNN/Daily Mail 数据集上进行了多组消融实验, 实验结果如表 4 所示.

表 4 消融实验结果 (%)

模型	Quality				Factuality
	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	FactCC
KESCSum–KE	45.68	21.74	42.61	87.79	50.48
KESCSum–SimCLS	46.00	22.02	43.00	88.70	51.25
KESCSum	46.50	22.22	43.42	88.77	51.92

表 4 中, KESCSum–KE 代表移除知识增强模块, KESCSum–SimCLS 代表移除了 SimCLS 两阶段对比学习训练模块, 而 KESCSum 则代表本文所提出的模型. 从实验结果来看, 两个模块均对模型性能带来了积极影响, 而在移除了知识增强模块后, 模型的评估指标得分显著下降, 证实了知识增强模块在提升文本质量与事实性方面的重要作用. 同时, 与 SimCLS 两阶段对比学习训练框架模块相比, 知识增强模块对于模型性能的影响更为显著, 表明该模块能够让模型学习更有价值的结构化知识, 帮助模型生成内容丰富且具有更高事实性的摘要. 观察移除 SimCLS 两阶段的对比学习训练模块后的实验结果可知, 衡量文本质量的 ROUGE 与 BERTScore 得分没有明显下降. 因为 SimCLS 两阶段的对比学习训练框架可以缩小训练与测试阶段之间的差距, 提高模型对文本质量的控制能力, 生成具有简

洁性、可读性的高质量摘要.

3.4.3 模型效率与性能分析

为了探究本文模型在处理大规模文本数据集时的效率与性能表现, 本文通过比较不同模型的提取度得分以及新 *n-gram* 在生成摘要中占据的比重两个角度进行实验分析. 提取度得分可以量化摘要取自源文本的程度, 得分越低代表模型生成摘要的质量越高, 从而验证模型能够更加高效地生成简洁清晰的摘要. 模型提取度得分的可视化结果如图 6 所示.

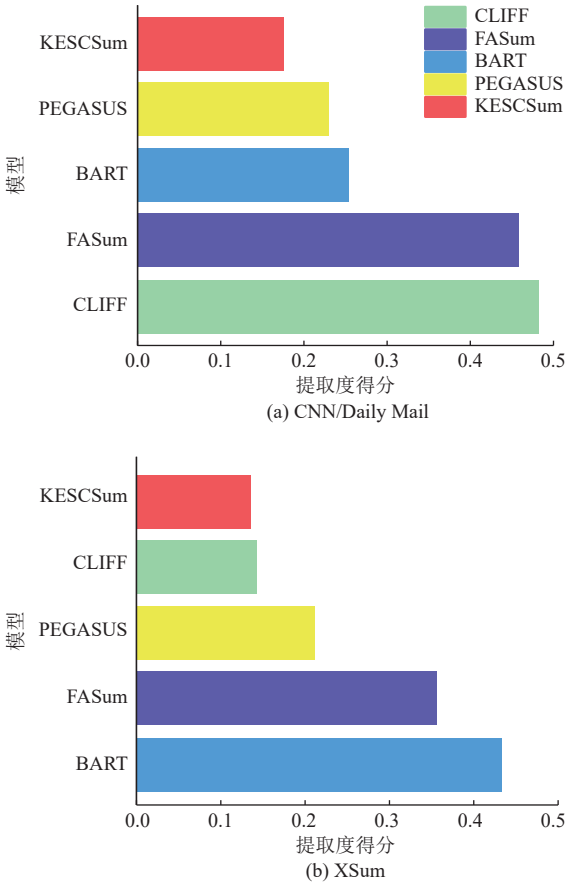

图 6 不同模型在 CNN/Daily Mail 和 XSum 数据集上的提取度得分

实验结果表明, KESCSum 模型在提取度得分上明显低于其他模型, 证明该模型能够高效地利用文本中获取到的知识, 从而提升处理大规模文本数据的效率.

新 *n-gram* 表示出现在摘要中但未出现在源文本中的 *n* 元词, 通过新 *n-gram* 在摘要中的比重可以衡量模型的性能, 模型性能越好则生成新 *n-gram* 在摘要中的比重越高, 其可视化结果如图 7 所示.

从图 7 中可以看出, 随着新颖度的提升, ROUGE 指标呈现下降趋势, 表明模型的性能也在逐渐下降. 相较于性能较好的 BART 模型和 PEGASUS 模型, 本文提出的模型下降幅度较小, 进一步证明了 KESCSum 模型性能方面的优越性.

3.4.4 泛化性分析

为了进一步评估模型是否具有较优的泛化能力, 本文采用中文摘要数据集 NLPCC2017 上进行泛化性实验. 该数据集包含 50 000 条数据, 每条数据均由摘要-正文对构成, 其中正文平均长度为 990 字, 摘要平均长度为 44 字. 数据集语料来自头条上的新闻文本, 内容主要以中文新闻为主, 涉及领域包含科技、政治、经济、娱乐等. 实验结果如表 5 所示.

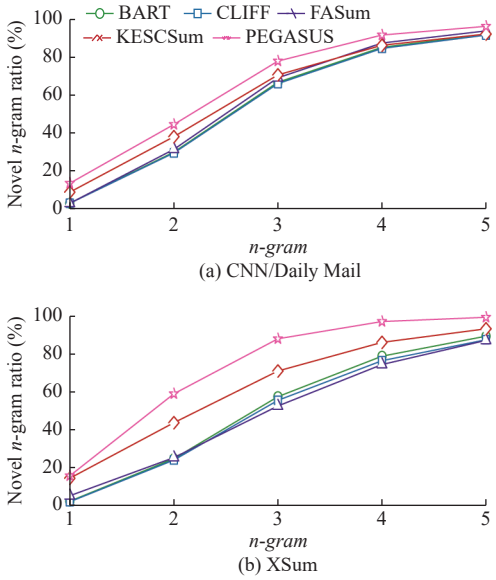

图 7 不同模型在 CNN/Daily Mail 与 XSum 数据集上的新 *n-gram* 比重

表 5 泛化性实验结果分析 (%)

模型	ROUGE-1	ROUGE-2	ROUGE-L
PEGASUS	**59.91**	43.54	55.88
BART	54.48	39.16	47.93
T5	59.71	**44.16**	**56.75**
KESCSum	59.36	43.55	56.10

从表 5 可以看出, KESCSum 模型在 NLPCC2017 数据集上表现略逊于其他模型, 但仍然取得了不错的结果. 在 ROUGE-2 和 ROUGE-L 指标上, 其得分仅次于 T5 模型, 分析其原因主要有两点. 首先, 中文语料的语言结构与英文有着显著差异, 存在缺乏显式词界、句法依存特征复杂等问题, 从而导致知识增强模块在构建知识图谱时易出现实体边界模糊与关系识别偏差, 从而削弱该模块的有效性. 其次, KESCSum 模型的知

识增强机制主要针对英文语料设计和优化, 在中文场景下图结构的稀疏性更明显, 使得模型的适应性较弱, 从而导致效果受限.

此外, 数据集涉及领域的多样性也验证了 KESCSum 模型拥有较好的泛化性, 具有较优的应用价值.

3.4.5 案例对比分析

为了直观分析实验的效果, 表 6 通过一个在 CNN/Daily Mail 数据集中的实例展示了本文提出模型的摘要生成能力. 相比较于使用对比学习的 CLIFF 模型, KESCSum 模型可以利用构建的知识信息从文本中捕获深层次的语义信息, 使得生成的摘要语义更为丰富. 此外, 相比较于 PEGASUS 模型, 本文提出模型通过综合考虑文本摘要中存在的 OOV 问题及语句冗余问题, 同时结合 SimCLS 两阶段对比学习训练框架, 能够更为精确地阐述源文本的含义, 生成内容简洁且信息量丰富的高质量摘要.

表 6 CNN/Daily Mail 数据集摘要示例

类别	内容
源文本	At the grand old age of 75, Jack Nicklaus is still capable of hitting aces. The Golden Bear added another magic moment to his storied career at Augusta National in the Par-3 Contest. Stepping up to the tee on the 130-yard fourth, the greatest golfer of all time saw his shot sail beyond the flag before spinning back into the hole. Jack Nicklaus gave the crowd something to cheer with a hole in one on the fourth during the Par-3 Contest. Nicklaus holds up his ball to an adoring crowd as Gary Player (left) and Ben Crenshaw salute the great. Crenshaw and Nicklaus fist pump following his ace on the 130-yard hole at Augusta National. Nicklaus was playing alongside Gary Player and Ben Crenshaw as the first roar of Masters week filled the skies in a sunswept Georgia. I had an interview this morning, and I said “well, all I’ve got to do is go out and win the Par-3 and make a hole and one”, and I make a hole in one, Nicklaus said afterwards. The ace was Nicklaus’ first on the hallowed turf of Augusta, but the veteran saw his achievement eclipsed by that of Camilo Villegas on a day which saw five hole-in-ones, levelling the record from 2002. The Colombian followed Nicklaus with an ace on the fourth and also managed one on the eighth, but he couldn’t quite land the title, losing out to world No 74 Kevin Streelman in a play-off. Nicklaus became the youngest person to wear a green jacket in 1963, and collected his sixth in 1986. He is one of five men to complete the career grand slam, an accolade which favourite Rory McIlroy can achieve if he wins his third major in succession.
参考摘要	Nicklaus, a six-time winner at augusta, added another glorious moment to his glittering masters career with an ace on the fourth. The 75-year-old saw his shot sail beyond the flag before spinning back into the cup. The crowd roared in celebration with several players stopping their round to cheer the golden bear. Colombian camilo villegas later eclipsed nicklaus ' achievement, hitting two aces during his round. Villegas was beaten in a play-off by world no 74 kevin streelman.
CLIFF	Jack Nicklaus hit a hole in one on the fourth during the Par-3 Contest. The 75-year-old was playing alongside gary player and ben crenshaw. Camilo Villegas followed Nicklaus with a hole-in-one on the eighth.
PEGASUS	Nicklaus hit a hole-in-one during the Par-3 contest at augusta national. He was playing alongside gary player and ben crenshaw. The golden bear is one of five men to complete the career grand slam. Rory mcilroy can achieve the accolade if he wins his third major in succession.
KESCSum	**Jack Nicklaus** hit a hole in one on the fourth during **the Par-3 Contest.** The golden bear added another magic moment to his storied career **at augusta national. The 75-year-old** was playing alongside gary player and **ben crenshaw. Camilo Villegas** followed Nicklaus with an ace on the same hole.

4 结论与展望

本文提出一种融合知识增强与 SimCLS 框架的生成式文本摘要模型, 该模型通过知识增强编码器提取文本中知识构建具有实体及其关系的知识图谱, 以获取文本中的深层次语义关系, 增强模型对重要信息的提取能力. 在解码器部分应用 Copy 机制, 精确捕捉上下文语义信息, 提升了模型生成摘要内容的准确性. 此外, 结合 SimCLS 两阶段的对比学习训练框架能够进一步优化模型, 使生成的摘要内容更为简洁且信息丰富. 在 CNN/Daily Mail 和 XSum 数据集上的多项实验结果表明, 本文提出模型在各项评价指标上超过了多个对比模型, 验证了模型的有效性.

在未来的研究中, 考虑在现有工作基础上, 进一步加强文本信息、主题信息等以提升模型的生成能力, 同时探索多样化的预训练方式, 以全面地提升生成式文本摘要模型的性能.

参考文献

1 Lin H, Ng V. Abstractive summarization: A survey of the state of the art. Proceedings of the 33rd AAAI Conference on Artificial Intelligence. Honolulu: AAAI Press, 2019. 9815–9822. [doi: 10.1609/aaai.v33i01.33019815]

2 李金鹏, 张闯, 陈小军, 等. 自动文本摘要研究综述. 计算机研究与发展, 2021, 58(1): 1–21. [doi: 10.7544/issn1000-1239202120190785]

3 Moratanch N, Chitrakala S. A survey on extractive text summarization. Proceedings of the 2017 International

Conference on Computer, Communication and Signal Processing. Chennai: IEEE, 2017. 1–6. [doi: 10.1109/ICCCSP.2017.7944061]

4 Gong YH, Liu X. Generic text summarization using relevance measure and latent semantic analysis. Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New Orleans: ACM, 2001. 19–25. [doi: 10.1145/383952.383955]

5 Mihalcea R, Tarau P. TextRank: Bringing order into text. Proceedings of the 42nd Conference on Empirical Methods in Natural Language Processing. Barcelona: Association for Computational Linguistics, 2004. 404–411.

6 Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. Proceedings of the 28th International Conference on Neural Information Processing Systems. Montreal: MIT Press, 2014. 3104–3112.

7 Rush AM, Chopra S, Weston J. A neural attention model for abstractive sentence summarization. Proceedings of the 53rd Conference on Empirical Methods in Natural Language Processing. Lisbon: Association for Computational Linguistics, 2015. 379–389. [doi: 10.18653/v1/D15-1044]

8 See A, Liu PJ, Manning CD. Get to the point: Summarization with pointer-generator networks. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver: Association for Computational Linguistics, 2017. 1073–1083. [doi: 10.18653/v1/P17-1099]

9 Cao ZQ, Wei FR, Li WJ, *et al*. Faithful to the original: Fact aware neural abstractive summarization. Proceedings of the 32nd AAAI Conference on Artificial Intelligence. New Orleans: AAAI Press, 2018. 4784–4791. [doi: 10.1609/aaai.v32i1.11912]

10 Gu JT, Lu ZD, Li H, *et al*. Incorporating copying mechanism in sequence-to-sequence learning. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin: Association for Computational Linguistics, 2016. 1631–1640. [doi: 10.18653/v1/P16-1154]

11 Bengio S, Vinyals O, Jaitly N, *et al*. Scheduled sampling for sequence prediction with recurrent Neural networks. Proceedings of the 29th International Conference on Neural Information Processing Systems. Montreal: MIT Press, 2015. 1171–1179.

12 Chopra S, Auli M, Rush AM. Abstractive sentence summarization with attentive recurrent neural networks. Proceedings of the 54th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego: Association for Computational Linguistics, 2016. 93–98. [doi: 10.18653/v1/N16-1012]

13 Nallapati R, Zhai FF, Zhou BW. SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. Proceedings of the 31st AAAI Conference on Artificial Intelligence. San Francisco: AAAI Press, 2017. 3075–3081. [doi: 10.1609/aaai.v31i1.10958]

14 Gehrmann S, Deng YT, Rush AM. Bottom-up abstractive summarization. Proceedings of the 56th Conference on Empirical Methods in Natural Language Processing. Brussels: Association for Computational Linguistics, 2018. 4098–4109. [doi: 10.18653/v1/D18-1443]

15 Chen YC, Bansal M. Fast abstractive summarization with reinforce-selected sentence rewriting. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne: Association for Computational Linguistics, 2018. 675–686. [doi: 10.18653/v1/P18-1063]

16 吴仁守, 王红玲, 王中卿, 等. 全局自匹配机制的短文本摘要生成方法. 软件学报, 2019, 30(9): 2705–2717. [doi: 10.13328/j.cnki.jos.005850]

17 Brown TB, Mann B, Ryder N, *et al*. Language models are few-shot learners. Proceedings of the 34th International Conference on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 159.

18 Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need. Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 6000–6010.

19 Devlin J, Chang MW, Lee K, *et al*. BERT: Pre-training of deep bidirectional Transformers for language understanding. Proceedings of the 57th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis: Association for Computational Linguistics, 2019. 4171–4186. [doi: 10.18653/v1/N19-1423]

20 Raffel C, Shazeer N, Roberts A, *et al*. Exploring the limits of transfer learning with a unified text-to-text Transformer. Journal of Machine Learning Research, 2020, 21(1): 140.

21 Liu Y, Lapata M. Text summarization with pretrained encoders. Proceedings of the 57th Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. Hong Kong: Association for Computational Linguistics, 2019. 3730–3740. [doi: 10.18653/v1/D19-1387]

22 Lewis M, Liu YH, Goyal N, *et al*. BART: Denoising sequence-to-sequence pre-training for natural language

generation, translation, and comprehension. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2020. 7871–7880. [doi: 10.18653/v1/2020.acl-main.703]

23 王萌, 王昊奋, 李博涵, 等. 新一代知识图谱关键技术综述. 计算机研究与发展, 2022, 59(9): 1947–1965. [doi: 10.7544/issn1000-1239.20210829]

24 Fernandes P, Allamanis M, Brockschmidt M. Structured neural summarization. Proceedings of the 2019 International Conference on Learning Representations. New Orleans: OpenReview.net, 2019.

25 Fan A, Gardent C, Braud C, *et al*. Using local knowledge graph construction to scale Seq2Seq models to multi-document inputs. Proceedings of the 57th Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. Hong Kong: Association for Computational Linguistics, 2019. 4186–4196. [doi: 10.18653/v1/D19-1428]

26 Koncel-Kedziorski R, Bekal D, Luan Y, *et al*. Text generation from knowledge graphs with graph Transformers. Proceedings of the 57th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis: Association for Computational Linguistics, 2019. 2284–2293. [doi: 10.18653/v1/N19-1238]

27 Li HR, Zhu JN, Zhang JJ, *et al*. Ensure the correctness of the summary: Incorporate entailment knowledge into abstractive sentence summarization. Proceedings of the 27th International Conference on Computational Linguistics. Santa Fe: Association for Computational Linguistics, 2018. 1430–1441.

28 Cao M, Dong Y, Wu JP, *et al*. Factual error correction for abstractive summarization models. Proceedings of the 58th Conference on Empirical Methods in Natural Language Processing. Stroudsburg: Association for Computational Linguistics, 2020. 6251–6258. [doi: 10.18653/v1/2020.emnlp-main.506]

29 Saha S, Mausam. Open information extraction from conjunctive sentences. Proceedings of the 27th International Conference on Computational Linguistics. Santa Fe: Association for Computational Linguistics, 2018. 2288–2299.

30 Manning C, Surdeanu M, Bauer J, *et al*. The Stanford CoreNLP natural language processing toolkit. Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. Baltimore: Association for Computational Linguistics, 2014. 55–60. [doi: 10.3115/v1/P14-5010]

31 Veličković P, Cucurull G, Casanova A, *et al*. Graph attention networks. Proceedings of the 6th International Conference on Learning Representations. Vancouver: OpenReview.net, 2018.

32 Liu YX, Liu PF. SimCLS: A simple framework for contrastive learning of abstractive summarization. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. Stroudsburg: Association for Computational Linguistics, 2021. 1065–1072. [doi: 10.18653/v1/2021.acl-short.135]

33 Liu YH, Ott M, Goyal N, *et al*. RoBERTa: A robustly optimized BERT pretraining approach. arXiv:1907.11692, 2019.

34 Lin CY. ROUGE: A package for automatic evaluation of summaries. Proceedings of the 2004 Text Summarization Branches Out. Barcelona: Association for Computational Linguistics, 2004. 74–81.

35 Hermann KM, Kočiský T, Grefenstette E, *et al*. Teaching machines to read and comprehend. Proceedings of the 29th International Conference on Neural Information Processing Systems. Montreal: MIT Press, 2015. 1693–1701.

36 Narayan S, Cohen SB, Lapata M. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels: Association for Computational Linguistics, 2018. 1797–1807. [doi: 10.18653/v1/D18-1206]

37 Zhang TY, Kishore V, Wu F, *et al*. BERTScore: Evaluating text generation with BERT. Proceedings of the 2020 International Conference on Learning Representations. Addis Ababa: OpenReview.net, 2020.

38 Kryscinski W, McCann B, Xiong CM, *et al*. Evaluating the factual consistency of abstractive text summarization. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2020. 9332–9346. [doi: 10.18653/v1/2020.emnlp-main.750]

39 Feng N, dos Santos CN, Zhu HH, *et al*. Improving factual consistency of abstractive summarization via question answering. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. Stroudsburg: Association for Computational

Linguistics, 2021. 6881–6894. [doi: 10.18653/v1/2021.acl-long.536]

40 Dou ZY, Liu PF, Hayashi H, *et al*. GSum: A general framework for guided neural abstractive summarization. Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2021. 4830–4842. [doi: 10.18653/v1/2021.naacl-main.384]

41 Ma TH, Pan Q, Rong H, *et al*. T-BERTSum: Topic-aware text summarization based on BERT. IEEE Transactions on Computational Social Systems, 2022, 9(3): 879–890. [doi: 10.1109/TCSS.2021.3088506]

42 Cao SY, Wang L. CLIFF: Contrastive learning for improving faithfulness and factuality in abstractive summarization. Proceedings of the 59th Conference on Empirical Methods in Natural Language Processing. Punta Cana: Association for Computational Linguistics, 2021. 6633–6649. [doi: 10.18653/v1/2021.emnlp-main.532]

43 Zhang JQ, Zhao Y, Saleh M, *et al*. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. Proceedings of the 37th International Conference on Machine Learning. JMLR.org, 2020. 1051.

44 Choubey PK, Fabbri A, Vig J, *et al*. CaPE: Contrastive parameter ensembling for reducing hallucination in abstractive summarization. Proceedings of the 2023 Findings of the Association for Computational Linguistics. Toronto: Association for Computational Linguistics, 2023. 10755–10773. [doi: 10.18653/v1/2023.findings-acl.685]

45 Xu SS, Zhang XX, Wu Y, *et al*. Sequence level contrastive learning for text summarization. Proceedings of the 36th AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2022. 11556–11565. [doi: 10.1609/aaai.v36i10.21409]

(校对责编: 张重毅)