尹金鑫, 尹军祖. 基于改进预训练模型的裁判文书摘要生成研究[J]. 智能计算机与应用,2025,15(6):50-57. DOI:10.20169/j.issn.2095-2163.24122604

基于改进预训练模型的裁判文书摘要生成研究

尹金鑫, 尹军祖
(中国人民公安大学 信息网络安全学院, 北京 100038)

摘 要: 裁判文书是人民法院公开审判活动、裁判理由、裁判依据和裁判结果的重要载体。然而,文书篇幅较长,影响了快速、有效的阅读体验。为解决这一问题,本文提出了一种基于预训练模型的裁判文书抽取式摘要生成方法。该方法改进了 Oracle 抽取方法,基于 BERT 和束搜索提取关键句子索引,并优化了检索生成模型的评分机制,结合 Transformers 和注意力机制,增强了模型的上下文理解能力和句子选择准确性。实验结果表明,该方法在 ROUGE-1、ROUGE-2 和 ROUGE - L 的 Recall 上分别提升了 16.53%、5.46%和 16.61%,优于现有的一些主流方法。

关键词: 裁判文书摘要; BERT; 束搜索; Transformers; 注意力机制
中图分类号: D926.13 **文献标志码**: A **文章编号**: 2095-2163(2025)06-0050-08

Research on the generation of abstracts for judicial documents based on improved pre-trained models

YIN Jinxin, YIN Junzu
(School of Information and Cyber Security, People's Public Security University of China, Beijing 100038, China)

Abstract: Judicial documents are important carriers for the public trial activities, reasons, basis, and results of the people's court. However, the length of the document is relatively long, which affects the fast and effective reading experience. To address this issue, this paper proposes a pre-trained model based method for extracting and generating judicial document abstracts. This method improves the Oracle extraction method by extracting key sentence indexes based on BERT and bundle search, optimizing the scoring mechanism of the retrieval generation model, and combining Transformers and attention mechanisms to enhance the model's contextual understanding ability and sentence selection accuracy. The experimental results show that this method improves the recall of ROUGE-1, ROUGE-2, and ROUGE - L by 16.53%, 5.46%, and 16.61%, respectively, which is superior to some existing mainstream methods.

Key words: judicial document summarization; BERT; beam search; Transformers; attention mechanism

0 引 言

随着互联网和人工智能的快速发展,裁判文书的摘要生成已成为提升司法效率的重要方向。裁判文书通常篇幅较长,信息量大,快速提取案件关键信息(如纠纷类型、裁判依据、判决结果)具有重要现实意义[1]。然而,现有基于预训练模型的摘要生成方法在处理长文本时效率较低,且难以有效捕捉长距离依赖关系。

以 Transformers[2]为主的预训练模型在短文本摘要中取得了先进的性能。但是由于全自注意力[3]的时空复杂度为 $O(n^2)$,即时间和空间消耗会随着输入序列的长度呈平方级增长[4]。传统的长文本处理方法通常将文本切分为多个片段,每部分大小设置为预训练模型的最大处理长度(如 512 字符),然后整合决策结果。然而,这种方法难以有效捕捉文本块之间的联系和长距离依赖。为此,基于 Transformer 的变种应运而生,优化了长文本建模。Transformer-XL[5]提出了 2 种改进策略:状态复用的块级别循环和相对位置编码。Reformer[6]主要引入了局部敏感哈希注意力和可逆 Transformer 技术。Longformer[7]通过滑动窗口和稀疏注意力模式扩展最大输入长度至 4096,降低计算复杂度。尽管这些变体有效处理长文本,但仍面临记忆限制和计算开

作者简介: 尹金鑫(1999—),男,硕士研究生,主要研究方向:自然语言处理。
通信作者: 尹军祖(1972—),男,副教授,主要研究方向:警务信息技术。Email:112808402@qq.com。

收稿日期: 2024-12-26

销较大的挑战。

本文以 DYLE[8] 模型为基础进行改进，旨在提升裁判文书摘要生成的效率和准确性。首先，改进的抽取器基于 MacBERT 模型，采用束搜索（Beam Search）从长文本中选取最相关的片段，为生成阶段提供精确的信息支持。其次，生成器结合中文 Bart 模型，通过进一步加工抽取的片段，生成简洁、连贯的摘要。该方法有效提升了模型对长文本的上下文理解能力和句子选择准确性，尤其在处理长文本时优于传统方法。

1　文本摘要的相关工作

文本摘要是一种自动化过程，通过使用自然语言处理技术将原始文本文档转换为较短的文本，根据给定的标准突出显示最重要的信息[9]。目前，文本摘要的方法主要分为抽取式摘要和生成式摘要。抽取式和生成式摘要的优缺点具体见表 1。

抽取式摘要通过选择原文中的关键句子或短语生成摘要。最早的重点句子识别方法基于启发式算法。1958 年，Luhn[10] 提出了基于单词频率评估句子重要性的关键句子选择方法，随后频率方法成为主流。1998 年，Carbonell 等学者[11] 提出了 TF-ISF 算法，通过迭代方式结合 TF-ISF 和句子相关性优化摘要生成。2003 年，Wu 等学者[12] 将本体编码为树形结构，通过段落中词汇的频繁出现评估相关性。之后，监督式机器学习方法逐渐成为热点。2007 年，Svore 等学者[13] 使用前馈神经网络和 RankNet 算法识别重要句子。近年来，基于强化学习的方法逐渐被应用于抽取式摘要中。2021 年，Gu 等学者[14] 提出了多步情景马尔可夫决策过程，通过动态提取关键句子增强模型的信息捕捉能力。

生成式摘要通过理解原文内容，生成全新句子来表达相同的意思，而不局限于原文中的句子。随着预训练模型的兴起，生成式摘要技术发展迅速。2012 年，Genest 等学者[15] 采用手工制定的信息抽取规则和生成模式生成摘要。基于 Transformer 的预训练模型逐渐成为主流。2019 年，Liu 等学者[16] 改进 BERT 模型，增加句子编码层和摘要判断层。Zhang 等学者[17] 构建了 PEGASUS 模型，通过屏蔽输入文档中的重要句子进行摘要训练。同年，Lewis 等学者[18] 提出了 BART 模型，通过对含噪输入文本进行去噪重构预训练。Longformer 作为 BART 和 PEGASUS 的扩展，采用线性自注意力机制，高效处理长文档。2020，Zaheer 等学者[19] 创建了 BigBird 模型，提出了 BigBird 模型，进一步优化了 Transformer 在处理长文档时的性能。

表 1　抽取式和生成式摘要的优缺点

Table 1　Advantages and disadvantages of extractive and generative abstracts

方法类型	优点	缺点
抽取式摘要	信息保真度高，生成过程相对简单	语句流畅性差，可读性差，容易遗漏关键信息
生成式摘要	生成摘要更灵活、连贯性强	质量不稳定，尤其在复杂文本中易丢失信息

本文采用抽取-生成两阶段的文本摘要生成方法[20]。这种方法既能够通过抽取关键句子保证信息的完整性，又能通过生成新句子提高摘要的连贯性和流畅性，从而有效提升摘要的质量。

2　抽取和评分机制的改进方法

2.1　改进的 Oracle 抽取

首先，数据格式为 {text，summary} 的 JSON 格式。Oracle 抽取的目的是从文本中选择高分句子序列，并将选中的句子索引映射回原始文本。通过此过程形成的数据对用于优化抽取器模型的训练。因此，如何基于 ROUGE 分数这个指标，尽可能多地抽取高分句子并将其加入到高分句子阵列，对于提升抽取器的语义理解能力至关重要[21]。

目前主流的方法是贪婪搜索（Greedy Search），该方法在每一步都选择局部最优解，但不一定获得全局最优解。为了解决这个问题，本文的 Oracle 摘要主要采用束搜索（Beam Search）进行初步特征提取（见图 1）。束搜索通过在输入批次中进行关键字匹配，寻找与摘要最相关的句子。此外，本文还对句子进行了边界检查和去重，以确保返回的句子索引合法且与原文内容高度相关。

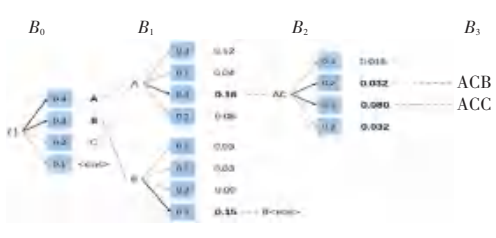

图 1　束搜索示意图

Fig. 1　Schematic diagram of beam search

本文在束搜索(Beam Search)的方法基础上进行改进优化。为了更好地识别一些专有名词,例如“担保物权纠纷”、“人身自由权纠纷”和“生命权”等,本文浏览了文本数据,记录了出现频率较高的专有名词,并在分词时引入使用。改进的 Oracle 抽取的结构图如图 2 所示,具体内容为:

(1)输入文本经过 Jieba 分词处理,首先将长文本分割为句子或片段,以便对文本进行逐句分析。接着,使用 MacBERT[22] 模型进行文本嵌入,获得每个句子的向量表示。

(2)在每个候选句子的生成步骤中,模型通过使用余弦相似度计算每个句子的语义向量和目标摘要的相似度,对句子进行评分。通过束搜索,每次选择前 N 个最相关的句子作为候选,并将其作为下一步的输入。

(3)为了确保返回的句子索引合法且与原文内容高度相关,本文在句子选择过程中加入了边界检查和去重机制。这不仅避免了重复句子的出现,还确保了最终选择的句子符合原文的逻辑结构。

(4)束搜索通过逐步选择并优化候选句子,最终输出最优的句子序列,作为摘要生成的输入。这些句子代表了文本中最关键的信息,并与目标摘要的内容高度契合。束搜索能够在更大范围内探索文本内容,从而捕捉更多的关键信息,并有效避免了贪婪搜索中可能忽视一些重要句子的风险。

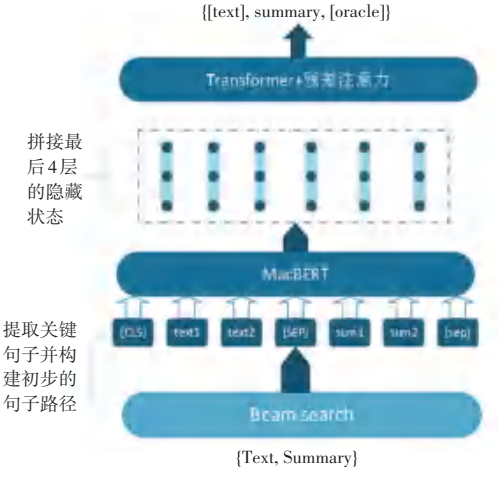

图 2 改进 Oracle 抽取的结构图

Fig. 2 Structure diagram of improving Oracle extraction

整体的训练 *Loss* 采用 *MSELoss*,计算的是预测特征和随机生成目标特征之间的均方误差。公式定义为:

$$MSELoss = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{1}$$

其中,N 表示样本数量;y_i 表示实际目标特征;$\hat{y}_i$ 表示模型预测特征。

关于抽取实验结果可以在 3.4.1 节中进行查看。

2.2 改进评分机制

RAG(Retrieval-Augmented Generation)[23] 模型是一种结合检索与生成的模型,主要用于处理长文档或多文档的生成任务。DYLE 基础模型中的 RAG 由 2 个主要部分组成:检索器和生成器。检索器负责从文档库中检索与输入相关的文档片段,而生成器则基于检索到的相关文档片段生成最终的文本。

本文在动态文档评分机制上进行了改进,主要体现在以下 2 个方面:

(1)引入多层注意力层和残差连接。这 2 项改进的目标是提升模型的表示能力、加速收敛,并增强模型对复杂数据的理解能力。

① 多层注意力层:注意力机制通过自注意力层能够根据输入数据的上下文信息,动态调整各个部分之间的依赖关系。这使得模型在处理长文档或多文档时,能够有效地捕捉跨文档和跨句子的语义联系,尤其适用于复杂文本的生成任务。

② 残差连接:残差连接在每一层的输入与输出之间建立直接的连接,有效地缓解了梯度消失问题,使得模型在训练过程中能够更稳定地传播梯度,尤其是在多层结构的情况下。这不仅帮助模型提高了训练效率,还使得更深层次的模型能够在不损失信息的情况下,进行复杂的特征转换。

(2)改进动态的多层感知机(MLP)结构。改进后的动态 MLP 结构结合了 Transformer 编码器和线性输入层,旨在提升模型对序列数据的处理能力和表达能力。

① Transformer 编码器:为了更好地处理输入的序列数据,MLP 的输入首先通过 Transformer 编码器进行特征提取。Transformer 能够有效捕捉序列中的长期依赖关系,并通过自注意力机制进行全局信息建模,从而增强对输入特征的理解。

② 线性输入层:在 Transformer 编码器的输出之后,使用线性层进行维度变换,以便适应后续的任务目标(如分类或生成任务)。这一结构使得 MLP 能够在保持 Transformer 强大建模能力的基础上,进一步对特定任务进行定制。

③ 多层结构:在 MLP 中使用了多层设计,这些层次之间的非线性变换可以进一步增强模型的表达能力,尤其是对于复杂任务的建模能力。

评分机制如图 3 所示。数据通过 Jieba 中文分词等预处理进入以 LawFormer 为主的检索器中，基于输入的问题，从文档库中检索相关文档或文档片段。之后进入以 BART 为主的生成器中，利用检索到的文档片段生成摘要或答案，调用预训练模型 BART 进行文本生成。该部分负责将文档片段输入到生成器模型中，通过生成网络输出最终的生成文本。数据流再经过多层注意力层，对生成器输出进行进一步处理，基于多头注意力机制对生成文本的上下文进行加权和处理。通过多层自注意力模块来强化序列中的长期依赖关系。此后经过残差连接层，将注意力层的输出与输入进行相加，以解决深层网络中的梯度消失问题，帮助模型更好地学习深层表示。最后经过动态多层感知机（MLP），根据处理过的特征进一步计算每个文档的相关性得分。生成最终的模型输出，包括预测的文本（logits）、以及动态评分的结果（dynamic_scores）。

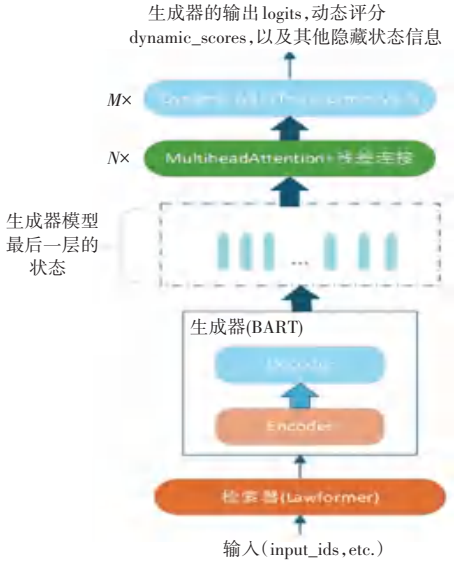

图 3　评分机制示意图

Fig. 3　Schematic diagram of scoring mechanism

对于上述改进的效果，以下 3.4.2 节对比实验和 3.4.3 节消融实验将展示其在模型性能上的明显提升。

3　实验及分析

3.1　数据集

本文采用的数据集是“法研杯”CAIL2020 提供司法摘要数据。CAIL2020 数据集文本和摘要长度统计见表 2。包含 4 047 篇裁判文书，其内容全部为民事案件类型。民事案件涉及到许多法律术语和复杂的事实分析，能够很好地代表司法领域的文本处理任务。对于数据的处理，本文使用了 openrefine 的工具，原训练数据还附带标签信息，但为了模型的通用性，通过该工具将标签信息删除。对于特殊字符、乱码、格式错误等，使用正则表达式进行清理，确保文本格式规范。经过数据预处理后，将数据集按照 8∶1∶1 的形式分为训练集 3 237 条，验证集 404 条，测试集 406 条。

表 2　CAIL2020 数据集文本和摘要长度统计

Table 2　CAIL2020 dataset text and abstract length statistics

内容	最长字数	最短字数	平均字数	字数标准差
text	14 413	1 001	2 629	1 221
summary	1 594	94	273	51

3.2　评价方法

摘要生成的评价方法采用 ROUGE 方法，ROUGE 评估方法主要通过计算生成摘要与参考摘要之间的重叠度来衡量摘要质量。本实验中采用 ROUGE 评价方法中的 ROUGE－1、ROUGE－2 和 ROUGE－L。

（1）ROUGE－1。计算生成摘要和参考摘要之间的单词重叠，主要反映摘要的覆盖度。

（2）ROUGE－2。评估的是生成摘要和参考摘要之间的二元组重叠，适用于衡量语义上较为复杂的匹配。

（3）ROUGE－L。计算最长公共子序列（LCS）的重叠，能够评估生成摘要在结构上的连贯性和表达能力。

其中，ROUGE 中的召回率（*Recall*，R）衡量的是生成摘要覆盖了多少参考摘要中的内容。精确率（*Precision*，P）衡量的是生成摘要中有多少信息与参考摘要重叠。$F1$－*Score* 是精确率和召回率的调和平均数，在这两者之间取得平衡。通过结合这 3 个指标，可以全面评估生成摘要在信息提取（召回）、信息准确性（精确率）以及整体质量（$F1$－*Score*）方面的表现。研究中可用如下公式进行计算：

$$R = \frac{N_L}{R_T} \tag{2}$$

$$P = \frac{N_L}{A_T} \tag{3}$$

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{4}$$

其中，N_L 表示生成摘要与参考摘要的重叠部分；R_T 表示参考摘要的总量；A_T 表示生成摘要的总量。

3.3 实验环境及设置

本实验的实验环境,GPU 是 NVIDIA GeForce RTX 4090,Python 是 3.9 版本,Pythorch 为 1.8 版本,CUDA 的版本为 11.1。Oracle 抽取模型的主要参数设置见表 3。

表 3 Oracle 抽取模型的主要参数设置

Table 3 Main parameter settings for Oracle extraction model

名称	数值
beam size	40
batch size	30
d_model	3 072
num_encoder_layers	6

在抽取-生成的整体模型的参数主要设置:梯度裁剪的最大范数为 1.0,避免梯度爆炸问题。分类和生成任务的学习率为 5e-5,适合大多数 NLP 任务。优化器为 Adam,是深度学习中广泛使用的优化算法,有助于处理稀疏梯度问题并加速收敛过程。抽取-生成的整体模型的主要参数见表 4。

表 4 抽取-生成的整体模型的主要参数

Table 4 Main parameters of the extracted generated overall model

名称	数值
train_batch_size	4
eval_batch_size	1
test_batch_size	1
top_k	20
max_source_len	512

3.4 实验结果及分析

3.4.1 抽取实验结果

下面是关于数据在贪婪搜索、束搜索以及上述提到的改进 Oracle 方法的实验数据。实验数据的结果是以原始数据按照比例进行划分的训练数据为主。上述实验设置的 *beam size* 为 40,在本研究中,通过实验确定了束宽的最佳值。通过交叉验证和 ROUGE 评估,选择了束宽为 40 作为最优配置。实验结果显示,在所有方法中,改进 Oracle 方法的性能最佳,ROUGE 分数均高于贪婪搜索和束搜索。束搜索相较于贪婪搜索也表现出显著的提升。这表明采用更复杂的搜索策略可以有效提高摘要生成的质量。不同方法在训练集的抽取结果展示见表 5。

图 4~图 6 是改进的 Oracle 抽取方法在训练数据集上的实时的 ROUGE-1,ROUGE-2,ROUGE-L 分数变化, loss 变化如图 7 所示。

表 5 不同方法在训练集的抽取结果展示

Table 5 Display of extraction results of different methods in the training set

方法	*AVG_ROUGE* - 1	*AVG_ROUGE* - 2	*AVG_ROUGE* - *L*
贪婪搜索	0.034	0	0.021
束搜索	7.667	4.433	7.465
改进 Oracle 抽取	**13.729**	**5.237**	**12.418**

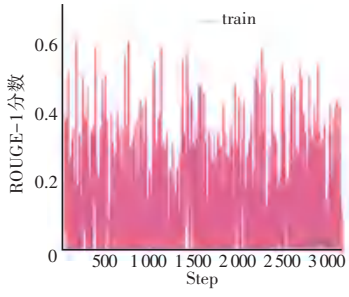

图 4 改进 Oracle 抽取 ROUGE-1 分数

Fig. 4 ROUGE-1 scores extracted by improved Oracle

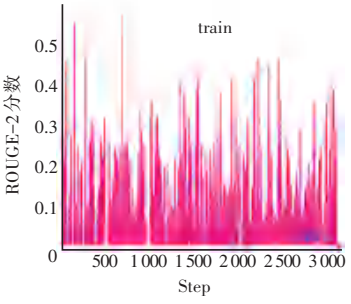

图 5 改进 Oracle 抽取 ROUGE-2 分数

Fig. 5 ROUGE-2 scores extracted by improved Oracle

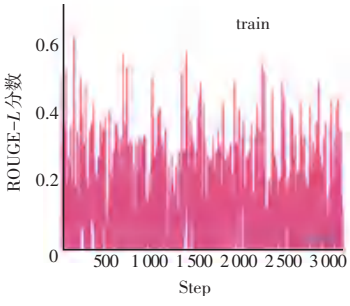

图 6 改进 Oracle 抽取 ROUGE-*L* 分数

Fig. 6 ROUGE-*L* scores extracted by improved Oracle

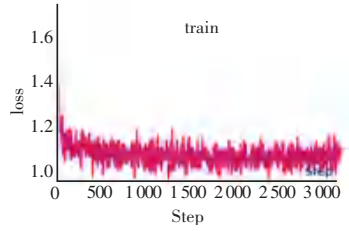

图 7 改进 Oracle 抽取实时 loss 变化

Fig. 7 Real-time loss changes extracted by improved Oracle

通过实验结果可以得到结论，改进后的 Oracle 抽取可以识别和抽取出尽可能多的高分句子。这一差异的主要原因在于改进 Oracle 抽取方法能够在全局语义信息和句子多样性之间取得更好的平衡。具体来说，贪婪搜索每次选择当前最优的句子，虽然能迅速找到高评分句子，但可能会错过一些全局相关性强的句子，导致摘要质量受到限制。而束搜索通过保留多个候选解，可以在一定程度上缓解这一问题，但由于其局部搜索特性，仍无法确保全局最优解。相比之下，改进后的 Oracle 抽取方法通过综合考虑句子的全局语义和语义多样性，能够在抽取过程中识别出更多相关且信息丰富的句子。

3.4.2　对比实验

为了进一步验证改进模型的有效性，下面选择一些文本摘要的模型进行对比，同时还有基础模型 DYLE。

(1) PageRank[24]：PageRank 是一种基于图的算法，将每个句子视为图中的一个节点，句子之间的关系通过边来表示。这些边的权重由句子间的相似性或相关性决定。在文本摘要中，PageRank 算法通过计算每个句子的“重要性”来选择摘要句子，然而，PageRank 依赖于局部相似性计算，未能充分捕捉文本的全局语义信息。

(2) Pointer-Generator Network[25]：该模型结合了指针机制和生成机制，旨在增强生成模型的灵活性。指针机制可以直接从源文本中选择句子，而生成机制则用于生成新的文本。尽管 Pointer-Generator Network 能较好地处理长文本中的信息重复问题，但其生成的摘要可能缺乏全局语义连贯性，尤其是在复杂的法律文本或多文档摘要任务中。

(3) BART：BART 结合了双向和单向自回归编码器，通过对噪声输入文本的去噪重构进行预训练。这使得 BART 在生成自然语言文本时具有很好的生成能力。然而，BART 在处理长文本时可能存在语义捕捉不足的情况，尤其是在文本的细节和长距离依赖关系的捕捉上。

(4) BERT_SUM[26]：BERT_SUM 是基于 BERT 模型的文本摘要生成方法，利用 BERT 预训练模型来生成文本摘要。其优点在于能够有效捕捉文本中的上下文信息，但由于 BERT 主要通过掩码机制进行训练，对于长文本的全局语义理解和精确生成仍然存在一定的挑战，尤其在法律文本这样的高结构化内容中，仍可能无法有效识别关键信息。

上述的实验结果见表 6。从实验结果可以看出，改进后的模型在 ROUGE 指标的 *Recall*（*R*）、*Precision*（*P*）和 *F*1 分数上均超过了原模型 DYLE，并且在与其他摘要模型的比较中也表现出明显的优势。

表 6　不同摘要模型在 CAIL2020 数据集上的性能比较

Table 6　Performance comparison of different abstract models on CAIL2020 dataset

Model	ROUGE-1			ROUGE-2			ROUGE-*L*		
	R	*P*	*F*1	*R*	*P*	*F*1	*R*	*P*	*F*1
PageRank	7.58	8.19	6.65	3.39	2.88	2.65	7.23	7.79	6.34
BERT+PGN	14.96	20.90	15.93	5.58	5.41	4.95	14.65	20.66	15.67
BART	13.56	23.19	15.02	7.34	12.71	8.10	13.33	22.56	14.69
BERT_SUM	35.15	27.62	30.04	12.81	10.45	11.18	23.59	19.60	20.89
DYLE(original)	32.69	26.53	23.88	19.44	14.05	13.06	32.33	26.89	23.72
DYLE(our)	**49.22**	**28.47**	**31.79**	**24.90**	**14.90**	**15.50**	**48.94**	**28.27**	**31.57**

与原模型相比，明显可以看出 ROUGE-1 的 *Recall* 提升 16.53%，*F*1 分数提高 7.91%。这表明改进后的模型在覆盖更多重要信息的同时，能更好地控制冗余信息，生成的摘要更加简洁、有效。此外，ROUGE-2 的 *Recall* 也提升 5.46%，表明改进模型的表现表明其在细粒度信息的捕捉上有了更好的表现。ROUGE-*L* 的 *Recall* 提升了 16.61%，*F*1 分数提高 7.85%。表明其在处理长文本和长距离依赖信息方面表现更加出色。通过对 Oracle 抽取方法的改进以及在检索生成模型中引入 Transformer 模块，上面的 2 项措施显著提高了对裁判文书等长文本的上下文理解和全局语义理解，基本解决了长

文本编码及信息冗余的问题。

3.4.3 消融实验

为了验证上述 2 个改进方法对抽取-生成模型摘要生成的有效性,本文进行了消融实验,具体结果见表 7。实验包括以下 2 种组合。

(1)改进的 Oracle 抽取法与只有动态 MLP 动态检索模型进行组合实验,该实验旨在评估改进的 Oracle 抽取法在与动态 MLP 模型结合时的效果。

(2)普通的 Beam search 与加入 TransfoermerMLP 的检索生成模型进行组合实验。该实验用于比较传统 Beam Search 与引入 Transformer MLP 后模型性能的差异。

实验结果表明,加入 Transformer MLP 的检索生成模型在文本摘要生成任务中的性能提升显著,在 ROUGE-1 的 *Recall* 性能提高 6.96%, *Precision* 性能提高 7.57%。ROUGE-2 的 *Precision* 提高 5.4%, ROUGE-*L* 的 *Recall* 性能提高 7.06%, *Precision* 性能提高 7.8%。说明该改进对模型的贡献较高。

表 7 消融实验结果:不同模型组合在 CAIL2020 数据集上的性能比较

Table 7 Experimental results of ablation: Performance comparison of different model combinations on the CAIL2020 dataset

Model	ROUGE-1			ROUGE-2			ROUGE-*L*		
	R	*P*	*F*1	*R*	*P*	*F*1	*R*	*P*	*F*1
Oracle+MLP	34.00	20.26	21.47	17.90	9.61	9.86	33.66	19.88	21.17
Beam+Transformer	40.96	27.83	28.61	22.28	15.01	14.98	40.72	27.68	28.43

原模型 DYLE 对于长文本的处理是进行分块,输入由 L 个文本片段组成,$x = (x_1, \cdots, x_l)$。这些文本块中,检索模型通过动态评分来评估每个文本块的相关性,然后将结果传递给生成模型,以指导摘要的生成。因此,加入 Transformer MLP 和注意力机制的检索生成模型可以增强文本块之间的关联性,提升了模型对长文本的上下文捕捉能力。Transformer 的自注意力机制可以帮助模型理解文本中的长距离依赖关系,并有效提高生成摘要的语义一致性和精确性,尤其是在需要进行细粒度语义理解的任务中。

综上所述,上述 2 种改进方法均对模型的摘要生成性能提升起到了积极作用。将这 2 种方法结合使用,能够显著改善生成效果。

下面是以 ROUGE-1 为例,展示不同的模型组合在 CAIL2020 数据集消融实验中 ROUGE-1 的 *Recall* (*R*)、*Precision* (*P*) 和 *F*1 分数的实时变化。结果如图 8 ~ 图 10 所示。

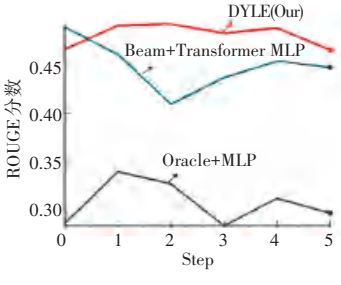

图 8 不同组合在数据集中 ROUGE-1 的 *Recall* 分数变化

Fig. 8 Changes in *Recall* scores of ROUGE-1 for different combinations in the dataset

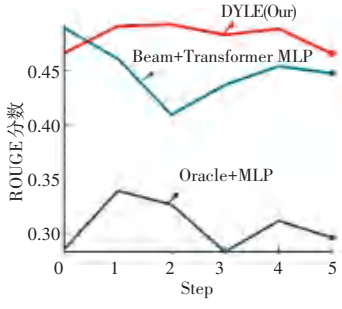

图 9 不同组合在数据集中 ROUGE-1 的 *Precision* 分数变化

Fig. 9 Changes in *Precision* scores of ROUGE-1 for different combinations in the dataset

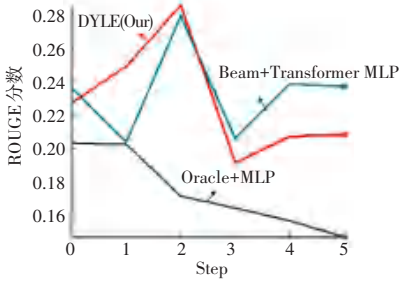

图 10 不同组合在数据集中 ROUGE-1 的 *F*1 分数变化

Fig. 10 Changes in *F*1 scores of ROUGE-1 for different combinations in the dataset

4 结束语

本文采用了抽取-生成两阶段的文本摘要生成方法,在原模型 DYLE 的基础上,通过改进 Oracle 抽取方法,从文本中选择高分句子并将其索引映射回原始文本。这一过程为抽取模型的训练提供了关键信息和全局语义指导。在检索模型的评分机制上,

引入了 Transformer 结构，从而提高了模型的表达能力和上下文理解能力。实验结果表明，这 2 个关键改进不仅显著提升了摘要的准确性、信息覆盖度和语义连贯性，还有效解决了原模型在处理长文本时的冗余信息问题。尽管如此，本文提出的方法仍然存在一些局限性和挑战。例如，模型在专有名词识别方面的表现仍然不尽人意，尤其是在处理法律文书等高结构化文本时，专有词汇的抽取和理解仍然是一个难题。此外，尽管引入了 Transformer 结构提升了上下文理解能力，但模型的计算复杂度和训练时间也有所增加。

未来的研究方向可以从以下几个方面进行改进和拓展：首先，在专有名词识别方面，可以考虑引入更多领域特定的知识，如法律术语、专有名词库等，结合命名实体识别（NER）技术进行优化；其次，可以通过模型压缩、量化等技术提高模型的效率，减少计算资源消耗；最后，未来的工作还可以探索更为先进的模型架构，如基于图神经网络（GNN）或多模态学习的模型，以进一步提升模型在复杂文档摘要生成中的表现。

参考文献

[1] 魏鑫炀，唐向红. 基于 BERT 的抽取式裁判文书摘要生成方法研究[J]. 软件工程，2022，25(5)：1-4.

[2] VASWANI A，SHAZEER N，PARMAR N，et al. Attention is all you need[J]. arXiv preprint arXiv，1706，03762，2017.

[3] TAY Y，DEHGHANI M，ABNAR S，et al. Long Range Arena：A Benchmark for Efficient Transformers[J]. arXiv preprint arXiv，2011. 04006，2020.

[4] ROHDE T，WU X，LIU Y. Hierarchical learning for generation with long source sequences[J]. arXiv preprint arXiv，2104. 07545，2021.

[5] DAI Zihang，YANG Zhilin，YANG Yiming. Transformer-xl：Attentive language models beyond a fixed-length context[J]. arXiv preprint arXiv，1901. 02860，2019.

[6] KITAEV N，KAISER Ł，LEVSKAYA A. Reformer：The efficient transformer[J]. arXiv preprint arXiv，2001. 04451，2020.

[7] BELTAGY I，PETERS M E，COHAN A. Longformer：The long-document transformer[J]. arXiv preprint arXiv，2004. 05150，2020.

[8] MAO Ziming，WU C H，NI Ansong，et al. DYLE：Dynamic latent extraction for abstractive long-input summarization[J]. arXiv preprint arXiv，2110. 08168，2021.

[9] CAJUEIRO D O，NERY A G，TAVARES I，et al. A comprehensive review of automatic text summarization techniques：method，data，evaluation and coding[J]. arXiv preprint arXiv，2301. 03403，2023.

[10] LUHN H P. The automatic creation of literature abstracts[J]. IBM Journal of research and development，1958，2(2)：159-165.

[11] CARBONELL J，GOLDSTEIN J. The use of MMR，diversity-based reranking for reordering documents and producing summaries[C]//Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York：ACM，1998：335-336.

[12] WU C W，LIU Chaolin. Ontology-based text summarization for business news articles[C]//Proceedings of the ISCA 18th International Conference Computers and Their Applications. Hawaii，USA：dblp，2003：389-392.

[13] SVORE K，VANDERWENDE L，BURGES C. Enhancing single-document summarization by combining RankNet and third-party sources[C]//Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning（EMNLP-CoNLL）. Prague，Czech Republic：dblp，2007：448-457.

[14] GU Nianlong，ASH E，HAHNLOSER R H R. MemSum：Extractive summarization of long documents using multi-step episodic Markov decision processes[J]. arXiv preprint arXiv，2107. 08929，2021.

[15] GENEST P E，LAPALME G. Fully abstractive approach to guided summarization[C]//Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics（Volume 2：Short Papers）. ACL，2012：354-358.

[16] LIU Yang，LAPATA M. Text summarization with pretrained encoders[J]. arXiv preprint arXiv，1908. 08345，2019.

[17] ZHANG Jingqing，ZHAO Yao，SALEH M，et al. Pegasus：Pre-training with extracted gap-sentences for abstractive summarization[J]. arXiv preprint arXiv，1912. 08777，2019.

[18] LEWIS M. Bart：Denoising sequence-to-sequence pre-training for natural language generation，translation，and comprehension[J]. arXiv preprint arXiv，1910. 13461，2019.

[19] ZAHEER M，GURUGANESH G，DUBEY K A，et al. Big bird：Transformers for longer sequences[J]. Advances in Neural Information Processing Systems，2020，33：17283-17297.

[20] 刘迪，奚雪峰，崔志明，等. 抽取-生成式自动文本摘要技术研究综述[J]. 计算机技术与发展，2023，33(5)：1-8.

[21] XU Yumo，LAPATA M. Text summarization with oracle expectation[J]. arXiv preprint arXiv，2209. 12714，2022.

[22] CUI Yiming，CHE Wanxiang，LIU Ting，et al. Revisiting pre-trained models for Chinese natural language processing[J]. arXiv preprint arXiv，2004. 13922，2020.

[23] LEWIS P，PEREZ E，PIKTUS A，et al. Retrieval-augmented generation for knowledge-intensive nlp tasks[J]. Advances in Neural Information Processing Systems，2020，33：9459-9474.

[24] BRIN S，PAGE L. The anatomy of a large-scale hypertextual Web search engine[J]. Computer Networks and ISDN Systems，1998，30(1-7)：107-117.

[25] SEE A，LIU P J，MANNING C D. Get to the point：Summarization with pointer-generator networks[J]. arXiv preprint arXiv，1704. 04368，2017.

[26] LIU Yang. Fine-tune BERT for extractive summarization[J]. arXiv preprint arXiv，1903. 10318，2019.