第 13 卷 第 1 期
2024 年 1 月
集 成 技 术
JOURNAL OF INTEGRATION TECHNOLOGY
Vol. 13 No. 1
Jan. 2024

引文格式：
温嘉宝, 杨敏. 面向中文法律裁判文书的抽取式摘要算法 [J]. 集成技术, 2024, 13(1): 62-71.
Wen JB, Yang M. Extractive summarization algorithm for Chinese legal judgment documents [J]. Journal of Integration Technology, 2024, 13(1): 62-71.

面向中文法律裁判文书的抽取式摘要算法

温嘉宝[1,2] 杨 敏[1*]

[1]（中国科学院深圳先进技术研究院 深圳 518055）
[2]（中国科学院大学 北京 100049）

摘 要 裁判文书自动摘要的目的在于让计算机能够自动选择、抽取和压缩法律文本中的重要信息，从而减轻法律从业者的工作量。目前，大多数基于预训练语言模型的摘要算法对输入文本的长度存在限制，因此无法对长文本进行有效摘要。为此，该文提出了一种新的抽取式摘要算法，利用预训练语言模型生成句子向量，并基于 Transformer 编码器结构融合包括句子向量、句子位置和句子长度在内的信息，完成句子摘要。实验结果显示，该算法能够有效处理长文本摘要任务。此外，在 2020 年中国法律智能技术评测（CAIL）摘要数据集上进行测试的结果表明，与基线模型相比，该模型在 ROUGE-1、ROUGE-2 和 ROUGE-L 指标上均有显著提升。

关键词 抽取式摘要模型；法律裁判文书；文本自动摘要；深度神经网络
中图分类号 TP 399 **文献标志码** A doi: 10.12146/j.issn.2095-3135.20230209001

Extractive Summarization Algorithm for Chinese Legal Judgment Documents

WEN Jiabao[1,2] YANG Min[1*]

[1](*Shenzhen Institute of Advanced Technology*, *Chinese Academy of Sciences*, *Shenzhen* 518055, *China*)
[2](*University of Chinese Academy of Sciences*, *Beijing* 100049, *China*)
[*]*Corresponding Author*: *min.yang@siat.ac.cn*

Abstract The purpose of automatic judgment document summarization is to allow computers to automatically select, extract, and compress important information from legal texts so as to reduce workload of practitioners. Currently, most summarization algorithms based on pre-trained language models have limitations on the length of the input text, so they cannot effectively summarize long texts. In this thesis, an innovative extractive summarization algorithm is introduced, which uses a pre-trained language model to generate sentence vectors. Based on the Transformer encoder structure, the summarization task can be completed by fused information

收稿日期：2023-02-09 **修回日期**：2023-03-24
基金项目：深圳市基础研究重点项目（JCYJ20210324115614039）
作者简介：温嘉宝，硕士研究生，研究方向为自然语言处理；杨敏（通讯作者），副研究员，研究方向为自然语言处理和推荐系统等，E-mail：min.yang@siat.ac.cn。

including sentence vectors, position and length of sentences. Experimental results showed that, the algorithm can effectively handle the task of summarizing long texts. In addition, the model was tested on the 2020 CAIL (challenge of AI in law) summarization dataset, and results showed that compared to the baseline model, the proposed model showed significant improvement in the ROUGE-1, ROUGE-2, and ROUGE-L metrics.

Keywords　extractive summarization model; legal judgment documents; automatic text summarization; deep neural network

Funding　This work is supported by Shenzhen Basic Research Foundation (JCYJ20210324115614039)

1　引　言

随着人民群众法律意识的提高和案件数量增长速度的加快，截至 2023 年 1 月，中国裁判文书网已公布超过 1.3 亿份的裁判文书，并以每日数千份的速度继续增加。这些文书包括民事、刑事、行政、赔偿、执行等多种类型，经过筛选和专业处理后被公布，形成了法律领域中数据量最大的数据库，为司法智能化、信息化提供了强有力的数据基础。优质裁判文书的不断增加对司法领域的发展具有重要意义，如可以为法律从业者提供更多的案例参考，缓解“同案不同判”的困境。然而，这也带来了一些新的问题。

裁判文书是法律从业者日常工作中接触的重要资料之一，包括进行类案检索和撰写类案检索报告。随着裁判文书数据量的增加，法律从业者需要从检索出的大量裁判文书中挑选出最合适的裁判文书。然而，裁判文书通常较长，平均长度可达数千字，甚至有少数文件长达上万字，从中查找出关键信息并进行分析无疑变得越来越困难。因此，对裁判文书进行自动摘要，以去除冗余信息、精简文本内容，能极大程度地减少法律从业者的工作量，使得他们有更多的精力从事更有价值的工作，从而提升司法行业的效率。由此可见，裁判文书自动摘要对司法领域有着重要研究价值和现实意义。

根据摘要方式，自动摘要模型可分为抽取式模型和生成式模型，本文提出的方法属于抽取式模型。抽取式模型通过从原文中直接选择若干个重要句子进行排列重组，以形成摘要。根据学习方式的不同，抽取式摘要算法可进一步划分为无监督式抽取和有监督式抽取。无监督抽取式文本摘要方法因运行速度快，且无须人工标注训练数据而广受应用。相比之下，有监督抽取式文本摘要方法的优势在于其具有更高的准确性。

2　抽取式摘要研究现状

抽取式摘要是一种直接从原文中抽取关键句的方式，这种方式在句法上错误率低。从学习方式上，抽取式摘要算法可以分为无监督和有监督两大类。无监督抽取式摘要通常采用图、聚类等方式。而有监督抽取式摘要多采用基于神经网络的方法。

2.1　无监督抽取式摘要算法

无监督抽取式摘要最简单的实现方式是“Lead-3”法，即从文本前 3 个句子中提取信息，并作为摘要。由于作者通常在文章的标题和开头部分阐述文章的主题(如新闻报道)，因此，这种方式在该类文本中较为有效。然而，法律裁判文书的关键信息分布较为均匀，因此，采用 Lead-3 法进行法律裁判文书的摘要不能达到理想效果。

2004 年，Erkan 等[1]提出的 LexRank 是一种基于图排序的抽取式摘要算法，其以句子为节点，以句子间相似度为边的权值，构建无向有权图。该算法采用词袋模型表示句子向量，维度是目标语言中单词的数量。对于出现在句子中的每个单词来说，句子向量中相应维数的值是该单词的 TF-IDF[2]值。通过计算句子向量与图中质心之间的相似度，判断句子是否为重要句子。其中，两个句子之间的相似性由向量间的余弦相似度定义，而质心是由文档中 TF-IDF 值超过某一阈值的词构成的向量。该算法具有简单和易于实现的优点。然而，它在表示句子向量的方式上存在一定的缺陷。其一，维度较大，导致生成的句子向量非常稀疏。其二，句子向量中相应维数的值是该单词的 TF-IDF 值。由于 TF-IDF 基于词频统计，无法考虑语义信息，因此，LexRank 在判断句子相似性时无法充分考虑语义层面的相似度。

2004 年，Mihalcea 等[3]提出的 TextRank 算法是一种基于图的排序算法，其设计灵感源自于 PageRank[4]网络排序算法。该算法将文档表示为图模型，将文档中的每个句子作为图中的一个节点，节点之间的连边表示句子之间的相关性。然后，通过 PageRank 算法计算每个节点的 TextRank 值，以确定文档中最重要的句子，并选择其中得分最高的几个句子作为摘要。然而，TextRank 的句子相似度衡量方式采用了两个句子之间的共现词数量，即采用了词袋模型，无法考虑同义词、词序等其他信息。这使得 TextRank 算法的表现会受到一定的限制。

2016 年，Padmakumar 等[5]提出了一种基于聚类的抽取算法。首先，利用 Skip Thought Vectors 进行无监督学习，得到句子的嵌入向量。然后，通过聚类算法对生成的句子嵌入向量进行聚类。最后，将距离簇质心最近的向量所对应的句子作为文本摘要。Skip Thought Vectors 的思想是通过一个句子预测它上下文的句子，具体做法是通过 LSTM[6]编码器将中间句子编码为向量，再用两个独立的 LSTM 解码器将句子向量解码出前后句子。这种方式与 2013 年 Mikolov 等[7]提出的 Word2Vec 中的 Skip-gram 训练策略相似，依据的原则是一个句子与其前后相邻句子之间存在语义联系。在该论文中，Padmakumar 等[5]尝试了 K-means 和 Mean-shift 两种聚类方法。然而，该算法的不足之处在于 LSTM 无法实现并行训练，以及在处理长序列时可能面临梯度消失和梯度爆炸的风险。

2021 年，Padmakumar 等[8]提出了一种基于点互信息的摘要算法。该算法利用 GPT-2[9]计算给定两个句子之间的点互信息，而点互信息定义为在给定前句的情况下，得到后句的概率。通过对摘要与原文档中所有句子对的点互信息进行求和，可以得到摘要与原文档的相关性。同时，通过对摘要内部所有句子对的点互信息进行求和，可以确定摘要的冗余性。最终，摘要由一个句子集合组成，该集合能最大化相关性减去冗余性的值。该算法的优点在于利用预训练语言模型计算句子间的点互信息，从而实现无监督抽取式摘要。然而，缺点在于其时间复杂度较高，需要计算所有句子之间的点互信息。当文档句子数量较大时，推理时间将会较长。

2.2 有监督抽取式摘要算法

2017 年，Nallapati 等[10]提出了一种名为 SummaRuNNer 的方法，将文本摘要任务转化为序列标注问题。对文本中的每个句子都进行二分类(0 或 1)：0 表示不纳入摘要，1 表示纳入摘要。最终的文本摘要由标记为 1 的句子组成。该模型包含两个双向门控循环单元[11]：第一个双向门控循环单元对句子进行词级建模，以获得词级表示，接着，对句子中各词的词级表示求平均，得到句子嵌入；第二个双向门控循环单元则对句子嵌入进行句级建模，以获取句级表示。最后，通过一个分类器对句级表示进行二分类，得到最

终的摘要结果。该模型的优点在于，它不仅双向考虑了句子的局部和全局信息，还充分考虑了句子与文档的关系、句子与前后句子的关系、绝对位置和相对位置等因素。然而，该模型的缺陷在于采用循环神经网络(recurrent neural network，RNN)进行特征提取，导致无法进行并行训练。此外，在处理长序列时，模型可能会面临梯度消失和梯度爆炸的风险。

2017 年，Isonuma 等[12]将文本分类任务与摘要任务相结合，以提升摘要效果。该方法首先采用卷积神经网络对句子进行编码，获取句子向量；然后使用基于 RNN 的编码器-解码器框架为每个句子生成摘要概率。具体而言，在编码器中，为每个句子生成隐藏状态；在解码器中，利用前一个句子的摘要概率、句子向量和隐藏状态生成当前句子的隐藏状态，并根据该隐藏状态计算当前句子的摘要概率。接着，以摘要概率为权重，对句子向量进行加权平均，得到文本向量。最后，利用文本向量预测文本的类别。Isonuma 等[12]认为，文本的类别可以被视为文本的粗糙摘要。若模型能根据文本向量准确预测文本类别，则说明模型具有抽取关键信息的能力。因此，可以将摘要概率较高的句子作为文本的摘要。该方法的优点在于利用文本分类任务增强模型的摘要能力。然而，该方法也存在一些缺点：(1)卷积神经网络主要通过局部卷积操作捕捉文本中的局部特征，对长距离依赖关系的捕捉能力较弱；(2)卷积神经网络的卷积操作对输入的顺序不敏感，在捕捉文本中的词序信息方面存在局限；(3)基于 RNN 的编码器-解码器架构无法并行计算，且存在梯度消失和梯度爆炸的风险；(4)计算过程为单向操作，在计算当前句子摘要概率时，无法考虑后续句子的信息。

2019 年，Liu[13]首次将 BERT[14]应用于抽取式摘要任务，提出了名为 BERTSUM 的方法。即在每个句子前插入[CLS]词元，句子后添加[SEP]词元，最终将每个[CLS]对应的输出视为每个句子的句子向量。Liu[13]采用了 3 种方式对句子进行分类：(1)连接线性层和 Sigmoid 函数，计算句子的重要性得分；(2)句子向量表示单独接入 Transformer[15]进行分类；(3)将句子向量表示单独接入 LSTM 进行分类。BERTSUM 的优势在于充分利用了预训练语言模型的强大特征提取能力。通过对输入数据进行简单的预处理和微调预训练语言模型，便可完成抽取式摘要任务。此方法具有实现相对简单、训练代价较低等优点。然而，BERTSUM 也存在一定的缺陷，即需要将整篇文本一次性输入到 BERT 中，当文本长度超出模型输入限制时，则难以完成摘要任务。

2020 年，Zhong 等[16]提出了 MatchSum 模型，将摘要任务转化为文本匹配任务。该模型利用预训练语言模型对文本进行编码，通过比较文档上下文表示与真实摘要及候选摘要的上下文表示，计算相似度并更新参数。模型认为目标摘要与原文档之间的相似度应最为接近，因此，当存在比目标摘要更接近原文档的摘要时，计算损失并更新参数。文档的候选集由多个句子的所有组合构成。为防止文档句子过多导致组合爆炸，Zhong 等[16]采用 BERTSUM 进行粗略摘要，将部分不重要的句子剔除。在推理阶段，选择与原文档语义相似度最高的候选摘要作为摘要结果。该算法的优点在于将摘要任务转换为文本匹配任务，仅需将匹配得分最高的候选结果作为摘要答案。然而，该算法也存在一些缺点：(1)使用 BERTSUM 进行粗略摘要，在处理长文本时，可能会截断文本，导致信息丢失；(2)不适用于处理句子较多的文本，当文本句子数量较大时，候选摘要集规模也会很大，将增加计算成本。

2022 年，Shi 等[17]提出了一个基于星形架构的抽取式摘要模型 StarSum：首先，通过 BERTSUM 生成每个句子的句子表示；其次，将句子表示与位置嵌入相加；再次，输入星

形 Transformer 进行文档级编码；最后，利用 Sigmoid 函数对最后一层每个句子的输出进行分类，从而得到文本摘要。星形 Transformer 由多个卫星节点和一个星节点组成，构成一个全连接的星形结构。在此结构中，文本序列中第 i 个句子的特征由第 i 个卫星节点的状态表示。星形 Transformer 包括环连接和基本连接两种连接方式。卫星节点通过类似双向 RNN 的环状连接从其邻居节点收集信息(其中，第一个和最后一个卫星节点相互连接)，而星节点则通过基本连接从所有卫星节点获取信息。卫星节点可以通过星节点以两跳的方式实现信息的相互传播。这种架构的优点在于提高了计算效率和处理长期依赖关系的能力。然而，其缺点在于利用 BERTSUM 生成句子向量表示，在处理长文本摘要时，BERTSUM 可能会截断文本，从而导致信息丢失。

3 基于 Transformer 编码器的抽取式摘要算法

本文所提抽取式摘要模型由一个基于 RoBERTa-Large[18]的句子向量生成模型和一个基于 Transformer 编码器的句分类模型组成，最后接入全连接层进行二分类，得到句子重要性，文本摘要则由重要句子组成，如图 1～2 所示。

3.1 基于预训练语言模型的句子向量生成模型

抽取式摘要实际上可以建模为序列标注任务，核心思想是对文本中每个句子进行二分类，0 表示不重要，1 表示重要，所有标签为 1 的句子组成文本摘要。使用预训练语言模型处理文本分类问题常见的方式是在文本前插入[CLS]词元，并使用该词元所对应的输出进行全连接分类。但裁判文书属于长文本，其长度普遍超过常见预训练语言模型的单次输入长度，如 BERT(512 个 token)，甚至会超过一些可以处理长文本的预训练语言模型的单次输入最大长度，如 Longformer[19](4 096 个 token)，因此无法使用在每个句子前插入[CLS]词元，并以[CLS]词元作为句子分类特征的方式。本文所用方法将抽取式摘要分解为句向量生成模型和句分类模型。

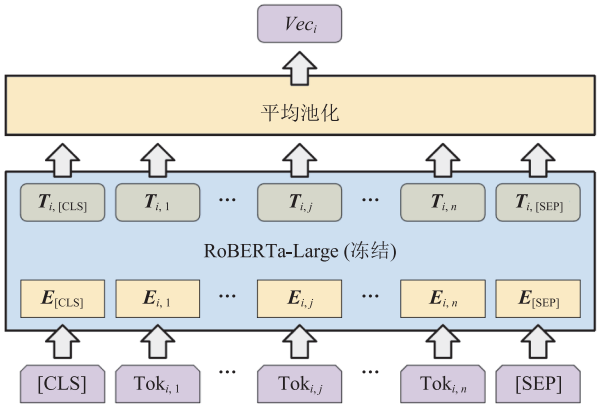

图 1 句向量生成模型

Fig. 1 Sentence vector generative model

基于预训练语言模型的句向量生成模型单次处理一个句子，将预训练语言模型最后一层的输出进行平均池化，得到句向量 $\boldsymbol{Vec}_i$，最终裁判文书的表示为 $\boldsymbol{T}=\{\boldsymbol{Vec}_1,\boldsymbol{Vec}_2,\cdots,\boldsymbol{Vec}_m\}$。这种方式可以有效增加模型可处理文本长度，并减少内存需要，但无法对句向量生成模型进行微调。

3.2 基于 Transformer 编码器的句分类模型

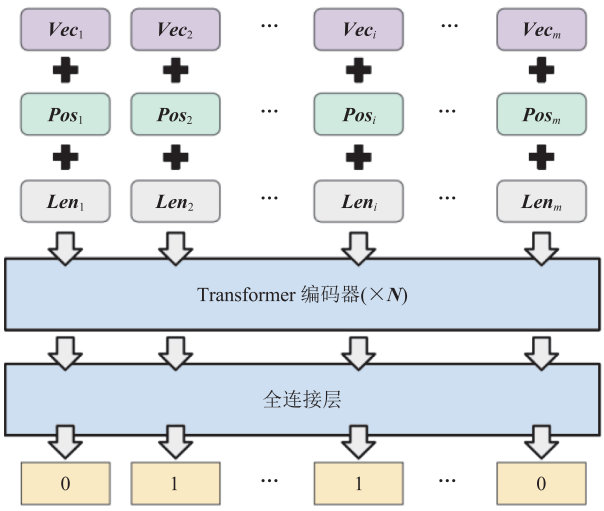

图 2 抽取式摘要模型结构

Fig. 2 Extractive summarization model

第 3.1 小节中用句子向量生成模型对每个句子单独编码，并不包含句子的上下文信息，而抽取式摘要需要考虑到上下文，因此不能直接对所生成的句向量进行分类。

对于一个给定的句子来说，它的输入表示由句子向量、位置嵌入及长度嵌入求和得到。句子向量代表句子所包含的基本语义信息。位置嵌入代表句子在裁判文书中的位置，其中，每个位置对应一个可训练向量。长度嵌入表示当前句子所包含的长度信息，将句子长度按区间划分，每一个长度区间对应同一个可训练向量，计算方式如公式(1)所示。其中，$Len(\text{sentence}_i)$为句子长度；interval 为区间间隔，一般取 5 或 10；idx 为区间下标，对应长度嵌入中具体的一个可训练向量。

$$idx=\frac{Len(\text{sentence}_i)}{\text{interval}} \tag{1}$$

模型训练过程分为 5 步：(1)将一个裁判文书中每个句子的句子向量、位置嵌入及长度嵌入求和，得到每个句子的输入特征；(2)将裁判文书中所有句子输入特征按顺序拼接输入由 Transformer 编码器组成的文本级编码器中；(3)通过多头自注意力机制，从多维度融合句子上下文信息，生成句子的文本级表示；(4)通过全连接层对句子文本级表示进行二分类，得到句子重要性，0 表示不重要，1 表示重要；(5)计算损失，更新模型。句子输入特征计算过程如公式(2)所示。

$$\boldsymbol{x}_i=\boldsymbol{Vec}_i+\boldsymbol{Pos}_i+\boldsymbol{Len}_i \tag{2}$$

其中，$\boldsymbol{Vec}_i$ 为句子向量；$\boldsymbol{Pos}_i$ 为位置嵌入；$\boldsymbol{Len}_i$ 为长度嵌入。模型计算过程如公式(3)所示。

$$z=\text{TransEnc}\left(\left[\boldsymbol{x}_1,\cdots,\boldsymbol{x}_m\right]\right) \tag{3}$$

其中，TransEnc 为多层 Transformer 编码器。模型输出经过 Softmax 函数及对数函数后得到句子重要性得分，如公式(4)所示，其中，$\hat{y}_{i,k}$ 为第 i 个句子属于第 k 类的得分预测值。

$$\hat{y}_{i,k}=\log\left(\frac{\exp\left(z_{i,k}\right)}{\sum_{q=0}^{1}\exp\left(z_{i,q}\right)}\right),k\in\{0,1\} \tag{4}$$

单个文本的损失函数如公式(5)所示。

$$\ell=\left(\hat{y},y\right)=\boldsymbol{L}=\left\{l_1,\cdots,l_n\right\}^{\mathrm{T}},l_i=-w_{y_i}\cdot\hat{y}_{i,y_i} \tag{5}$$

其中，$\hat{y}$ 为预测值；y 为目标值；l_i 为文档中第 i 个句子的损失值；w_{y_i} 为第 i 个句子目标类别所对应权重。由于正负样本不均衡，因此采用惩罚权重，以缓解样本失衡问题。根据正负样本统计结果，正样本权重设置为 3.55，负样本权重设置为 1。$\hat{y}_{i,y_i}$ 为第 i 个句子的目标类别所对应的预测值。

4　实验分析与评估

4.1　数据集与评价指标

本文使用 CAIL2020 摘要数据集作为实验数据，该数据集共收录 13 531 份一审民事判决书，涵盖了侵权责任、借款合同、继承合同、劳动合同、租赁合同等多种民事纠纷类别。样本中的裁判文书预先以多个句子划分，每个句子均有是否重要的标签，同时提供与之对应的全文参考摘要。文书字数平均为 2 586 个，其中，最长的一篇达 14 413 个字，所有文本长度超过 512 个字，99.7% 的文书长度超过 1 024 个字，63.4% 的文书长度超过 2 048 个字。平均每个文书包含 57 个句子，最多的一份达 496 个句子。每篇文书平均抽取 12 个句子作为摘要，最多抽取 69 个句子。每个句子平均包含 44 个字，最长的一个句子长达 640 个字。文书摘要字数平均为 791 个字，最长的一个摘要长达 3 790 个字。数据集中共包含 782 879 个句子，其中，171 745 个句子为重要句子，611 134 个句子为非重要句子，即正负样本的比例为 1∶3.55。该数据集属于长文本摘要数据集。

数据集格式如表 1 所示。其中，id 表示案例

唯一标识；summary 字段表示人工总结的与文档对应的全文参考摘要；text 字段是一个列表，按顺序包含案例中每个句子，text 中的一个元素表示文本的一个带标签的句子，带标签的句子中包含 sentence 和 label；sentence 表示句子具体内容；label 表示这个句子是否重要，0 表示不重要，1 表示重要。本文所研究的方法为抽取式摘要算法，只用到 text 字段，summary 字段属于生成式摘要所需标注的数据，对本文所研究方法作用不大，因此舍弃。

表 1 司法摘要数据集格式

Table 1 Legal summarization dataset format

字段	说明
id	文书唯一标识符
summary	生成式摘要内容
text	包含案例的所有句子，每个句子包含两个字段
-sentence	句子原文
-label	句子重要性，1 表示重要，0 表示不重要

鉴于将抽取任务转化为对句子的分类任务，将 F_1 作为评估指标，比精确率(Precision)和召回率(Recall)更能准确评价一个模型的好坏。F_1 指标如公式(10)所示。为了更好地与基线模型对比，本实验还使用 ROUGE[20] 指标，该指标包含多种度量摘要之间相似性的自动评估方法，是一种常用的文本摘要评价指标。ROUGE-N 的召回率如公式(6)所示。

$$R_{\text{ROUGE}-N}=\frac{\sum_{S\in\{\text{RefSums}\}}\sum_{\text{gram}_n\in S}Count_{\text{match}}\left(\text{gram}_n\right)}{\sum_{S\in\{\text{RefSums}\}}\sum_{\text{gram}_n\in S}Count\left(\text{gram}_n\right)} \tag{6}$$

其中，RefSums 为目标摘要；$Count_{\text{match}}(\text{gram}_n)$ 为句子中 n-gram 的数量；$Count(\text{gram}_n)$ 为目标摘要和预测摘要中 n-gram 共现的次数。ROUGE-N 的精确率如公式(7)所示。

$$P_{\text{ROUGE}-N}=\frac{\sum_{S\in\{\text{PredSums}\}}\sum_{\text{gram}_n\in S}Count_{\text{match}}\left(\text{gram}_n\right)}{\sum_{S\in\{\text{PredSums}\}}\sum_{\text{gram}_n\in S}Count\left(\text{gram}_n\right)} \tag{7}$$

其中，PredSums 为预测摘要。ROUGE-L 的计算公式如公式(8)和公式(9)所示。

$$R_{\text{ROUGE-L}}=\frac{LCS\left(\text{RefSum},\text{PredSum}\right)}{m} \tag{8}$$

$$P_{\text{ROUGE-L}}=\frac{LCS\left(\text{RefSum},\text{PredSum}\right)}{n} \tag{9}$$

$$F_1=\frac{2\times P\times R}{P+R} \tag{10}$$

其中，LCS 用于计算两个字符串最长公共子串的长度；m 为目标摘要长度；n 为预测摘要长度。

本实验的实验超参数说明如表 2 所示。Max_sent_len 表示句子向量生成模型所能处理的最大句子长度。Seq_len 表示句子分类模型能处理的最大句子数量。Pos_weight 和 Neg_weight 表示正例和负例的权重，由于数据集中摘要句子的正负例失衡，负例数量是正例的 3.55 倍，因此，设置正负例权重纠正偏差。Interval 表示长度间隔，模型通过长度间隔引入句子的长度信息，具体做法是将处于相同长度区间内的句子映射到同一个可训练的长度嵌入，具体如公式(1)所示。Layer 表示句子分类模型中 Transformer 编码器的层数。

表 2 实验超参数

Table 2 Hyperparameters in experiment

参数	参数值	说明
Seed	42	随机种子
Max_sent_len	512	RoBERTa-Large 可处理的每个句子最大长度
Seq_len	512	Transformer 可处理的最大句子数
Batch size	32	训练批大小
Pos_weight	3.55	正例权重
Neg_weight	1	负例权重
Dropout	0.1	网络中随机失活参数的比例
Interval	10	长度间隔大小
Hidden size	1 024	隐藏层维度
Layer	4	Transformer 编码器层数
Lr	0.001	学习率
Attn_heads	8	多头注意力机制中 head 数量
Epochs	30	训练轮数
train∶valid∶test	8∶1∶1	训练集、验证集和测试集比例

4.2　实验结果

本文提出的模型将 RoBERTa-Large 作为句子编码器，将多层 Transformer 编码器结构作为句子分类模型，将抽取任务转化为句子的分类任务，并且与无监督 TextRank 基线模型进行了对比，证明了该模型的有效性。此外，本文在抽取式摘要模型中引入了与长度相关的特征，进一步提升了模型效果。实验结果如表 3～5 所示，与基线模型相比，本文提出的抽取式摘要模型在 ROUGE-1、ROUGE-2 和 ROUGE-L 指标上均有明显提升。

4.3　讨论与分析

经过数据分析发现，句子长度信息对句子重要程度有一定影响，因此，本实验引入了句子长度信息。具体方式是将文本长度位于相同区间的文本共享同一个可训练向量，并在文本输入阶段与句子嵌入、句子位置融合，得到句子向量。实验结果如表 3 所示，表中结果为 Transformer 编码器为 4 层的实验结果。由表 3 可知，加入长度信息后，F_1 指标有一定提升。长度间隔为 5 时，F_1 提升 2.586%；长度间隔为 10 时，F_1 提升 2.706%。可以看出，句子长度信息是抽取句子所需的重要信息之一。

表 3　模型在不同长度间隔上实验的结果

Table 3　Experimental results of the model on different length intervals

长度间隔	Recall	Precision	F_1
	0.872 98	0.633 30	0.721 46
10	0.806 38	0.721 92	0.748 52
5	0.791 38	0.732 41	0.747 32

为了选择最优的 Transformer 编码器层数，本研究进行了一系列实验，针对 1～6 层分别计算了 Recall、Precision 和 F_1 指标，结果如表 4 所示。从表 4 可以看出，随着编码器层数的增加，Precision 指标逐渐提高，而 Recall 指标则呈逐渐下降的趋势。然而，F_1 指标与层数之间并没有太大的相关性，这表明层数对模型的整体性能影响有限。在本实验中，当编码器层数为 4 时，模型在 F_1 指标上取得了最佳效果。

表 4　模型在不同 Transformer 编码器层数上实验的结果

Table 4　Experimental results of the model on different Transformer encoder layers

Layer	Recall	Precision	F_1
6	0.774 45	0.738 40	0.741 43
5	0.795 31	0.724 12	0.744 84
4	0.806 38	0.721 92	0.748 52
3	0.837 30	0.686 02	0.741 05
2	0.840 86	0.685 05	0.741 92
1	0.846 87	0.680 48	0.741 65

在与基线模型进行对比的实验中，采用了 ROUGE-1、ROUGE-2、ROUGE-L 指标，实验结果详见表 5。其中，RoBERTa-Large-Transformer 编码器的 Transformer 层数为 4。

Lead-3 模型直接将文档前 3 句话作为摘要，这种方式并不适用于重要信息比较均匀的法律文本，因此其指标并不高。

TextRank 在选择召回分数最高的 15 个句子时表现最佳。由表 5 可知，与无监督的 TextRank 模型相比，RoBERTa-Large-Transformer 编码器在 3 个指标上均有较大提升。其中，ROUGE-1 的 F_1 指标提升 16.44%；ROUGE-2 的 F_1 指标提升 21.87%；ROUGE-L 的 F_1 指标提升 18.05%，3 个指标平均提升 18.79%。

BERTSUM 模型是 BERT 在抽取式摘要中的首次应用。其具体做法是在句子之前插入[CLS]词元，在句子之后插入[SEP]词元，通过预训练语言模型提取特征，根据[CLS]词元对应的输出预测句子的重要性。然而，这种方法的缺点是输入长度受到预训练语言模型的限制。例如，BERT 仅能输入 512 个词元，而即使 Longformer 和 Lawformer 极大程度地增加了模型的可输入长度，数据集中依旧有超过一半的数据超出其长度限制。本实验采用了截断的方式来处理超出长度

限制的部分。

基于 BERT 的 BERTSUM 模型在精确率方面具有较高的表现，但由于信息截断的影响，其召回率较低，因此，F_1 值相对较低。相反，基于 Longformer 的 BERTSUM 模型具有较高的召回率，但精确率偏低，同样导致 F_1 值较低。与基于 Longformer 的 BERTSUM 模型相比，RoBERTa-Large-Transformer 编码器在 ROUGE-1 的 F_1 指标上提升了 22.58%；在 ROUGE-2 的 F_1 指标上提升了 26.42%；在 ROUGE-L 的 F_1 指标上提升了 23.08%，3 个指标的平均提升为 24.03%。

5 结 论

本文针对中文裁判文书摘要任务提出一种基于 Transformer 编码器的抽取式摘要模型。该模型首先通过预训练语言模型采用平均池化的方式为每个句子生成句嵌入；然后通过 Transformer 编码器将句子嵌入、句子位置嵌入及句子长度嵌入融合；最后通过全连接网络对句子表示进行分类，从而完成抽取式摘要任务。本文所提出模型避免了直接将长文本输入模型导致超出预训练语言模型的最大输入长度的问题，从而极大程度地扩大了摘要模型可处理的文本长度。此外，抽取式摘要算法将句子长度以区间划分，并进行映射，从而使模型可以考虑句子的长度信息，提升模型效果。本文所提出的抽取式摘要算法在 ROUGE-1、ROUGE-2、ROUGE-L 指标上均远超过基线模型。

中文法律裁判文书的抽取式摘要任务仍处于早期探索阶段，由于缺乏公开数据集，或是数据集中裁判文书种类不全，裁判文书属于长文本，而常见的预训练语言模型并不支持长文本等原因，该任务依然面临着巨大挑战。而从模型层面，长文本抽取式摘要仍有许多待解决的问题，例如：根据现在的分句标准，句子长度极其不均衡，一些句子长度过长，进行句子嵌入生成时，信息可能会被过度压缩，导致语义信息丢失。而如果对句子进行更细致的分割，则又可能导致少部分文本句子数量过多，从而使得处于末尾部分的位置嵌入无法得到充分训练，并会出现模型抽取少句子的文本效果好，而抽取多句子的文本效果差的情况。这些问题亟待进一步研究解决。

参 考 文 献

[1] Erkan G, Radev DR. LexRank: graph-based lexical centrality as salience in text summarization [J]. Journal of Artificial Intelligence Research, 2004, 22: 457-479.

表 5 模型与基准模型的指标对比

Table 5 Comparison of metrics between the model and the baseline model

模型	ROUGE-1			ROUGE-2			ROUGE-L		
	R	*P*	F_1	*R*	*P*	F_1	*R*	*P*	F_1
Lead-3	0.062 2	0.520 0	0.110 4	0.025 6	0.357 3	0.047 5	0.059 2	0.493 8	0.105 0
TextRank(top10)	0.720 3	0.673 8	0.686 8	0.608 5	0.557 3	0.570 1	0.686 0	0.640 8	0.653 7
TextRank(top15)	0.805 7	0.613 6	0.688 1	0.708 5	0.500 9	0.576 6	0.775 5	0.589 8	0.661 8
TextRank(top20)	0.861 2	0.568 9	0.677 3	0.775 6	0.460 3	0.568 6	0.836 1	0.551 6	0.657 1
BERTSUM(BERT)	0.341 8	0.931 7	0.487 0	0.253 8	0.897 9	0.381 9	0.336 4	0.919 6	0.479 7
BERTSUM(Longformer)	0.938 4	0.477 6	0.626 7	0.905 0	0.383 4	0.531 1	0.915 9	0.465 8	0.611 5
BERTSUM(Lawformer)	0.923 8	0.481 4	0.626 6	0.883 8	0.385 6	0.529 3	0.896 6	0.467 2	0.608 2
RoBERTa-Large-Transformer	0.880 4	0.841 8	0.852 5	0.838 8	0.779 0	0.795 3	0.869 8	0.831 6	0.842 3

[2] Sparck Jones K. A statistical interpretation of term specificity and its application in retrieval [J]. Journal of Documentation, 1972, 28(1): 11-21.

[3] Mihalcea R, Tarau P. TextRank: bringing order into text [C] // Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, 2004: 404-411.

[4] Page L, Brin S, Motwani R, et al. The PageRank citation ranking: bringing order to the web [J]. Stanford Digital Libraries Working Paper, 1998. DOI:10.1007/978-3-319-08789-4_10.

[5] Padmakumar A, Saran A. Unsupervised text summarization using sentence embeddings [J]. Technical Report, University of Texas at Austin, 2016: 1-9.

[6] Hochreiter S, Schmidhuber J. Long short-term memory [J]. Neural Computation, 1997, 9(8): 1735-1780.

[7] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space [Z/OL]. arXiv Preprint, arXiv: 1301.3781, 2013.

[8] Padmakumar V, He H. Unsupervised extractive summarization using pointwise mutual information [R/OL]. 2016. https://www.cs.utexas.edu/~asaran/reports/summarization.pdf.

[9] Radford A, Wu J, Child R, et al. Language models are unsupervised multitask learners [J]. OpenAI Blog, 2019, 1(8): 9.

[10] Nallapati R, Zhai FF, Zhou BW. SummaRuNNer: a recurrent neural network based sequence model for extractive summarization of documents [C] // Proceedings of the AAAI Conference on Artificial Intelligence, 2017.

[11] Cho K, Van Merriënboer B, Bahdanau D, et al. On the properties of neural machine translation: encoder-decoder approaches [Z/OL]. arXiv Preprint, arXiv: 1409.1259, 2014.

[12] Isonuma M, Fujino T, Mori J, et al. Extractive summarization using multi-task learning with document classification [C] // Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017: 2101-2110.

[13] Liu Y. Fine-tune BERT for extractive summarization [Z/OL]. arXiv Preprint, arXiv: 1903.10318, 2019.

[14] Devlin J, Chang MW, Lee K, et al. BERT: pre-training of deep bidirectional transformers for language understanding [Z/OL]. arXiv Preprint, arXiv: 1810.04805, 2018.

[15] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need [C] // Proceedings of the 31st Conference on Neural Information Processing Systems, 2017.

[16] Zhong M, Liu PF, Chen YR, et al. Extractive summarization as text matching [Z/OL]. arXiv Preprint, arXiv: 2004.08795, 2020.

[17] Shi K, Cai XY, Yang LB, et al. StarSum: a star architecture based model for extractive summarization [J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2022, 30: 3020-3031.

[18] Liu YH, Ott M, Goyal N, et al. RoBERTa: a robustly optimized BERT pretraining approach [Z/OL]. arXiv Preprint, arXiv: 1907.11692, 2019.

[19] Beltagy I, Peters ME, Cohan A. Longformer: the long-document transformer [Z/OL]. arXiv Preprint, arXiv: 2004.05150, 2020.

[20] Lin CY. ROUGE: a package for automatic evaluation of summaries [C] // Proceedings of the Text Summarization Branches Out, 2004: 74-81.