Computer Science and Application 计算机科学与应用, 2021, 11(12), 2913-2923
Published Online December 2021 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2021.1112295

Hans汉斯

融合分词和语义感知的中文文本摘要模型

冯正平，王　勇

广东工业大学计算机学院，广东 广州

收稿日期：2021年11月8日；录用日期：2021年12月6日；发布日期：2021年12月13日

摘　要

针对文本摘要生成过程中词组搭配不当、语义表达偏差导致可读性和准确性降低的问题，提出一种融合分词(Word Segmentation, WS)和语义感知(Semantic Awareness, SA)的中文文本摘要模型。编码器使用预训练语言模型，在输入阶段添加中文分词嵌入，获得包含词组信息的语义向量送入解码器；在编解码器间引入语义感知评估，提高摘要的语义契合度。在新闻和科学文献摘要数据集上的仿真结果表明，该模型能有效提高文本摘要的质量。

关键词

文本摘要，分词，语义感知，预训练语言模型

A Chinese Text Summarization Model Combining Word Segmentation and Semantic Awareness

Zhengping Feng, Yong Wang

School of Computer, Guangdong University of Technology, Guangzhou Guangdong

Received: Nov. 8th, 2021; accepted: Dec. 6th, 2021; published: Dec. 13th, 2021

Abstract

Aiming at the problem of improper collocation of phrases and deviation of semantic expression in the process of generating text summarization, the readability and accuracy are reduced. This paper proposes a Chinese text summarization model that combines word segmentation (WS) and semantic awareness (SA). The encoder uses a pre-trained language model to add Chinese word segmentation in the input stage to obtain a semantic vector containing phrase information and send it to the decoder and introduces semantic awareness evaluation between the codecs to im-

文章引用: 冯正平, 王勇. 融合分词和语义感知的中文文本摘要模型[J]. 计算机科学与应用, 2021, 11(12): 2913-2923.
DOI: 10.12677/csa.2021.1112295

prove the semantic fit of the summarization. The simulation results on the news and scientific literature summarization data sets show that the model can effectively improve the quality of text summarization.

Keywords

Text Summarization, Word Segmentation, Semantic Awareness, Pre-Trained Language Model

1. 引言

互联网时代下，文本信息呈现爆炸式增长，信息过载问题日益严重，因此自动文本摘要成为当前一个热点研究课题。

随着机器学习技术在自然语言处理领域广泛应用，Rush 等[1]首次将序列到序列(sequence to sequence, Seq2Seq)模型应用于文本摘要任务中，模型由编码器和解码器组成。针对生成摘要过程中出现语义表达偏差的问题，倪[2]和 Ma [3]等加入语义评价，以提高摘要的语义相关性。Devlin 等[4]提出基于双向 Transformer [5]编码的预训练模型 BERT，双向编码使得每个词向量都包含丰富的上下文语义信息。预训练语言模型在自动文本摘要领域的应用策略主要分为基于特征和基于微调的方法，如 Wang [6]等使用 BERT 提取文本的特征向量作为下游摘要任务的输入，BERT 仅仅作为特征抽取器，参数不随训练过程改变。而 Wei [7]和 Liu [8]等使用微调的训练方法让 BERT 为摘要任务提供先验知识的同时也在训练过程中更新内部参数，后者同时提出 BERTabs 基础框架；大量实验结果证明微调的方法更能凸显预训练模型的作用。针对原 BERT 模型使用词掩码作为预训练任务的局限性，Cui [9]等结合中文分词的特性，提出中文全词掩码预训练方法(Whole Word Masking, WWM)。百度提出 ERNIE [10]模型将字、词组和实体等知识引入到预训练过程中，目的都是使模型学习更多中文词组和实体信息。

受上述已有研究启发，本文提出一种融合分词和语义感知的中文文本摘要模型。模型以预训练语言模型为编码器，在文本输入阶段添加中文分词嵌入编码，极大程度地编码词组和上下文语义信息；使用多层 Transformer 解码单元作为解码器实现摘要的并行输出；在编解码器间引入语义感知模块，通过计算标准摘要和生成摘要之间的语义相关性，促使模型生成语义完整的内容。仿真结果表明该模型能有效提高摘要的可读性和语义准确性。

2. 预训练语言模型

语言模型能在海量语料的预训练中学习到通用的语言表示，带来更强的泛化性能并加快目标任务的收敛速度。BERT 采用双向 Transformer 编码结构，共 12 层。其中 Transformer 编码单元包含两个子层：多头自注意力机制层(MultiHead Attention)和全连接前馈神经网络层(Feed Forward Neural Network)，每个子层中都加入了残差连接[11]和层归一化操作(Layer normalization, LN)。以 x 作为输入，编码单元输出可写为：

$$SubLayer\_Output = LN\left(x + \left(SubLayer\left(x\right)\right)\right) \tag{1}$$

全连接前馈神经网络包含两层全连接神经网络 *FFN*，激活函数是 *ReLU*，提供非线性变换。

$$FFN(x)=\max(0,xW_1+b_1)W_2+b_2 \tag{2}$$

多头自注意力机制 *MultiHead* 使每个词在编码时可以关注输入文本中的其他单词，它将输入向量 x 经过 h 组不同的线性变换矩阵 W_i^Q,W_i^K,W_i^V 映射产生查询向量 Q、键向量 K 和值向量 V；其中 i 表示第 i 组线性变换矩阵；通过键向量维度 d_k 的缩放计算出合适范围的注意力，最后将不同的注意力结果拼接起来与权重矩阵 W^O 相乘作为多头注意力机制层的输出。

$$MultiHead(Q,K,V)=Concat(head_1,\cdots,head_h)W^O \tag{3}$$

$$head_i=Attention(xW_i^Q,xW_i^K,xW_i^V)=softmax\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V \tag{4}$$

3. 融合分词和语义感知的中文文本摘要模型

本文提出一种融合分词和语义感知的中文文本摘要模型，该模型由三部分组成：1) 融合分词嵌入的 BERT 模型作为编码器；2) 多层 Transformer 解码单元组成的解码器；3) 语义感知模块。其中编码器分别对源文本 *src* 和标准摘要 *tgt**进行编码，产生相应的语义向量；解码器根据注意力机制配合源文本语义向量生成摘要；语义感知模块用于评估标准摘要与生成摘要的之间的语义相关性，鼓励模型生成高语义相关的摘要。模型的整体结构如图 1 所示。

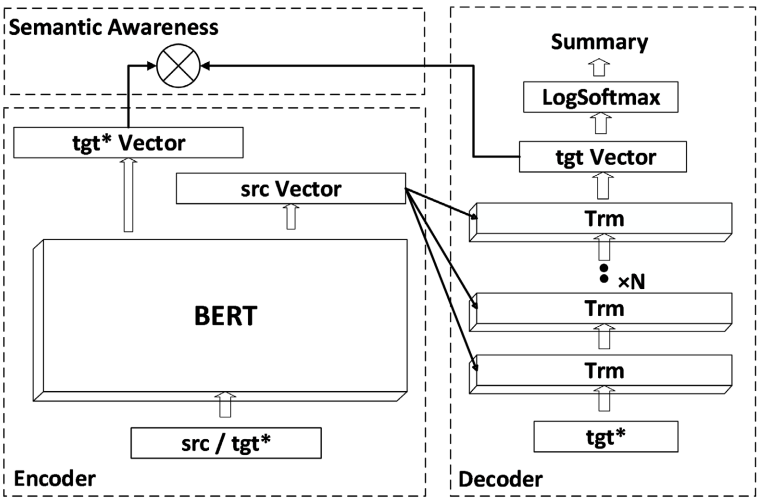

Figure 1. Overall structure of model
图 1. 模型整体结构

3.1. 编码器

编码器使用 BERT 预训练语言模型，它能较好的解决长文本上下文依赖问题，极大程度地获取语义信息。Google 发布的 BERT 模型使用 WordPiece 分词器，通过减小输入颗粒度实现缩减字典的目标。然而中文文本中多以词组为基本单位进行语义表达，部分词组样例在中文分词器和 WordPiece 分词器下对比，如表 1 所示。

可见以字为颗粒度作为输入必然会导致部分词组的语义信息丢失。为缓解这一问题，提出添加中文分词嵌入，使编码器学习到词组信息。

Table 1. Word segmentation comparison
表 1. 分词器对比

源词组	中文分词器	WordPiece
“油价”	“油价”	“油” “价”
“0.5%”	“0.5%”	“0” “.” “5” “%”
“元宵节”	“元宵节”	“元” “宵” “节”
“6834 人”	“6834” “人”	“68” “#34” “人”

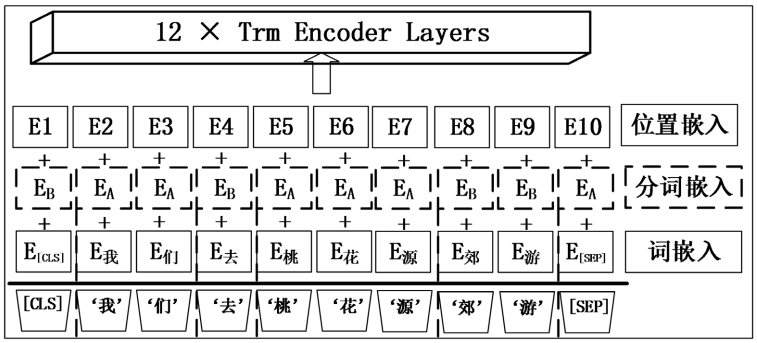

Figure 2. Add word segmentation embedding
图 2. 添加分词嵌入

如图 2 所示，为有效表示文本序列的开始和结束，在文本序列头部和尾部分别添加标签[CLS]和[SEP]。图中虚线为中文分词分割边界，交替使用 E_A 和 E_B 为各对词组分配分词嵌入编码，取值 0 和 1。生成分词嵌入编码需使用中文分词器，但是它与 WordPiece 分词器的分割逻辑不同。为实现两个分词器的编码对齐，设计以下处理流程：

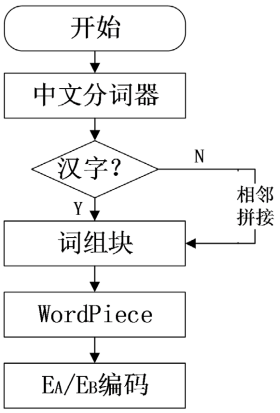

Figure 3. Word segmentation code alignment
图 3. 分词器编码对齐

如图 3，首先使用中文分词器分词获得词组块；将相邻的非汉字部分拼接成词组块；再输入 WordPiece 分词器分词；最后根据分词数量确定本词组块中 E_A 或 E_B 的编码数量。现编码器嵌入阶段可分成三个阶段：词嵌入(WordPrice Embedding, WE)、分词嵌入(Word Segmentation Embedding, WSE)和位置嵌入(Position Embedding, PE)。其中词嵌入和位置嵌入使用与 BERT 相同的方法。以文本序列 x 作为输入，嵌入阶段过程如下：

$$E_x = WE(x) + WSE(x) + PE(x) \tag{5}$$

将嵌入阶段的输出向量 E_x 作为后续双向 Transformer 编码单元的输入，在内部经过多头注意力机制层 *MHAtt*、全链接层 *FFN*、层归一化 *LN* 以及残差连接等一系列处理。令初始隐藏状态 $h_0 = E_x$，h^l 表示第 l 层输出的隐藏状态。以最后一层编码单元的隐藏状态作为 BERT 最终输出的语义向量 $T_x = h^{last}$。

$$\tilde{h}^l = LN\left(h^{l-1} + MHAtt\left(h^{l-1}\right)\right) \tag{6}$$

$$h^l = LN\left(\tilde{h}^l + FFN\left(\tilde{h}^l\right)\right) \tag{7}$$

3.2. 解码器

解码器是利用编码器产生的语义向量，根据自注意力机制，生成简要的摘要序列。解码器由两部分组成：多层 Transformer 解码单元和全连接网络。在训练过程使用 Teacher Forcing [12]训练方法，即每个时间步都使用标准摘要(Ground Truth)作为输入，可使训练过程更快收敛。Transformer 解码单元比编码单元多一个编解码注意力层 *EDMHAtt*，该层利用编码器输出的语义向量 T_x 完成对源文本语义的捕捉，则需在公式(6)、(7)之间添加以下过程：

$$\tilde{h}^l = LN\left(h^{l-1} + EDMHAtt\left(h^{l-1}, T_x\right)\right) \tag{8}$$

最后一层解码单元的隐藏状态 h^{end} 作为解码输出的特征向量，然后送入到全连接网络，W_0 和 b_0 是需要学习的参数，激活函数为 *Logsoftmax*。将特征向量映射成概率值，对应字典 V 中的每一个字，选择概率最高的词作为当前输出。

$$Y_i = Logsoftmax\left(W_0 h_i^{end} + b_0\right) \tag{9}$$

3.3. 语义感知模块

在模型训练过程中，解码过程的每个时间步以最大化似然估计作为训练目标。用 y_t^* 表示标准摘要中第 t 个单词，T_x 表示编码器输出的语义向量，最大化似然估计等价于最小化下面的损失函数：

$$Loss_{ml} = -\sum_{t=1}^{n} \log p\left(y_t \mid y_1^*, y_2^*, \cdots, y_{t-1}^*, T_x\right) \tag{10}$$

该损失函数要求模型生成的摘要与标准摘要对应位置的词尽可能相同。但是时常会出现生成摘要与标准摘要字面上相似，但表达不通顺、语义相关性较低的问题。为了解决这个问题，提出在编解码器间加入语义感知模块，如图 1 左上角模块。训练过程中编码器既要对文本 *src* 编码，也要对标准摘要 *tgt**进行编码。后者产生语义向量 V_{tgt^*} 过程中需固定归一化层参数、不进行随机舍弃神经元和参数梯度累积。解码器通过若干步解码后生成摘要序列 *tgt* 的特征向量 V_{tgt}。使用余弦相似度函数来评估两个摘要序列的语义相关性。则损失函数定义为：

$$Loss_{sim} = 1 - \cos\left(V_{tgt^*}, V_{tgt}\right) = 1 - \frac{V_{tgt^*} \cdot V_{tgt}}{\left\|V_{tgt^*}\right\| \left\|V_{tgt}\right\|} \tag{11}$$

将生成摘要内容的可靠性和可读性作为最终的训练目标，本模型设计混合目标损失函数：

$$Loss_{mixed} = (1-\lambda) Loss_{ml} + \lambda Loss_{sim} \tag{12}$$

其中，λ 是一个比例因子，用于表示 $Loss_{ml}$ 和 $Loss_{sim}$ 之间的幅度差异。此外，为提高模型的泛化能力，防止发生过拟合，在训练过程加入标签平滑技术(Label Smoothing) [13]。在测试过程为了扩大搜索空间，

使用集束搜索(Beam Search) [14]算法。

4. 实验

4.1. 数据集及预处理

实验仿真基于两个摘要数据集：大规模中文短文本摘要数据集(LCSTS) [15]和中国科学文献数据(CSL)。前者采集于新浪微博，内容涉及多个领域，每个样本包含标题和正文分别对应摘要和源文本，该数据集在检验集和测试集上进行 1~5 分的手工评分，分数越高源文本与摘要相关性越高。CSL 数据集选取计算机相关领域论文作为样本数据，以论文摘要作为源文本，标题作为摘要。如表 2 所示。

Table 2. LCSTS and CSL datasets
表 2. LCSTS 和 CSL 数据集

数据集		规模/对	源文本－摘要 平均长度/词
LCSTS	训练集	2,400,591	103.7/17.9
	检验集	10,666	107.8/18.0
	测试集	1106	108.1/18.7
CSL	训练集	3200	200.0/18.2
	测试集	300	200.1/18.0

数据集预处理包括下列步骤：去除 emoji 表情符号及特殊符号；使用符号“-”拼接连续的英文单词；将文本中英文标点符号转成对应的中文符号；以中文标点符号“，”、“。”、“！”、“？”等进行分句；根据 BERT 最大输入长度 512 进行裁剪，舍弃少于 3 个字符的句子等。使用结巴中文分词工具完成中文分词操作。

4.2. 评价指标

本文使用内部评价指标 ROUGE [16]。下列实验将使用以下几种常用方法。

1) ROUGE-N：

$$ROUGE\text{-}N = \frac{\sum_{S\in\{\text{Reference}\}}\sum_{gram_n\in S} Count_{match}\left(gram_n\right)}{\sum_{S\in\{\text{Reference}\}}\sum_{gram_n\in S} Count\left(gram_n\right)} \tag{13}$$

其中 $N=1,2,3$，分子表示生成摘要与标准摘要共同出现的 n-gram 个数，分母表示标准摘要的 n-gram 总个数。

2) ROUGE-L：

$$R_{lcs} = \frac{LCS\left(X,Y\right)}{m} \tag{14}$$

$$P_{lcs} = \frac{LCS\left(X,Y\right)}{n} \tag{15}$$

$$F_{lcs} = \frac{\left(1+\beta^2\right)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2 P_{lcs}} \tag{16}$$

其中 $LCS(X, Y)$表示标准摘要 X 与生成摘要 Y 的最长公共子序列长度；m 和 n 分别表示标准摘要和生成摘

要的长度；R_{lcs} 和 P_{lcs} 分别表示召回率和准确率；参数 β 一般设为 ∞。F_{lcs} 即所求的 ROUGE-L。

下列仿真实验将使用 ROUGE-1/2 评价生成摘要的信息丰富性，ROUGE-L 评价摘要的流畅性。

4.3. 实验与结果分析

4.3.1. 实验参数

本文模型使用 PyTorch 深度学习框架，在 NVIDIA RTX 2060 SUPER GPU 上进行实验。编码器使用 BERT-wwm-ext 预训练语言模型；解码器使用 6 层 Transformer 解码单元，其中隐藏层大小为 768，多头注意力个数为 8，前向神经网络大小为 2048，各子层 Dropout 为 0.1。因为编码器已进行预训练，而解码器采用随机初始化，为使编码器和解码器训练更加平稳，采用 BERTabs 相同的方法，分别为编解码器设置不同的 Adam 优化器及学习率。其中标签平滑因子为 0.1，束搜索宽度为 5，最短生成摘要长度为 14。

4.3.2. 比例因子 λ

为测试混合损失函数中比例因子 λ 对摘要质量的影响，设置以下实验。在完整 CSL 数据集上进行 K 折交叉验证，其中 K 取 10，Batch size 为 4，计算测试结果的平均值，结果如图 4。

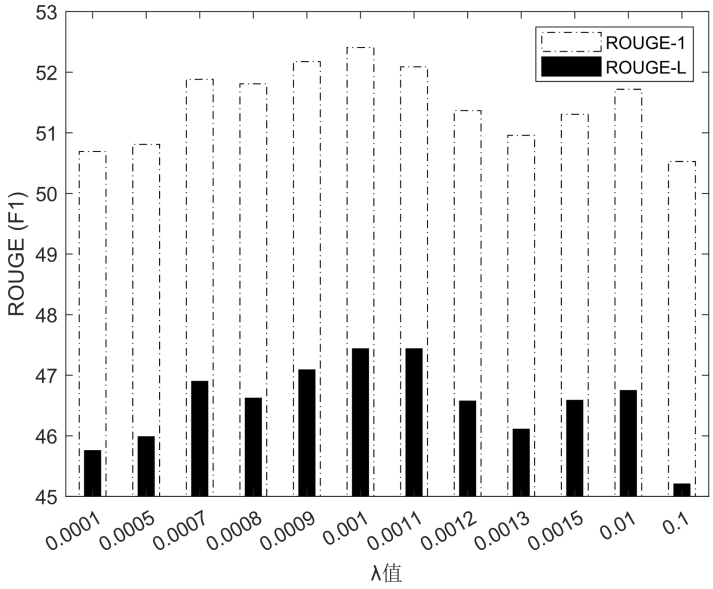

Figure 4. Results of different scaling factors λ
图 4. 不同比例因子 λ 的结果

图 4 中以 ROUGE-1 和 ROUGE-L 指标作为参考，在比例因子 λ 取 0.001 附近对文本摘要任务产生较好的效果。

4.3.3. 分词嵌入实验

由于 CSL 数据集专业术语较多，词组普遍较长。为验证分词嵌入对模型生成词组数量和准确率的影响，因此设置以下仿真实验。在 CSL 的训练集上迭代训练 15 次，Batch size = 4，λ = 0.001；实验组添加分词嵌入，对照组不添加；使用测试集进行测试，实验结果如表 3 所示。

其中词组占比表示生成摘要中词组的字数与总字数的比值；词组准确率表示正确出现在生成摘要中的词组数量与标准摘要词组总数量的比值。从表 3 可知，添加分词嵌入的实验组词组占比提高了 2.34%，且词组准确率提升了 1.69%，证明添加分词嵌入能使模型产生更多高质量的词组块。样例如表 4 所示。

Table 3. The statistical results
表 3. 统计结果

组别	词组占比	词组准确率
对照组	86.84%	45.61%
实验组	89.18%	47.30%

Table 4. Example of segmentation embedding experiment
表 4. 分词嵌入实验样例

源文本	网络编码是 2000 年提出的一种新算法，其主要优点是使组播传输速率能达到理论上限值。介绍了传统组播路由算法的局限性，分析了现有网络编码算法的优点和不足，在某个改进的网络编码数学模型上，提出了一种静态分布式分层网络编码 SDLNC 算法(Static Distributed Layered Network Coding)。模拟实验表明，该算法可以显著提高组播路由的数据传输速率。
标准摘要	一种基于网络编码的组播路由算法
对照组	一种基于动态分层的组网络编码算法
实验组	一种基于分布式网络编码的组播路由算法
标准摘要分词	“一种”“基于”“网络”“编码”“的”“组播”“路由”“算法”
对照组分词	“一种”“基于”“动态”“分层”“的”“组”“网络”“编码”“算法”
实验组分词	“一种”“基于”“分布式”“网络”“编码”“的”“组播”“路由”“算法”

4.3.4. 模型对比实验

为验证本文模型的综合性能，从 LCSTS 数据集随机选择近 60 万条样本参与训练(占总样本量的 1/4)，Batch size = 8，λ = 0.001，每 2 步进行一次梯度更新，共计训练 40 万步。检验与测试选用评分 3~5 分的文本－摘要对。每 2500 步保存一个 Checkpoint 并在检验集上验证，最后选择效果最好的 5 个 Checkpoint 在测试集上测试并计算平均值作为最终结果。

选择以下对比模型参与评估：

- RNN-context：LCSTS 数据集论文模型，以循环神经网络为编码器，组合所有编码器输出的隐状态作为解码器的输入。
- CopyNet [17]：通过指针网络直接从输入中复制词组作为输出。
- Global Encoding [18]：编解码器之间加入卷积门控单元执行全局编码，以提升词和全文之间的联系，解决词重复和语义无关的问题。
- BERTabs：使用 BERT-base，Chinese 为编码器，6 层 Transformer 为解码器。
- ProphetNet-Zh [19]：每个时间步基于上下文编码同时预测未来的 n 个输出，防止模型对强局部相关过拟合。

Table 5. LCSTS dataset experimental results
表 5. LCSTS 数据集实验结果

Model	R-1	R-2	R-L
RNN-context	29.900	17.400	27.200
CopyNet	34.400	21.600	31.300
Global Encoding	39.400	26.900	36.500
BERTabs	40.841	27.065	36.787
ProphetNet-Zh	**42.320**	27.330	37.080
Ours Model	42.251	**28.497**	**38.559**

表 5 模型均以词为颗粒度作为输入，数值为 ROUGE 评价体系中的 F1。本文模型在 ROUGE-2 和 ROUGE-L 指标上均有较好的效果，很好地验证了本模型在产生精确的中文词组和文本连贯性方面带来积极影响。

其中模型 ProphetNet-Zh 在完整 LCSTS 数据集上训练 30,000 GPU hours；本文模型使用通用语料训练的 BERT-wwm-ext 作为编码器，以生成式摘要为下游任务微调，消耗 40 GPU hours；对比发现，本文模型在训练语料量及训练时长均小于 ProphetNet-Zh 的情况下，仍能在 Rouge-2 和 Rouge-L 上实现领先，体现了本模型在文本摘要任务中的高效性和泛化能力。与 BERTabs 模型相比，在 Rouge-1、Rouge-2 和 Rouge-L 上分别相对提高了 3.45%、5.29%、4.82%。从表 6 样例可知，BERTabs 产生的摘要存在表达不连贯和部分语义缺失的问题，本文模型及 ProphetNet-Zh 产生的摘要能较为连贯且全面地总结文本信息，能基本达到文本摘要要求。

Table 6. Experimental sample on LCSTS dataset
表 6. LCSTS 数据集实验样例

源文本	即将于明年元旦起实施的《上海市停车场(库)管理办法》中，明确了本市将推行“错时停车”制度。记者日前走访后发现，本市的错时停车需求确实较为旺盛。然而，在该制度的推行过程中，却仍面临着超时停放、收费不一、业主反对等诸多问题。
参考摘要	申城实施错时停车面临诸多问题
BERTabs	上海错时停车收费不一等问题待解
ProphetNet-Zh	上海实施错时停车仍面临诸多问题
Ours Model	上海错时停车制度面临诸多问题

4.3.5. 消融实验

为验证本文模型不同组件对文本摘要任务的贡献度，本实验采用与实验 4.3.4 相同的参数和数据集，通过逐一添加组件进行对比实验，在测试集上实验结果如表 7 所示。

Table 7. Add different component results
表 7. 添加不同组件结果

Model	R-1	R-2	R-L
BERT-wwm-ext	41.163	27.594	37.712
+WS	41.473	27.909	37.719
+SA	41.748	27.976	38.217
+WS+SA	**42.251**	**28.497**	**38.559**

表 7 数据可知，单独添加分词嵌入(WS)或单独添加语义感知(SA)模块都能一定程度地提升文本摘要的质量，将两个组件融合能发挥出更好的效果，使得模型在 Rouge-1、Rouge-2 和 Rouge-L 上比基准模型相对提高 2.64%、3.27%、2.25%。图 5 展示表 7 中四个模型训练过程中，在检验集上关于 ROUGE-L 的训练曲线。

5. 结束语

在文本摘要的研究中，针对中文摘要生成过程中词组搭配不当、语义表达偏差等问题，本文提出了一种融合分词和语义感知的中文文本摘要模型；提出在预训练语言模型中强化中文分词的先验知识；配合编解码器间的语义感知评估，使得模型产生更多合理词组并关注整体语义的契合度，有效提高摘要的

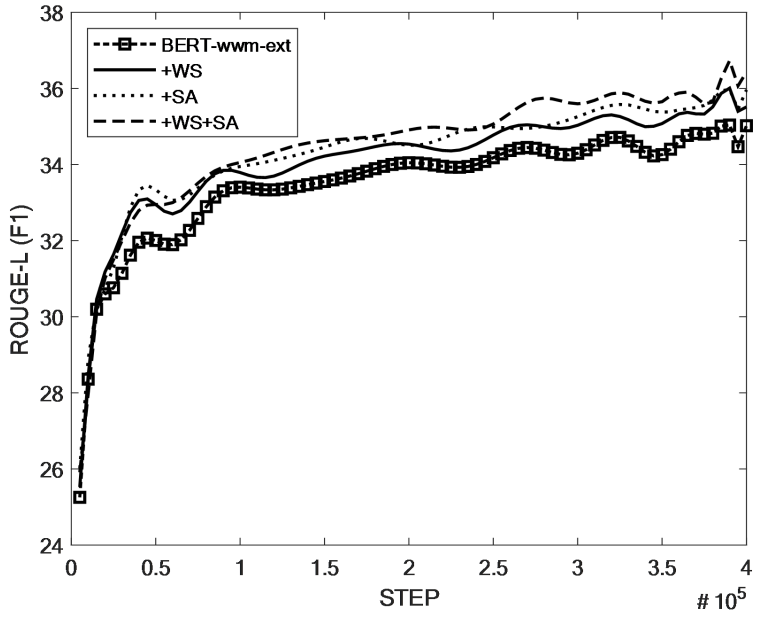

Figure 5. Rouge-L training curve
图 5. ROUGE-L 训练曲线

语言流畅性和语义完整性。使用 ROUGE 评价体系，分别在 LCSTS 和 CSL 数据集上的仿真实验结果表明，本文模型能有效提高文本摘要的质量。

参考文献

[1] Rush, A.M., Chopra, S. and Weston, J. (2015) A Neural Attention Model for Abstractive Sentence Summarization. *Proceedings of the* 2015 *Conference on Empirical Methods in Natural Language Processing*, Lisbon, September 2015, 379-389. https://doi.org/10.18653/v1/D15-1044

[2] 倪海清, 刘丹, 史梦雨. 基于语义感知的中文短文本摘要生成模型[J]. 计算机科学, 2020, 47(6): 74-78.

[3] Ma, S., Sun, X., Xu, J., *et al*. (2017) Improving Semantic Relevance for Sequence-to-Sequence Learning of Chinese Social Media Text Summarization. *Proceedings of the* 55*th Annual Meeting of the Association for Computational Linguistics*, Vol. 2, 635-640. https://doi.org/10.18653/v1/P17-2100

[4] Devlin, J., Chang, M.W., Lee, K., *et al*. (2018) BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding.

[5] Vaswani, A., Shazeer, N., Parmar, N., *et al*. (2017) Attention Is All You Need. *Annual Conference on Neural Information Processing Systems* 2017, Long Beach, 4-9 December 2017, 5998-6008.

[6] Wang, Q., Liu, P., Zhu, Z., *et al*. (2019) A Text Abstraction Summary Model Based on BERT Word Embedding and Reinforcement Learning. *Applied Sciences*, **9**, 4701. https://doi.org/10.3390/app9214701

[7] Wei, R., Huang, H. and Gao, Y. (2019) Sharing Pre-Trained BERT Decoder for a Hybrid Summarization. In: *China National Conference on Chinese Computational Linguistics*, Springer, Cham, 169-180. https://doi.org/10.1007/978-3-030-32381-3_14

[8] Liu, Y. and Lapata, M. (2019) Text Summarization with Pretrained Encoders. *Proceedings of the* 2019 *Conference on Empirical Methods in Natural Language Processing and the* 9*th International Joint Conference on Natural Language Processing* (*EMNLP-IJCNLP*), Hong Kong, November 2019, 3730-3740. https://doi.org/10.18653/v1/D19-1387

[9] Cui, Y., Che, W., Liu, T., *et al*. (2019) Pre-Training with Whole Word Masking for Chinese BERT.

[10] Sun, Y., Wang, S., Li, Y., *et al*. (2019) Ernie: Enhanced Representation through Knowledge Integration.

[11] He, K., Zhang, X., Ren, S., *et al*. (2016) Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 770-778. https://doi.org/10.1109/CVPR.2016.90

[12] Williams, R.J. and Zipser, D. (1989) A Learning Algorithm for Continually Running Fully Recurrent Neural Networks.

Neural Computation, **1**, 270-280. https://doi.org/10.1162/neco.1989.1.2.270

[13] He, T., Zhang, Z., Zhang, H., *et al*. (2019) Bag of Tricks for Image Classification with Convolutional Neural Networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, 16-17 June 2019, 558-567. https://doi.org/10.1109/CVPR.2019.00065

[14] Wu, Y., Schuster, M., Chen, Z., *et al*. (2016) Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation.

[15] Hu, B., Chen, Q. and Zhu, F. (2015) LCSTS: A Large Scale Chinese Short Text Summarization Dataset. *Proceedings of the* 2015 *Conference on Empirical Methods in Natural Language Processing*, Lisbon, September 2015, 1967-1972. https://doi.org/10.18653/v1/D15-1229

[16] Lin, C.Y. (2004) Rouge: A Package for Automatic Evaluation of Summaries. *Workshop on Text Summarization Branches Out*, Barcelona, 25-26 July 2004, 74-81.

[17] Gu, J., Lu, Z., Li, H., *et al*. (2016) Incorporating Copying Mechanism in Sequence-to-Sequence Learning. *Proceedings of the* 54*th Annual Meeting of the Association for Computational Linguistics*, Vol. 1, 1631-1640. https://doi.org/10.18653/v1/P16-1154

[18] Lin, J., Sun, X., Ma, S., *et al*. (2018) Global Encoding for Abstractive Summarization. *Proceedings of the* 56*th Annual Meeting of the Association for Computational Linguistics*, Vol. 2, 163-169. https://doi.org/10.18653/v1/P18-2027

[19] Qi, W., Gong, Y., Yan, Y., *et al*. (2021) ProphetNet-X: Large-Scale Pre-Training Models for English, Chinese, Multi-Lingual, Dialog, and Code Generation. *Proceedings of the* 59*th Annual Meeting of the Association for Computational Linguistics and the* 11*th International Joint Conference on Natural Language Processing*: *System Demonstrations*, 1-6 August 2021, 232-239. https://doi.org/10.18653/v1/2021.acl-demo.28