Computer Science and Application 计算机科学与应用, 2021, 11(10), 2546-2554
Published Online October 2021 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2021.1110258

Hans汉斯

基于柔性粒度的文本摘要自动化技术创新研究

涂著刚，李正军，杨　敏

贵阳高新数通信息有限公司，贵州　贵阳

收稿日期：2021年9月20日；录用日期：2021年10月17日；发布日期：2021年10月25日

摘　要

本文对使用序列到序列模型进行文本摘要时的方法进行研究，重点分析了集外词难以生成以及单词间联系缓慢两个不足产生的原因；结合字节对编码算法，提出了柔性粒度字节对编码算法FG-BPE。改进后的FG-BPE算法将完整单词分割为不相交的子词单元，通过降低文本粒度大小解决缓解集外词难以生成的问题，通过子词单元二次分割实现单词之间联系的更好学习。关于Gigaword集的实验证明，与原始子词分割算法相比，FG-BPE实现了一元组、二元组及最长公共子串的共现召回率整体提升。

关键词

文本摘要自动化，子词，字节对编码，粒度

Research on Innovation of Text Summarization Automation Technology Based on Flexible Granularity

Zhugang Tu, Zhengjun Li, Min Yang

Guiyang HiTech Data Communication Co., Ltd., Guiyang Guizhou

Received: Sep. 20th, 2021; accepted: Oct. 17th, 2021; published: Oct. 25th, 2021

Abstract

In this paper, the method of text summarization using sequence-to-sequence model is studied, and the causes of two shortcomings, which are difficult to generate extra words and slow connection between words, are emphatically analyzed. Combined with byte pair coding algorithm, a flexible

文章引用: 涂著刚, 李正军, 杨敏. 基于柔性粒度的文本摘要自动化技术创新研究[J]. 计算机科学与应用, 2021, 11(10): 2546-2554. DOI: 10.12677/csa.2021.1110258

granularity byte pair coding algorithm FG-BPE is proposed. The improved FG-BPE algorithm divides the whole word into disjoint sub-word units, solves the problem that it is difficult to generate words outside the set by reducing the text granularity, and realizes better learning of the relationship between words through the secondary segmentation of sub-word units. Experiments on Gigaword set show that compared with the original sub-word segmentation algorithm, FG-BPE can improve the recall rate of co-occurrence of one tuple, two tuples and the longest common substring as a whole.

Keywords

Text Automation, Sub Words, Encoding Byte Pairs, Granularity

1. 引言

新一代泛在网的高速发展促使信息数据呈现爆炸式发展，截至 2021 年 6 月，中国网民规模达 10.11 亿，较 2020 年 12 月增长 2175 万，互联网普及率达 71.6%；在各类信息中，最常用的文本数据类型呈指数级增长，因此如何快速准确地定位目标文本数据，成为现阶段持续研究的热点和难点[1]。

目前主要采用搜索引擎解决上述问题，但搜索引擎被检出的原始信息普遍存在冗余化、离散化、偏离化等不足，通常需要用户进行二次确认导致效率低下。因此，采用简短概要信息对文本数据进行描述的方法被提出，使得用户通过摘要实现对原文信息的传递领会，实现信息定位的效率性和准确度提升。但具体使用过程中，海量的文本数据规模致使常需采用人工方式进行文本摘要结果的二次核准，导致耗费巨大、实时性差、描述失真等问题不断出现，甚至出现刻意夸大扭曲事实的情况。

为解决人工生成摘要的局限性，自动文本摘要(Automatic Text Summarization, ATS)技术应运而生[2]。传统的专家系统自动文本摘要方法，根据既定规则截取原文片段组成摘要，难以适应不同领域的文本摘要，规则化的结果往往易出现关键信息缺失、信息冗余、语义错误等弊端，无法满足实际使用需求。新一代的自动文本摘要技术融合深度机器学习技术，以标准篇或多篇文档为准则，实现文本关键信息摘要的自动、快速、准确、实时地生成保存，并保证摘要语义通顺、简洁准确，可有效弥补人工摘要的不足。

随着数据量和数据类别的不断丰富，以及计算能力的大幅提升，以神经网络为代表的深度机器学习技术在自然语言处理、图形图像处理等领域得到显著而有效地应用。与传统专家系统的规则文本自动摘取技术不同，基于深度学习的文本摘要可实现原始文本特征的自动提取，通过端到端学习，极大简化自动文本摘要模型的复杂性，并随着运用次数和场景的增加，可持续实现精确度、准确度的提升，较基于规则的方法更具前景。现阶段，基于深度学习的文本自动摘要技术的难点集中在集合外词语难以生成、摘要与原始语义不匹配等方面。

针对当今的文本摘要模型难以生成集外词以及缺乏对单词之间的联系进行有效建模的问题，本文提出了一种基于改进子词单元的生成式文本摘要模型。主要创新点集中在使用改进的子词分割算法将一个完整的单词分割成不相交的子词单元，实现同一含义但不同形态单词之间的联系加强，例如受单复数、时态影响的单词，有助于模型对单词之间的联系进行建模。同时，通过粒度更小的子词单元构成集外词，从而更好地体现单词之间的联系，缓解集外词难以生成的问题。

2. 自动文本摘要的形式化定义

自动文本摘要技术始于 20 世纪 50 年代的 Luhn 等人[3]，给定输入文本 $X=(x_1,x_2,\cdots,x_m),x_i\in\theta_s$，其中 θ_s 为输入词汇表，m 为输入文本长度，x_i 为第 i 个输入文本单词。自动文本摘要系统根据输入文本生成摘要 $Y=(y_1,y_2,\cdots,y_m),y_i\in\theta_t$，其中 θ_t 为输出词汇表、n 为生成摘要长度，y_i 为第 i 个生成摘要的单词，生成的摘要应该满足 $n\ll m$ 条件。

发展至今主要分为 4 类，根据输入文档数量的不同可分为单文档摘要和多文档摘要，根据输入文本长短的不同可分为短文本摘要和长文本摘要，根据输入输出语言的异同可分为单语言摘要、多语言摘要和跨语言摘要，根据输入文本领域的不同可分为新闻摘要、法律摘要等[4]。本文从生成摘要中的句子或单词是否完全来自于输入文本的角度出发，将自动文本摘要分为抽取式(Extractive)文本摘要、生成式(Abstractive)文本摘要，如图 1 所示。

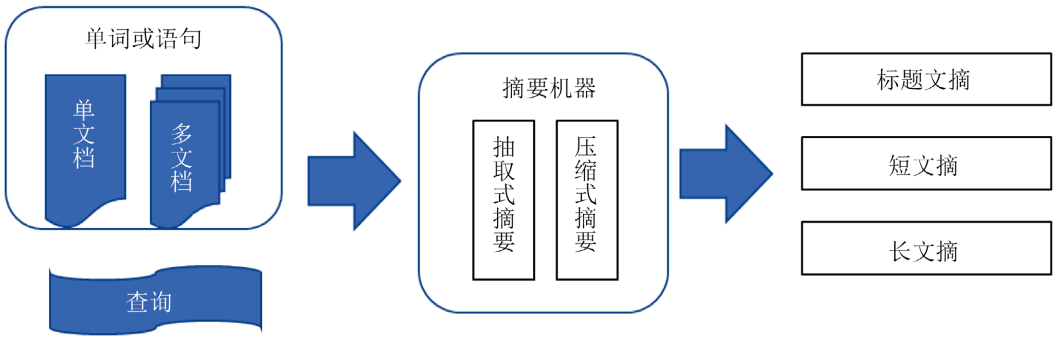

Figure 1. Automatic text abstract classification model diagram
图 1. 自动文本摘要分类模型图

2.1. 抽取式文本摘要技术

抽取式文本摘要(Extractive Text Summarization, ETS)从输入文本中抽取出特定词句组成最终摘要，生成摘要中的每个单词 y_i 都属于输入文本中，即 $y_i\in X$ 。ETS 技术包含输入文本的词句重要度得分计算、基于得分的句子筛选及组成两个过程，包括基于统计特征的方法(Statistical-Based)、基于主题的方法(Topic-Based)、基于图的方法(Graph-Based)和基于神经网络的方法(Neural Network-Based)四种[5]。

1) 基于统计特征技术(Latent Statistical Indexing)结合句子的统计特征来评估相关重要程度，然后根据重要程度对句子进行选择后组成摘要，在研究初期得到了广泛使用；

2) 基于主题的方法通过隐含语义索引(Latent Semantic Indexing)、概率潜在语义分析(Probabilistic Latent Semantic Analysis)和隐含狄利克雷分布(Latent Dirichlet Allocate)等模型计算文档的主题分布，然后根据计算得到的主题分布对文档中句子的重要程度进行评估并加以选择；

3) 基于图的方法将输入文本作为一张图处理，将原文句子作为节点，句子间的关联作为节点边，其中最为具代表的为 Text Rank 算法[6]。

4) 基于神经网络的方法作为最近的研究热点，将神经网络引入到抽取式文本摘要当中，取得良好效果[7]。

2.2. 生成式文本摘要技术

生成式文本摘要(Abstractive Text Summarization, ATS)中存单词 y_i 未在输入文本中出现过，ATS 首先构建模型对原始文本进行理解，然后以模拟人类的方式对原文进行概括生成摘要。与抽取式摘要不同，

生成式方法生成的摘要中的词语或者句子不必完全来自于原文，较 ETS 具备更大的灵活性，模型输出的摘要也更像人工生成的摘要。但是，由于生成式文本摘要中包含文本理解、句子改写、同义词替换等底层自然语言处理，这导致相比较于抽取式文本摘要模型，生成式文本摘要模型的训练要更加困难。

近年来，随着大规模文本摘要数据集的出现以及算力的提升，人们逐渐将研究的重点放在生成式文本摘要上[8]。按照方法的不同，ATS 技术分为基于图的方法(Graph-Based)、基于模板的方法(Template-Based)和基于神经网络的方法(Neural Network-Based)三种。

3. 自动文本摘要不足分析

自动文本摘要在序列模型和注意力机制得到大规模应用后获得长足发展，但依旧存在以下三方面主要问题：

1) 生成式文本摘要模型难以生成低频词、罕见词等集外词(Out of Vocabulary, OOV)。在进行文本摘要之前首先需要构建词汇表白名单，生成摘要中的单词均来自于该词汇表。词汇表容量大小限制并不会收入所有词语，导致诸如地名、人名、民族语言等低频词不在词汇表中，在外的这部分单词为集外词。在摘要生成时，集外词会被映射为未知，从而导致模型泛化能力的降低，显著影响生成摘要的可读性。

2) 生成式文本摘要模型难以在单词间建立有效的模式联系。以英文为例，时态、语态、词态、复数等因素导致大量的派生词以及前缀和后缀出现，如果以完整单词作为输入输出模型，会难以在单词之间的联系建立可靠联系，严重影响摘要表现。例如“think”、“thinks”、“thought”、“thinking”等，还存在表示词性的前缀或后缀等。

3) 生成式文本摘要模型难以实现对关键信息建模的清晰和充分抽取。绝大部分的生成式文本摘要大都基于序列模型构建并采取端到端训练，训练过程中的编码器会同时对关键信息和噪声进行编码，现有的序列到序列模型缺乏对输入文本噪声的过滤建模过程。

4. 基于子词单元的改进

4.1. 基于概率和粒度的融合

针对难以生成集外词、缺乏单词间联系的两个问题，现有的解决方法可以分为两类：

1) 综合考虑词汇表中单词概率分布的生成概率和输入文本中单词的拷贝概率两个因素，模型根据拷贝概率从输入文本中直接拷贝集外词到输出摘要当中，代表性包括指针生成器(Pointer-generator)模型、拷贝网络(CopyNet)等。拷贝概率的引入可以适当缓解集外词难以生成的问题，但依旧不能对单词间联系进行建模。

2) 降低输入输出文本中单词的粒度，例如将单词转为不相交的子词单元，并通过细粒度的子词单元表示集外词。此外，由于不同单词之间可能会共享某一个子词单元，所以将单词分割为子词单元能够使模型更好地对单词与单词之间的联系进行建模。代表算法为字节对编码算法(Byte Pair Encoding, BPE)，BPE 将单词将输入输出文本中的单词分割为不相交的子词单元，但由于子词分割是在有限语料上训练得到，在训练语料规模较小情况下的子词分割方式未必是最优解、甚至是不恰当的分割，影响单词联系和模型性能。

4.2. 基于子词单元的序列到序列文摘模型

基于 Transformer 的编码器－解码器框架构建模型整体结构，通过编码器对输入文本编码获得符合深度模式处理的文本表示，通过解码器对文本表示进行解码，输出生成摘要，如图 2 所示。

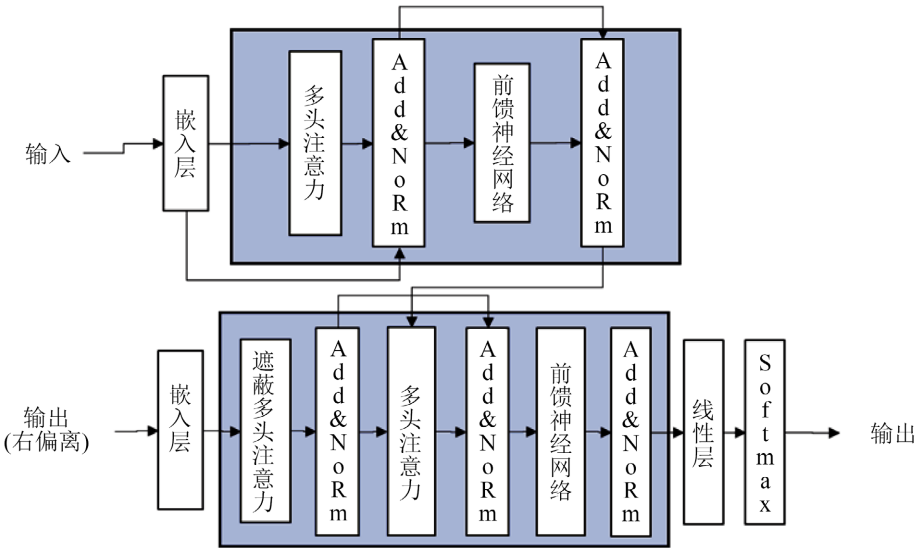

Figure 2. Frame diagram of encoder-decoder based on Transformer
图 2. 基于 Transformer 的编码器－解码器框架图

4.2.1. 编码器

输入文本 $X=(x_1,x_2,\cdots,x_m)$ 模型编码器由 N 个相同的编码层堆叠构成，第 i 层编码过程如公式(1)、(2)、(3)所示：

$$Z_i^l = SelfAttention\left(h_i^{l-1}\right) \tag{1}$$

$$o_i^l = LayerNorm\left(h_i^{l-1} + z_i^l\right) \tag{2}$$

$$h_i^l = LayerNorm\left(o_i^l + FeedForwardNetwork\left(o_i^l\right)\right) \tag{3}$$

其中，h_i^{l-1}表示编码器第$l-1$层对于输入文本 x 中第 i 个单词 x_i 的编码，第$l-1$层的输出为第l层的输入。*SelfAttention*() 表示对输入应用自注意力(Self-Attention)机制，*LayerNorm* 表示层标准化，*FeedForwardNetwork* 表示前馈神经网络(Feed Forward Network)，Z_i^l 和 O_i^l 为计算过程的中间结果。

自注意力机制是注意力机制的变体。注意力机制包含查询矩阵 *Query*、键矩阵 *Key* 和值矩阵 *Value* 输入参数，根据查询矩阵和键矩阵，计算注意力分布 α，如公式(4)所示：

$$\alpha = Likelihood\left(Query, Key\right) \tag{4}$$

其中 *Likelihood* 表示兼容函数，其目的是计算查询矩阵 Q 和键矩阵 K 之间的相似度，也就是注意力分布 α。通常，*Likelihood* 有加性(Addictive)、权值映射(General)和点积(Dot-Product)三种方式[9]，如公式(5)所示：

$$Likelihood\left(\mathcal{Q},K\right)=\begin{cases} v_a^{\mathrm{T}}\tanh\left(W_a\left[\mathcal{Q};K\right]\right), & \text{Addictive} \\ \mathcal{Q}W_aK^{\mathrm{T}}, & \text{General} \\ \mathcal{Q}K^{\mathrm{T}}, & \text{Dot-Product} \end{cases} \tag{5}$$

其中，v_α 和 W_α 为训练参数，tanh 为激活函数。将注意力分布 α 和值矩阵相乘，得到基于注意力的文本表示，如公式(6)、(7)、(8)所示：

$$Attention\left(\mathcal{Q},K,V\right)=f\left(\mathcal{Q},K\right)V=\alpha V=\frac{\mathcal{Q}K^{\mathrm{T}}}{\sqrt{d}}V \tag{6}$$

$$Q = W_q H \tag{7}$$

$$K = W_k H \tag{8}$$

其中，d 为矩阵 Q 维度，Q 为解码器前一时刻的输出，而 K 为编码器对输入文本的编码表示。多头注意力(Multi-Head Attention)机制将 H 进行多次线性变换得到多组 Q 和 K，并将自注意力机制计算得到的结果连接起来进行线性变换，如公式(9)所示：

$$Multi\text{-}Head\left(Q,K,V\right) = \left[head_1; head_2; \cdots; head_{hn}\right]W \tag{9}$$

W 为可训练参数，hn 为注意力头个数，$head_i$ 为 $Attention\left(QW_i, KW_i, VW_i\right)$。多头注意力能够从不同的角度对文本进行编码，获得更好的文本表示。

4.2.2. 解码器

解码器对编码器输出 H 进行解码生成摘要。解码器为 N 层，且每层除多头注意力模块和前馈神经网络模块外，还有一个解码器注意力模块 *EncDecAttn()*用于联合编码器和解码器。*EncDecAtt()*中的 K 和 V 来自于编码器的输出 H，而 Q 来自于解码器第 l 层对输入的编码 S_l。

解码器最后一层的输出经线性变换后，通过 *Softmax*()函数得到词汇表上的概率分布 P_{vocab}，通过概率分布 P_{vocab} 得到当前步的输出单词，如公式(10)所示：

$$P_{vocab} = Softmax\left(W_0 S + b_0\right) \tag{10}$$

4.3. 文本的子词单元表示

4.3.1. 字节对编码算法

字节对编码算法包含子词词汇表构建和子词分割两个步骤，其中子词词汇表构建算法基于频率统计，包含训练语料 D 和合并次数 n 两个参数。其生成过程是，将训练语料 D 中的所有不重复字符加入到词汇表 V 中构建初始词汇表，将每个字符为一个单独的子词单元，合并频率最高的两个相邻子词单元并加入词汇表。循环 n 次算法结束时，子词词汇表构建完成。子词词汇表记录了子词的分割方式，可通过子词词汇表对训练样本中的单词进行分割。

本文使用算法 1 来对原始文本进行子词分割。如算法 1 所示，子词分割算法输入原始文本 D 和子词词汇表 V，返回经过子词分割的文本 D'。

算法 1 子词分割算法

Step1：读取原始文本 D 并进行分词；

Step2：将子词词汇表 V 根据子词长度从长到短排序；

Step3：for 原始文本 D 中的每个单词 do；

Step4：for 子词词汇表中每个子词单元 do；

Step5：尝试将单词中的子串替换为当前的子词单元，重复；

Step6：重复第 4 步，最终生成经过子词分割的文本 D'；

通过上述算法构建子词词汇表，可以得到假设训练语料中包含 m 个字符，子词分割过程共包含 n 次合并操作，则最终的词汇表大小介于 $m + n$ 到 $2m + n$ 之间。

4.3.2. 改进的字节对编码算法

由于训练语料是有限的，将当前语料中频率最高的子词单元对合并可能并不是最优的选择，其原因在于子词分割方式过拟合于训练语料，当训练语料规模较小时，该问题会更容易发生。为了缓解这个问题，本文在子词词汇表的构建过程中添加噪声扰动，提出细粒度字节对编码算法 FG-BPE (Fine Grained

Byte Pair Encoding)。通过选择概率和频率高的子词单元对进行合并，实现对原始文本分割的更加合理的结果，如算法 2 所示。

算法 2 细粒度字节对编码算法 FG-BPE

Step1：读取训练语料 D 并进行分词；

Step2：在每个单词后加符号</w>表示单词结束，统计单词频次；

Step3：将每个单词分割为字符序列，此时子词单元为单个字符，初始的词汇表 V 由单个字符组成；

Step4：while 当前合并次数小于 n do；

Step5：统计语料 D 中相邻的两个单元对出现的频次；

Step6：设置当前待合并的单位对为频率最高的单元对；

Step7 ::：if random()<p then；

Step8：设置当前待合并的单元对为频率次高的单元对；

Step9：重复第 7 步，直到结束；

Step10：将当前待合并单元对中的两个子词单元合并为一个单元，并加入到词汇表 V 中；

Step11：重复第 10 步，直到结束。

算法 2 中通过概率 p 与频率次高的子词单元融合考虑进行合并。

5. 实验

5.1. 训练目标

模型的训练目标是在给定输入文本 x 以及模型参数 θ 的情况下最大化生成每个目标单词的概率，其等价于最小化模型生成的单词和目标单词之间的负对数似然(negative log-likelihood)，如公式(11)所示：

$$\mathcal{L}(\theta)=\frac{1}{|D|}\sum_{(x,y)\in D}\log p(y\mid x;\theta) \tag{11}$$

其中，D 为训练数据集，x 为输入文本，y 为目标摘要，θ 为模型的参数。

5.2. 评价规则

针对生成的文本评价规则，包括人工和自动两种，其中人工评价方法从是否易读、句子之间是否连贯、是否存在语法错误、摘要中的指代是否明确、摘要是否覆盖了原始文本中的关键内容、是否存在冗余的问题六方面衡量摘要质量；人工评价方法会引入评价人的主观倾向，当数据量较大时不太可行。为了解决人工评价摘要缺点，很多自动评价摘要的方法出现，其中具有代表性的有 ROUGE (Recall-Oriented Understudy for Gisting Evaluation)和 meteor，本文采用 ROUGE 规则进行评判。

ROUGE 是 Lin [10]基于召回率提出的文本摘要自动评价指标，该方法通过计算参考摘要(Reference Summaries)和摘要间基本单元(n 元组)的重叠程度来衡量模型摘要质量。ROUGE 通常包含以下几种指标：

- ROUGE_N：该指标统计参考摘要和模型生成摘要 n 元组之间的共现召回率，如公式(12)所示：

$$\text{ROUGE\_N}=\frac{\sum_{S\in\text{Reference Summaries}}\sum_{gram_n\in S}Count_{match}(gram_n)}{\sum_{S\in\text{Reference Summaries}}\sum_{gram_n\in S}Count(gram_n)} \tag{12}$$

其中 S 代表了参考摘要中的句子，$gram_n$ 代表 n 元组，$Count(gram_n)$ 表示 s 中 n 元组的数量，$Count_{match}(gram_n)$ 表示模型生成的摘要和参考摘要匹配的 n 元组数量。

- ROUGE_S：该指标与 ROUGE-2 类似，区别在于 ROUGE_2 二元组单词必须相邻，而 ROUGE_S 中

的二元组不一定相邻。

- ROUGE_L：该指标根据模型生成的摘要和参考摘要间的最长公共子串(Longest Common Subsequence, *LCS*)衡量模型生成摘要的质量，如公式(13)所示：

$$\begin{aligned}&\text{ROUGE\_L}=\frac{\left(1+\beta^2\right)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2P_{lcs}}\\&\text{where }R_{lcs}=\frac{LCS\left(X,Y\right)}{m},P_{lcs}=\frac{LCS\left(X,Y\right)}{n},\beta=\frac{P_{lcs}}{R_{lcs}}\end{aligned}\tag{13}$$

其中，X 为参考摘要、m 为参考摘要长度、Y 为模型生成摘要、n 为生成摘要长度。

5.3. 实验集及效果

本文使用公开的自动文本摘要数据集 Gigaword [11]进行实验，Gigaword 作为大规模的自动文本摘要数据集，由 400 万条模型输入的新闻以及模型输出的标题摘要组成。本文将 1,198,668 条样本划分为训练集，将 2,397,335 条样本划分为验证集，将 399,556 条样本划分为测试集，输入平均长度为 32 个字符，输出摘要平均长度为 9 个字符。

在 Gigaword 数据集上各模型的实验结果表明，使用原始子词分割算法，模型取得了 37.71 (ROUGE_1)、18.43 (ROUGE_2)和 34.87 (ROUGE_L)的结果；使用改进子词分割算法，模型取得了 37.92 (ROUGE_1)、18.94 (ROUGE_2)和 35.05 (ROUGE_L)的结果。分别在 ROUGE_1 上提升了 0.21、在 ROUGE_2 上提升了 0.51、在 ROUGE_L 上提升了 0.18。

实验结果证明，本文提出的柔性粒度字节对编码算法 FG-BPE 通过使用改进的子词分割算法，在保证同一含义的前提下，将完整单词分割成保持不同形态间联系的子词单元，实现面向单词间联系的有效建模系，模型性能得到有效提升；在此基础上，通过粒度更小的子词单元构成集外词，有效缓解了集外词难以生成的问题。

下一步研究在于，如何跳出两段式噪声过滤步骤，设置全局向量实现噪声过滤算创新法，避免计算过程中的信息损失。

基金项目

《基于自然语言处理技术的招投标公共服务平台研究与应用推广》，2020 年度贵阳市国家创新城市“百城百园”行动项目，贵阳市科技局(筑科项目[2020] 22 号)。

参考文献

[1] 张敏, 刘建华, 谢靖. 网络科技信息监测中富文档识别与信息提取技术研究[J]. 情报科学, 2017(1): 128-132.

[2] 唐晓波, 顾娜, 谭明亮. 基于句子主题发现的中文多文档自动摘要研究[J]. 情报科学, 2020(3): 11-16.

[3] 刘志明, 于波, 欧阳纯萍, 等. 基于主题的 SE-TextRank 情感摘要方法[J]. 情报工程, 2017(3): 97-104.

[4] 罗毅辉, 熊曙初. 一种集成框架下的分布式多文档自动摘要方法[J]. 情报杂志, 2013(11): 133-136.

[5] 马骏. 自动文本摘要技术的关键问题研究[D]: [硕士学位论文]. 成都: 电子科技大学, 2020.

[6] 黄波, 刘传才. 基于加权 TextRank 的中文自动文本摘要[J]. 计算机应用研究, 2020, 37(2): 407-410.

[7] 邹蕾, 崔斌, 樊超, 孙豫峰. 基于双向编码文本摘要-长短期记忆-注意力的检察建议文本自动生成模型[J]. 科学技术与工程, 2021, 21(25): 10780-10788.

[8] 王凯祥. 面向查询的自动文本摘要技术研究综述[J]. 计算机科学, 2018, 45(S2): 12-16.

[9] 王晴. 基于统计的多文本网站文本内容抽取算法[J]. 安徽电子信息职业技术学院学报, 2021, 20(4): 6-12.

[10] 孙宝山，谭浩. 基于 ALBERT-UniLM 模型的文本自动摘要技术研究[J/OL]. 计算机工程与应用: 1-8. http://kns.cnki.net/kcms/detail/11.2127.TP.20210802.0922.002.html, 2021-08-02.

[11] 侯圣峦，张书涵，费超群. 文本摘要常用数据集和方法研究综述[J]. 中文信息学报, 2019, 33(5): 1-16.