计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
2025,34(2):135−144 [doi: 10.15888/j.cnki.csa.009764] [CSTR: 32024.14.csa.009764]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于编码器-解码器架构大语言模型的关键句抽取①

彭俊峰, 俞　凯, 李国靖

(杭州师范大学 信息科学与技术学院, 杭州 311121)
通信作者: 俞　凯, E-mail: yk@hznu.edu.cn

摘　要: 关键句抽取技术是指利用人工智能, 自动从一段长文本中寻找核心句. 该技术可用于信息检索的预处理, 对文本分类、抽取式摘要等下游任务有着重要意义. 传统的无监督关键句抽取技术多数基于统计学以及图模型的方法, 存在着精度不高以及需要提前建立大规模语料库等问题. 本文提出了一种中文环境下的无监督抽取关键句方法 T5KSEChinese, 该方法利用编码器-解码器架构, 通过输入和输出提示词来忽略目标句与原文长度不匹配的问题, 以得到更准确的结果. 同时, 本文提出一种对比学习正样本构造方式, 并将该方式结合对比学习来对模型编码器部分进行半监督训练, 提升下游任务效果. 本研究使用轻量化的模型, 在无监督下游任务中得分优于参数量大于自身数十倍的大语言模型, 最终实验结果证明了提出方法的准确度和可靠性.

关键词: 关键句抽取; 生成式预训练语言模型; 对比学习; 正样本; 编码器-解码器

引用格式: 彭俊峰,俞凯,李国靖.基于编码器-解码器架构大语言模型的关键句抽取.计算机系统应用,2025,34(2):135–144. http://www.c-s-a.org.cn/1003-3254/9764.html

Key Sentence Extraction Based on Encoder-decoder Architecture Large Language Model

PENG Jun-Feng, YU Kai, LI Guo-Jing

(School of Information Science and Technology, Hangzhou Normal University, Hangzhou 311121 China)

Abstract: Key sentence extraction technology refers to using artificial intelligence to automatically find key sentences from a long text. This technology can be used for preprocessing information retrieval and is of great significance for downstream tasks such as text classification and extractive summarization. Traditional unsupervised key sentence extraction technologies are mostly based on statistics and graphical model methods, which have problems such as low accuracy and the need to build a large-scale corpus in advance. This study proposes T5KSEChinese, a method that can extract key sentences without supervision in the Chinese context. This method uses an encoder-decoder architecture to ignore the mismatch in length between the target sentence and the original text by inputting and outputting prompt words to obtain more accurate results. At the same time, a contrastive learning positive sample construction method is also proposed and combined with contrastive learning to conduct semi-supervised training on the encoder part of the model, which can improve the performance of downstream tasks. The method uses lightweight models to outperform the large language model with tens of times the number of parameters in the unsupervised downstream task. The final experimental results prove the accuracy and reliability of the proposed method.

Key words: key sentence extraction (KSE); generative pre-training language model; contrastive learning; positive sample; encoder-decoder

① 收稿时间: 2024-07-16; 修改时间: 2024-08-13; 采用时间: 2024-08-27; csa 在线出版时间: 2024-12-19
CNKI 网络首发时间: 2024-12-20

关键句抽取任务是一种从给定的文本语料库中抽取重要句子的技术, 该技术使用句子中包含的特征来识别和抽取文本中最重要的句子. 这一技术适用于许多场景, 例如自动摘要、中心思想抽取、提升 Rerank 精确度[1]等. 例如:

(1) 在文本摘要领域. 在期刊、报纸的文档中, 摘要总结会放在头版位置, 读者可通过该信息了解文章是符合自身阅读需求. Sheng 等[1]提出了 TextRank 方法, 该方法对原有的 PageRank 算法进行改进, 将句子作为图网络的节点, 识别文中最重要的句子, 构建抽取式摘要.

(2) 在文本向量匹配领域. 搜索引擎通过较少的关键字符 (例如关键词) 来搜索某篇文章时, 由于关键词和文档之间的向量长度差距过大, 很有可能导致搜索结果出现很大偏差; 在关键字符和关键句之间的长度差距小, 同时关键句能够涵盖文章大部分主题信息时, 可以提高搜索结果的精准度. 在检索增强生成 (retrieval-augmented generation, RAG)[2]任务中的 Rerank 环节, 需要搜索与问题相关的文章以辅助内容生成, 可通过关键句抽取技术提升匹配精度.

本文提出了一种名为 T5KSEChinese 的无监督方法, 通过提示词对编码器-解码器架构的生成式模型进行提示得到初始得分, 再通过句子自身的信息进行加权得分计算, 分数最高的句子被视为原文的关键句. 该方法可忽略关键句和文档自身之间的长度差距, 能够在无监督环境中得出文档关键句, 并保证准确性. 其次, 本文提出了一种半监督正样本构造方式对模型进行对比学习训练, 训练后模型相对于原本无监督方法效果有进一步提升.

1 相关工作

1.1 关键句抽取

关键句抽取 (key sentence extraction, KSE) 是自然语言处理 (natural language processing, NLP) 学习中重要任务之一, 目的是从文本中抽取出能够代表文本核心思想的句子. 这一技术涵盖许多应用场景, 例如抽取式摘要. 抽取式摘要任务的目的是从文档中选取关键词、关键句, 以组成摘要. 抽取式摘要在语法、句法级别上错误率低, 在准确性要求较高的场景中表现出色, 如新闻报道摘要、法律文书摘要等.

传统的无监督关键句抽取技术主要基于统计学以及图模型等方法. TextRank[3]将文档中的句子比做无向图的节点, 将句子间的相似性作为边的权重构成图, 再通过 top-K 确定关键句. Berkhin 等[4]将 PageRank 算法利用到句子颗粒度上, 将关键句作为节点, 根据句子的相关性和多样性进行句子得分计算, 得出每个句子的重要性. Padmakumar 等[5]使用聚类的方式, 将文章中的句子使用 Skip-thought[6]方法进行编码, 得到向量表示, 再通过聚类得到多个类别, 最后从每个类别中, 选择距离质心最近的句子, 将多个句子组合作为最终摘要.

近年来, BERT 等预训练语言模型在 NLP 的各个领域任务取得了较好的成果. 这些预训练语言模型能够产生准确的语义嵌入向量, 更好地代表句子. 其中, Liu[7]将抽取式摘要任务视为序列标注任务, 利用 BERT 的语义表达能力, 将文章中关键的句子标注为 1, 其余为 0, 提出了 BERTSUM 模型. Liu 等[8]将 BERT 的编码器进行微调, 得到能够适用于文档级别的语义抽取任务的模型, 将该成果用于抽取式摘要任务, 提出 PreSumm 模型. Gokhan 等[9]提出一种特征融合的抽取式摘要模型, 该方法利用句子长度、句子中的名词数目、数词数目、首字母是否大写等特征, 结合 Reimers 等[10]提出的 Sentence-BERT 模型, 得出句子相似度, 并构建图模型, 从而抽取关键句.

传统的图模型方法以及多特征融合方法, 存在指代错误以及精确度不高等问题. 使用预训练语言模型进行关键句抽取的方法, 由于目标句和文档之间的长度差异较大, 在进行语义嵌入向量处理时出现了准确度表现不理想的问题. 同时这一类方法需要使用人工进行关键句的标注, 过程费时费力. 信息化时代中, 这一缺点正被逐渐放大. 此外, 当更强大的语言模型出现时, 上述方法会阻碍新模型的快速应用. 本文使用编码器解码器架构的模型, 通过双提示词来弥补目标句和原文档之间长度不匹配带来的问题, 同时本方法在不需要额外预训练的情况下, 能够快速应用于同结构的其他语言模型, 以达到更好的效果.

1.2 Encoder-decoder 架构的生成式大语言模型

2017 年, 谷歌发布了 Transformer[11]架构, 该架构基于自注意力机制的神经网络, 能够更好地捕捉长文本中的信息. OpenAI 基于 Transformer 架构发布了仅解码器 (decoder-only) 架构的生成式语言模型 GPT-1 (generative pre-trained Transformer, GPT)[12], 为后续的 GPT 系列奠定了架构基础. Cao 等[13]于 2022 年提出了

基于编码器-解码器架构的生成式语言模型 T5, 该模型的最大版本具备 110 亿参数量, 是早期的大语言模型之一, 能够通过提示词来完成翻译、分类等任务. Lewis 等[14]提出了基于编码器-解码器架构的 BART 模型, 该模型在解码器中使用双向自注意力机制, 可以同时捕捉到文本序列中的前向和后向依赖关系. Chung 等[15]对 T5 模型进行改进, 通过指令微调 (instruction fine-tuning) 的形式提高模型的泛化能力, 并提出 Flan-T5 模型. Tay 等[16]提出了 UL2 模型, 通过混合去噪器 (mixture-of-denoisers, MoD) 进行预训练, 该模型在实验中超过了 GPT-3[17]模型和 T5 模型.

目前, 仅解码器架构的大语言模型已成为主流, 被广泛应用于聊天对话、智能问答以及其他领域, 但仍未有关于该类模型在关键句抽取领域的探讨. 本文在第 3 节对目前主流的大语言模型进行了关键句抽取的对比实验, 结果表明, 本文提出的方法达到较好的效果, 并发现仅解码器大语言模型在关键句抽取任务中存在的缺点. 同时, 本文使用的模型参数量及运行要求远少于当前的大语言模型, 能够在实际环境中进行大规模部署和应用.

1.3 对比学习

对比学习 (contrastive learning) 早先用于计算机视觉领域[18], 通过数据增强等方式得到与原始锚样本相似的正样本, 以及与原始锚样本不同的负样本. 在得到样本的嵌入向量后, 在向量空间中通过拉近锚样本与正样本的距离、增大锚样本与负样本的距离, 从而得到准确度高的图片向量表示. 近年来, NLP 领域的研究者参考对比学习的核心思想, 得到了许多重要研究成果. Gao 等[19]在得到原始锚样本向量表示后, 通过对原始锚样本向量进行随机失活 (dropout) 获得正样本, 同一批次中其他样本向量作为负样本, 提出了 SimCSE 模型, SimCSE 在许多任务中取得了领先地位. Wu 等[20]指出 SimCSE 的正样本构造方式导致句子长度并不会变化, 模型会将具有相同长度的句子生成相似度偏高的嵌入向量.并提出了 ESimCSE, 通过随机叠词的方式生成具有不同长度的正样本, 在下游任务得分上超过了 SimCSE. Ni 等[21]将对比学习和 T5 的编码器部分结合, 提出了 Sentence T5, 在下游任务中超过了 Sentence-BERT 等基线模型. Sentence T5 仅包含了编码器部分, 并不具备自然语言生成能力, 据我们了解, 这是第 1 份将对比学习与编码器-解码器生成式模型进行结合的工作.

受到 PromptRank[22]的启发, 该方法可利用 T5、BART 等生成式语言模型, 通过使用提示词的方式提示模型生成关键词, 本文针对上述问题结合生成式语言模型提出一种基于提示词的无监督抽取式摘要方法; 并在后续用本文提出的构造正样本方式进行半监督学习, 在对比试验中达到了 SOTA (state of the art) 效果.

2 方法

本文在关键句抽取领域提出了两种方法, 无监督的关键句抽取和半监督的关键句抽取. 无监督的方法在未训练的前提下进行关键句抽取, 该方法利用 T5 模型的生成能力, 对原文中所有句子进行遍历, 预测其成为关键句的可能, 再将预测得分最高的前几名句子排为关键句. 半监督的方法需要提前进行训练, 该方法不需要具备强标签原文关键句的训练数据, 而是通过对比学习的方式增强模型的语义, 提高模型在下游任务中的表现.

2.1 T5 模型

本文选择 T5 模型作为基线模型, 主要原因有以下 3 点.

1) 在长文本中抽取关键句会出现目标句和原文长度差异大的问题, 导致在进行关键句选择时精度不高, 这一问题可通过利用编码器-解码器架构的输入提示词和输出提示词来弥补.

2) Raffel 等[23]的研究表明, 使用编码器-解码器架构的模型在处理摘要等需要深入理解输入内容并生成相关响应的任务中, 比使用仅解码器的模型更具优势.

3) T5 模型是由 Google 提出的近年来第 1 个在参数量上突破百亿的生成式大语言模型, 具有里程碑式的意义及影响力. 近年来编码器-解码器架构的大语言模型涌现了具备代表性的 UL-2、Flan-T5 等. 以 T5 模型作为基线能帮助本文方法在应用中易于实现, 便于方法在模型之间迁移应用.

综上所述, 本文考虑通过编码器-解码器架构中 T5 模型作为基线模型进行实验.

2.2 方法

2.2.1 对比学习

对比学习原用于计算机视觉领域, 核心思想是使用样本增强等方式得到正负样本, 通过令原本的锚样本在向量空间中靠近正样本、远离负样本, 使模型能

够获得更好的鲁棒性, 提升特征向量表达能力. 近年来, NLP 的无监督学习领域中有许多使用对比学习的研究成果, 如 SimCSE、ESimCSE 等. 截至目前, 据本文了解, 仍未有关于对比学习是否能够改善模型生成能力的探讨. 为进一步推动这一领域发展研究, 为后续研究者提供支持, 本文使用基线模型 T5 Chinese 的编码器部分进行了对比学习, 期望能帮助生成式模型更好的理解原文信息, 提升模型生成效果, 下游任务上的实验结果证实了这一观点.

已有的对比学习方法大多通过对单词的随机替换、叠词、随机失活等方式进行正样本构建, 例如 SimCSE 等, 而在关键句抽取领域中, 锚样本往往较长, 使用上述方式在文字数量及语义上均可能产生较大影响, 导致模型在下游任务上表现不佳. 为在训练时最大程度保留长文本的语义, 同时结合关键句抽取这一下游任务进行训练, 本文提出一种对比学习正样本构造方式. 并在第 3.5 节中, 通过对比实验验证了提出方法在下游任务上的有效性.

2.2.2 样本处理

本文方法的处理流程可概括如下: 在得到包含段落原文和人工总结的样本后, 将样本的 short_text 进行句子划分, 并进行随机删减, 再将该样本的总结添加至内容中. 正样本的构造包含以下 3 步.

(1) 使用 5 个中文标点符号[，。！？；]对 short_text 进行句子划分, 并将划分后的多个句子组合成一个句子列表. 划分规则参考 BERTSUM.

(2) 进行判断: 如果第 1 步中句子列表长度大于 3, 便从句子列表中随机移除一个句子; 对于某些特殊样本, 即原文中没有出现上述常规标点符号的样本, 以及划分后句子列表长度小于 3 的样本, 则认为其包含信息较少, 不适用于训练, 将跳过该样本. 最后, 将处理后的句子列表重新拼接组合, 得到替换后句子列表.

(3) 得到替换后句子列表后, 将该样本对应的人工总结拼接至替换后句子列表前, 再将句子列表重组成段落, 得到针对原文的正样本. 在重组时, 将保留句子之间原本的标点.

正样本的构造示例如图 1 所示. 对于负样本的设置, 我们参考了 SimCSE 等无监督方法, 在每个训练批次中, 选取与当前处理样本不同的其他样本作为负样本.

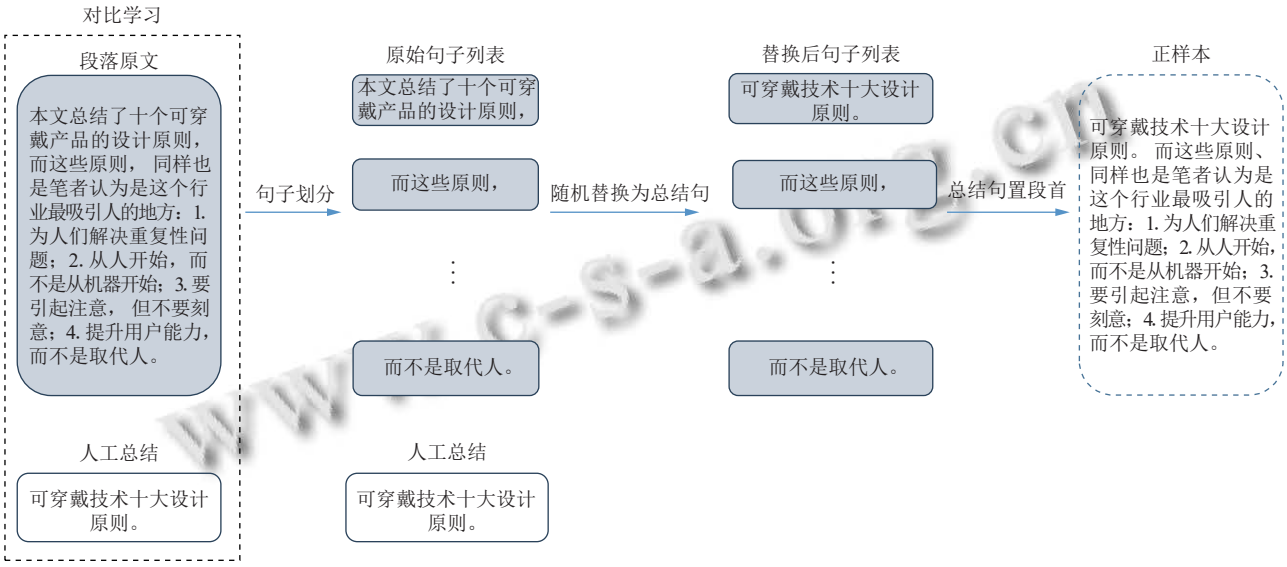

图 1 对比学习工作流程

2.2.3 模型训练

模型训练时, 本文将模型的随机失活设置为 0, 这一做法与 SimCSE 等的做法有所不同. 在 SimCSE 等对比学习方法中, 需采用随机失活用于为正样本与锚样本之间添加噪声, 从而增强模型的语义表达能力. 然而, 在本文方法中, 正样本结合了人工标签, 它们在语义层面能够更准确地体现原文所要表达的内容含义. 因此, 本文认为随机失活在该方法中并不必要, 并将其设置为 0.

在后续消融实验中, 可发现当随机失活率设置为 0

时, 本文方法可以取得最好的效果. 本文中温度系数预设为 0.05, 使用无监督的交叉熵损失方法计算损失值.

训练时, 本文将预设的正样本与原始数据混合, 并将同一批次中其他句子作为负样本. 然后, 将所有样本输入到 T5 模型的解码器部分, 得到正负样本以及锚样本的特征向量. 之后将这 3 组特征向量进行组合输入模型, 通过交叉熵损失方法计算函数损失值. 最后将该损失值反向传播至 T5 模型编码器部分进行参数优化, 以提升模型语义特征表达能力. 本文仅对 T5 模型编码器部分的参数进行优化, 解码器部分参数保持不变, 旨在通过加强模型的特征抽取能力, 观察其在下游生成式任务中的表现. 训练部分的具体做法参考了 Sentence T5.

在半监督训练结束后, 将得到的模型进行保存, 用于进行关键句抽取的下游任务.

2.2.4 数据预处理

关键句抽取前, 需将待抽取关键句的原文通过提前声明好的句子划分规则进行句子划分, 得到多个候选句. 与训练阶段相同, 本文采用中文标点[，。！？；]进行句子划分. 划分后, 原文被分为多个候选句组成的列表; 与训练阶段有所不同的是, 本文将原文对应标签句添加至列表尾, 目的是用于模型进行后续关键句得分计算.

2.2.5 提示词构建

T5 模型具备通用模型能力, 通过不同提示词, T5 模型可处理多种 NLP 下游任务, 如翻译、问答等. 本文通过提示词使 T5 模型具备抽取关键句能力. T5 模型具备双向的编码器以及单向的解码器, 本文将提示词设置为编码器提示词以及解码器提示词, 模型理解原文语义, 输出准确信息.

本文将编码器提示词设置为“文章: ”, 解码器提示词设置为“这篇文章的关键句是: ”具体做法是, 将编码器提示词结合原文内容、解码器提示词结合候选句内容, 通过 Tokenizer 转换后, 映射为机器中的向量, 组成 *encoder_input_ids* 和 *decoder_input_ids*. 将 *encoder_input_ids* 和 *decoder_input_ids* 一并输入 T5 模型, T5 模型将根据已有的输入信息进行提示输出. 编码器提示词是为了帮助 T5 模型能够更准确地将原文信息转换为向量, 编码器提示词的设置影响了模型是否能够充分理解原文内容并根据该内容进行输出; 而解码器提示词决定了模型输出方式, T5 模型将根据已有的解码器提示词的提示进行后续内容生成.

2.2.6 得分计算

T5 模型作为生成式模型, 输出结果可以转化为每个位置上出现某个字词的概率. 具体来说, T5 模型会对输出结果进行归一化指数转换, 得到一个长度为 $seq\_len \times voc\_size$ 的向量. 其中, *seq_len* 代表 T5 模型最终输出结果中的字数, *voc_size* 代表 T5 模型所能够处理表达的词汇量. 向量中的每个元素表示某个字词在该位置出现的概率, 这一概率值在 0–1 之间. 对于生成式任务, T5 模型会选择概率最大的字词作为最终输出结果. 具体做法是找到向量中最大值的索引, 并将其转换为对应的字词. 该索引称为 *token_id*, 可以通过分词器进行转换.

本文并不将概率值转换为可读文本, 而是获取候选句出现在模型输出结果中的可能性. 具体来说, 我们首先忽略由解码器提示词生成的 k 个字, 其中 k 是解码器提示词经过分词后的长度. 然后, 将候选句转换为 *token_id* 列表, 并根据 *token_id* 列表反向得到候选句中每个字出现在模型输出结果中的可能性. 最后, 将每个字的出现可能性累加, 得到当前候选句作为关键句的可能性得分, 记为 *key_score*.

为了防止句子的长度在这一步骤中造成最终得分的偏差, 本文方法中引入长度系数 l. 得出 *key_score* 后, 将该得分除以该句子的 $\log(seq\_len, l)$ 值, 进而得到最终得分 *score*. 最终, 本文通过 top-K 算法, 将 *score* 分数最高的候选句选为该原文的关键句.

关键句得分计算流程可概括为以下公式:

$$P_{\text{all}} = Softmax(O_i) \tag{1}$$

$$w_j = Tokenize(S_i) \tag{2}$$

$$\text{Index}_j = \left\{Encode\left(w_j\right)\right\} \tag{3}$$

$$P_i = \frac{\left(\sum_{j=0}^{n-1} P_{\text{all}}\left[Index_j\right]\right)}{\log(seq\_len, l)} \tag{4}$$

其中, *tokenize* 代表将文字序列转换为 token (标记) 序列的计算过程, *Encode* 表示编码步骤.

3 实验

3.1 数据集

由于关键句抽取领域研究资料稀缺, 已经公开的关键句抽取数据集仍是空白, 从头标注数据需要耗费

大量的时间及精力. 本文选用与关键句抽取领域相似的抽取式摘要领域的中文数据集 LCSTS[24]. LCSTS 数据集取自新浪微博, 数据集包含 3 个部分, 第 1 部分为训练数据集, 包括 2 400 591 个样本数据, 每条样本数据由短文本及作者给出标题组成. 第 2 部分为验证数据集, 包括 10 666 个人工标注样本, 每条样本包含短文本、作者给出标题以及人工评分. 人工评分由专业人员手工标注, 对每个样本都进行了 1–5 分的评分, 用于评判短文本与新闻标题的相关程度; 其中 1 代表最不相关, 5 代表最相关. 第 3 部分为测试数据集, 包括了 1 106 条样本, 每条样本中同样包含人工评分信息. 图 2 是测试数据集中一则样本示例.

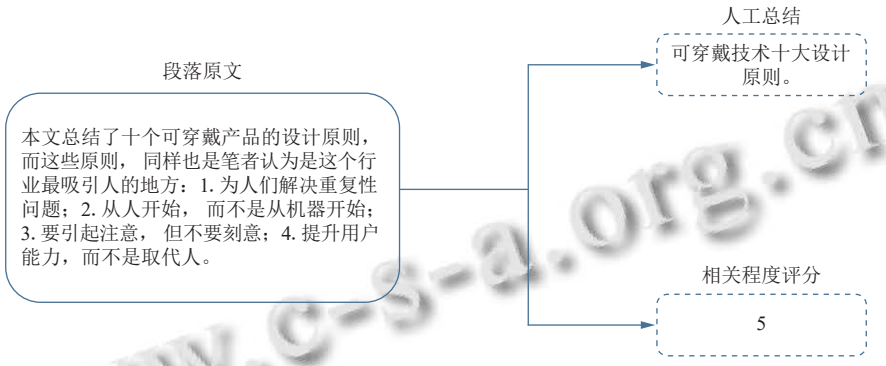

图 2 LCSTS 部分样本示例

3.2 对比方法

本文最终选用相关领域 4 个方法进行对比实验. 分别为 TextRank、GUSUM、BERTSUM、句嵌入向量方法. 由于现有许多方法均以英文文献为目标, 经过中文文献样本测试, 效果与英文存在一定偏差. 对于 TextRank 和 BERTSUM 方法, 采用目前可信度较高的中文版复现; 对于 GUSUM, 没有发现公开具有可信度的中文语言环境复现代码. GUSUM 采用了无监督的训练方式, 本文在进行实验时将 GUSUM 所使用的 BERT 替换为 BERT-Chinese, 并将分词等部分英文工具替换为中文 NLP 领域常用工具, 以便其能够适应中文语言环境.

3.2.1 TextRank

TextRank 算法是一种基于图模型的无监督关键句抽取方法, 其灵感来源于网页重要性排序算法 PageRank. PageRank 算法通过分析网页之间的链接关系构建网络, 并根据网页的链接数量和质量来计算网页的重要性. TextRank 算法则将 PageRank 算法中的网页替换为句子, 并将网页之间的链接关系替换为句子间的相似度. 如果两个句子有相似性, 则认为两个句子之间存在一条无向有权边, 并将图中权值较高的句子选为关键句.

3.2.2 GUSUM

GUSUM 是一种基于无监督学习的文档摘要抽取方法. 与传统的图模型抽取摘要方法相比, GUSUM 改进了句子中心性计算方式, 使用顶点来表示句子的特征得分; 利用 Sentence-BERT 获取句子的嵌入向量, 计算语义相似性. 通过上述改进, GUSUM 构建了一个包含句子重要性和相似性的无向图. 在摘要抽取过程中, GUSUM 能够在确保摘要包含最重要的句子的同时, 排除具有相似含义的句子, 提高输出结果质量.

3.2.3 BERTSUM

BERTSUM 将 BERT 预训练模型应用于抽取式文本摘要, 模型主要由句子编码层和摘要判断层组成; 句子编码层通过 BERT 模型获取文本中每个句子的句向量编码, 摘要判断层通过 3 种不同的结构进行选择判断, 为每个句子进行打分, 最终选取分数最大的前 3 个句子作为文本摘要.

复现时, 本文参考了网络上已公开的 BERTSUM-Chinese 代码, 在 LCSTS 数据集上选用了 20 万条样本进行微调. 同时, 本文分别使用 BERT-Base-Chinese 和 Sentence-BERT-Chinese 这两个不同的骨干模型进行实验.

3.2.4 句嵌入向量

本文使用了传统的关键句抽取算法进行对比实验, 将所有句子和原文分别进行向量化, 将所得向量放在同一嵌入空间中比较, 得到句子和原文的相似度, 相似度得分最高的前 3 名句子即作为候选关键句. 本文选用 BERT-Base-Chinese 作为基础模型实现这一方法.

3.3 评估工作

本文提出了一种关键句抽取任务的评估标准. 本文认为, 在一篇新闻中存在一个最关键的句子, 模型应

当能够在新闻的所有句子中给出标签句最高的得分. 实验中, 本文将 LCSTS 中的标签句当作最关键句, 使用 *P* (*Precision*), *R* (*Recall*), *F*1 (*F*1_*score*) 值来进行关键句的评判, 其中 *pred_keySentence_list* 为模型给出的 top-K 的候选关键句, *gold_label* 为数据集中的人工评审关键句. 评判指标算法如算法 1.

算法 1. 关键句评估标准

```
if (pred_keySentnce_list [0] == gold_label):
    TP += 1
    TN += 3–1
else:
    for i in topks.get_key_sentences(num=3):
        if i[index] == gold_label:
            break
        FP += 1
        FN += 1
Precision = TP/(TP + FP)
Recall = TP/(TP + FN)
F1_score = 2 × Precision × Recall/(Precision + Recall)
return Precision, Recall, F1_score
```

算法 1 中, *TP* 为 true positive, 计为模型识别出的关键句为 label 句, 即正样本被正确识别的数量; *FP* 为 false positive, 计为模型识别出关键句不为标签句, 即误报的负样本数量; *TN* 为 true negative, 即负样本被正确识别的数量; *FN* 为 false negative, 模型认为是非关键句, 但实际上是关键句, 即漏报的正样本数量.

模型输出的候选关键句存储在 *pred_keySentence_list* 中. 实验中, 该列表的长度被限制为 3. 对于每个样本, 模型对所有切分后的句子进行关键性评分, 并选择得分最高的 3 个句子进行评判. 此外, 如果切分后文章的句子数不足 3, 则认为该文章内容过少, 将其舍弃. 评判时, 有下列 3 种情况: (1) 模型将标签句放在 *pred_keySentence_list* 的第 1 个位置, 认为标签句在全文中占有最重要的关键句得分, 计 *TP*+=1, *TN*+=2; (2) 模型将标签句放在 *pred_keySentence_list* 的其他位置, 没有正确给标签句打出最高得分, 需要判断标签句出现在 *pred_keySentence_list* 的具体位置下标 *idx*, 计 *FP*+=*idx*, *FN*+=1; (3) *pred_keySentence_list* 并没有出现标签句, 计 *FP*+=3, *FN*+=1.

最终通过 *TP*、*FP*、*TN*、*FN*, 计算出 *P*, *R*, *F*1 得分, 并将该得分作为输出结果. *P*, *R*, *F*1 得分的计算公式参考式 (5)–式 (7).

$$P = \frac{TP}{TP + FP} \tag{5}$$

$$R = \frac{TP}{TP + FN} \tag{6}$$

$$F1 = 2\frac{P \times R}{P + R} \tag{7}$$

3.4 实验结果

实验结果如表 1 所示, 其中 CL 代表使用对比学习的方式进行半监督预训练, T5KSEChinese 代表本文中提出的方法. LCSTS 包含了 200 余万条中文新闻数据样本, 数据集后的括号数字代表了本文用于进行训练的样本规模: 实验结果中, 本文方法在实验中达到了新的 SOTA 效果. 未经过对比学习的 T5 模型能够达到 35.70% 的 *F*1 分数, 领先于无监督的 GUSUM 方法. 经过对比学习微调后, 本文方法达到了 59.17% 的 *F*1 分数. 这一实验结果从侧面印证了 encoder-decoder 架构中提升 encoder 部分的语义表达能力可以改善模型的生成能力, 改善最终生成效果.

表 1 模型在 LCSTS 数据集上的实验结果 (%)

方法	微调数据集	准确率	召回率	*F*1分数
TextRank	N/A	20.46	28.16	23.70
GUSUM	N/A	31.36	38.64	34.62
T5KSEChinese	N/A	**32.65**	**39.38**	**35.70**
BERTSUM + BERT-Base-Chinese + CL	LCSTS (20w)	40.91	51.82	45.71
BERTSUM + Sentence-BERT-Chinese + CL	LCSTS (20w)	44.51	54.44	48.98
BERT-Base-Chinese + CL	LCSTS (20w)	32.81	52.21	40.29
T5KSEChinese + CL	LCSTS (20w)	**55.81**	**62.97**	**59.17**

本文将 BERTSUM 分别应用于 BERT 和 Sentence-BERT 上. 实验表明, Sentence-BERT 在关键句抽取任务中表现优于 BERT 模型. 这验证了模型自身语言理解能力在相关下游任务中的重要性, 尤其在摘要等需要理解长文本语义的任务上. GUSUM 模型在实验结果中表现较差, 本文分析可能是数据的特异性导致, 即 LCSTS 数据集中关键句包含着较少 GUSUM 模型需要统计的特征, 如名词、数词等.

本文同时邀请了相关专业的 5 位评审员进行人工评判. 每位评审员需要对 15 条包含了原文与模型抽取

关键句的样本进行打分, 0 分代表不相关, 1 分代表部分相关, 2 分代表完全相关. 评分结果如表 2 所示. 平均分代表每个模型所得到的平均分, 最高得分数代表了最高分 2 分的个数, 最低得分数代表了模型得到的 0 分的个数.可看出, 本文方法在人工评审阶段保持了较好的成绩.

表 2 人工评判结果

方法	总分	平均分	最高得分数	最低得分数
BERTSUM	26	1.04	9	8
GUSUM	29	1.16	10	6
T5+Ours	**34**	**1.36**	**10**	**1**

3.5 对比学习

本文设置对比实验以验证文中提出的使用标题构建正样本方法的有效性, 对照方法我们选用 SimCSE、ESimCSE, 骨干模型均选用 T5 模型. 对比实验结果如表 3 所示.

表 3 使用不同对比学习方式所得到的结果 (%)

方法	微调数据集	准确率	召回率	*F*1分数
T5+SimCSE	LCSTS	35.03	44.23	39.10
T5+ESimCSE	LCSTS	35.01	45.06	39.40
T5+Ours	LCSTS	**55.81**	**62.97**	**59.17**

SimCSE 将特征向量进行随机失活来进行样本增强, 从而得到正样本; ESimCSE 在 SimCSE 的基础上采取随机叠词方式进行进一步的样本增强, 确保样本之间的长度有所区别, 以获得更好的鲁棒性, 避免模型倾向于判断相同或相似长度的句子在表达上更相近. 本文遵从原作者的步骤, 将 SimCSE 和 ESimCSE 的随机失活率设置为 0.3, 以便达到最好效果.

在关键句抽取实验中, 本文方法在指标上优于前两种方法, 取得了 SOTA 结果. 这进一步验证了利用标题构建正样本这一方法能够在不显著损失原文信息的情况下进行样本增强; 此外, 该构造方法能够随机改变文章长度, 避免模型将相同长度的语句错误地判断为正样本.

3.6 大语言模型

为了说明方法的有效性, 本文进一步与使用仅解码器的大语言模型进行对比实验. 对比模型选取为 GPT-3.5-Turbo[25]、Gemini[26]、Spark-General-3.5、GLM4[27]. GPT-3.5-Turbo 是 OpenAI 开发的大语言模型, 相对于原本的 GPT-3.5 进行了改进, 在保证准确率的同时提升了处理问题的效率和速度. Gemini 是谷歌 AI 推出的多模态人工智能模型, 能够处理文字、图像、音频、视频等信息, 在多项任务中达到了领先效果. Spark 是科大讯飞发布的一款多模态认知大语言模型, 该模型使用中文语料训练, 在文本生成、语言理解、知识问答等领域表现优异. GLM4 是智谱清言发布的一款大语言模型, 该模型在指令跟随能力、长文本能力等超越了 GPT-4, 并开源了 9B 版本.

其中 Spark-General-3.5 参数量为千亿级别; GPT-3.5-Turbo 参数量未公开, 作为参考; GPT-3 参数量为 175B, 而 T5-base 参数量约为 220M, 更小的参数量让本文的方法在低资源环境下的部署和应用具备更多优势.

本文参考官方提供的 API 示例进行模型调用, 对每个模型使用语义相似的提示词以及同样的样本进行生成内容提示, 以完成实验. 提示词构造如下: “将下面这段话通过‘，。！？；’5 个符号进行划分, 从而得到多个句子 sen_list. 并从 sen_list 中选出并回答最能代表原文语义的前 3 个句子, 使用换行符进行分割, 并按重要性进行排序. 我只需要和原文相关的 3 个句子, 句子应当完全来自原文, 不需要加上任何修饰. 回答时不要考虑句子的位置信息. 原文内容:”.

为了忽略关键句出现位置的影响, 实验将关键句随机放在原文中的不同位置, 实验结果如表 4 所示, 表格中的分数为实验得到的“最高分/中位数/最低分”评分. 可以观察到, 本文方法在无监督的情况下领先于已有的主流大语言模型.

表 4 主流大语言模型的实验结果 (%)

方法	准确率	召回率	*F*1分数
GPT-3.5-Turbo	21.7/20.0/19.0	11.1/10.2/9.7	14.7/13.5/12.9
Gemini-1.5-Flash	20.5/20.3/18.5	9.5/9.4/8.6	13.2/12.8/11.7
Spark-3.5-Max	28.3/24.8/23.1	15.4/12.9/12.2	20.0/17.0/16.0
GLM4	26.8/22.1/21.6	14.3/11.3/11.1	18.6/14.9/14.7

表 5 关键句为第 1 句时, 主流大语言模型的实验结果 (%)

方法	准确率	召回率	*F*1分数
GPT-3.5-Turbo	80.0	57.1	66.7
Gemini-1.5-Flash	78.6	55.2	64.8
Spark-3.5-Max	81.7	60.3	69.4
GLM4	76.6	52.9	62.6

3.6.1 关键句位置

为了探究模型生成时是否对于句子出现位置敏感, 我们将关键句添加在原文中的第 1 个位置, 此时大语言模型的实验结果如表 5 所示. 通过实验结果可以发现,

在 prompt 中已经明确告知“不要考虑句子的位置信息”的情况下, 关键句出现的位置对于模型效果影响仍较大. 我们推测这一原因可能是大语言模型不能充分地理解 prompt. 以 GLM4 为例, 提示词中内容包括: “句子应当完全来自原文, 不需要加上任何修饰”, 而模型的回答中对原文中的标点符号进行了修改, 从而影响最后的实验结果, 如图 3 所示. 以及我们在提示词中要求模型输出 3 个句子, 模型回答中出现了 4 个或更多个. 为了完成后续实验, 本文在确保语义相同的情况下对不同模型之间的 prompt 进行了细微调整, 例如额外声明: “不要在句子之间添加标点符号或为句子标上序号”.

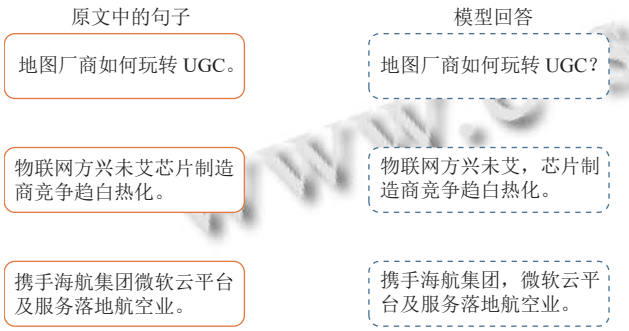

图 3 原文中的句子与模型回答的句子

3.6.2 生成内容不稳定性

本文使用同样的样本和提示词对同一模型进行了 10 次提问, 模型的回答中出现了 5 种不同的结果, 其中正确的回答出现了 1 次. 这一现象说明大语言模型在生成回答时具有不稳定性, 为大语言模型在关键句抽取领域应用中带来了一定的问题. 由于仅解码器架构本身的特性以及实验条件有限, 在进行本次实验时, 我们使用相同的输入对每一个模型进行 5 次实验, 将模型的最高分、最低分以及中位数信息呈现在表 4 中.

4 总结

本文提出了一种中文关键句抽取方法, 该方法采用编码器-解码器架构的生成式模型, 能够在无监督的环境中对中文文本进行自动的关键句抽取. 此外, 为进一步提高抽取准确性, 本文尝试在编码器部分引入对比学习进行半监督学习. 实验结果表明, 对比学习能够有效提高关键句抽取的准确性. 未来工作可以基于本研究进行拓展, 解决大语言模型在给予原文信息的问答中回答不准确的问题; 或对关键句抽取后进行分类, 以实现更深入的应用.

参考文献

1 Sheng Q, Cao J, Zhang XY, *et al*. Article reranking by memory-enhanced key sentence matching for detecting previously fact-checked claims. arXiv:2112.10322v1, 2021.
2 Gao YF, Xiong Y, Gao XY, *et al*. Retrieval-augmented generation for large language models: A survey. arXiv:2312.10997v5, 2024.
3 Mihalcea R, Tarau P. TextRank: Bringing order into text. Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. Barcelona: Association for Computational Linguistics, 2004. 404–411.
4 Berkhin P. A survey on PageRank computing. Internet Mathematics, 2005, 2(1): 73–120. [doi: 10.1080/15427951.2005.10129098]
5 Padmakumar A, Saran A. Unsupervised text summarization using sentence embeddings. Technical Report, Austin: University of Texas at Austin, 2016.
6 Kiros R, Zhu YK, Salakhutdinov R, *et al*. Skip-thought vectors. Proceedings of the 28th International Conference on Neural Information Processing Systems. Montreal: ACM, 2015. 3294–3302.
7 Liu Y. Fine-tune BERT for extractive summarization. arXiv:1903.10318v2, 2019.
8 Liu Y, Lapata M. Text summarization with pretrained encoders. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong: Association for Computational Linguistics, 2019. 3730–3740. [doi: 10.18653/v1/D19-1387]
9 Gokhan T, Smith P, Lee M. GUSUM: Graph-based unsupervised summarization using sentence features scoring and Sentence-BERT. Proceedings of the TextGraphs-16: Graph-based Methods for Natural Language Processing. Gyeongju: Association for Computational Linguistics, 2022. 44–53.
10 Reimers N, Gurevych I. Sentence-BERT: Sentence embeddings using siamese BERT-networks. Proceedings of the 2019 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. 3982–3992.
11 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: ACM, 2017. 6000–6010.

12 Radford A, Narasimhan K, Salimans T, *et al*. Improving language understanding by generative pre-training. https://hayate-lab.com/wp-content/uploads/2023/05/43372bfa750340059ad87ac8e538c53b.pdf. [2023-10-16].

13 Cao R, Wang YH, Liang YX, *et al*. Exploring the impact of negative samples of contrastive learning: A case study of sentence embedding. arXiv:2202.13093v3, 2022.

14 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. ACL, 2020. 7871–7880. [doi: 10.18653/v1/2020.acl-main.703]

15 Chung H W, Hou L, Longpre S, *et al*. Scaling instruction-finetuned language models. Journal of Machine Learning Research, 2024, 25(70): 1–53.

16 Tay Y, Dehghani M, Tran VQ, *et al*. UL2: Unifying language learning paradigms. arXiv:2205.05131v3, 2023.

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

18 Hadsell R, Chopra S, LeCun Y. Dimensionality reduction by learning an invariant mapping. Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2006. 1735–1742. [doi: 10.1109/CVPR.2006.100]

19 Gao TY, Yao XC, Chen DQ. SimCSE: Simple contrastive learning of sentence embeddings. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Punta Cana: ACL, 2021. 6894–6910.

20 Wu X, Gao CC, Zang LJ, *et al*. ESimCSE: Enhanced sample building method for contrastive learning of unsupervised sentence embedding. Proceedings of the 29th International Conference on Computational Linguistics. Gyeongju: ICCL, 2022. 3898–3907.

21 Ni J, Hernandez Abrego G, Constant N, *et al*. Sentence-T5: Scalable sentence encoders from pre-trained text-to-text models. Proceedings of the 2022 Association for Computational Linguistics. Dublin: ACL 2022. 1864–1874. [doi: 10.18653/v1/2022.findings-acl.146]

22 Kong AB, Zhao SW, Chen H, *et al*. PromptRank: Unsupervised keyphrase extraction using prompt. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics. Toronto: ACL, 2023. 9788–9801. [doi: 10.18653/v1/2023.acl-long.545]

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

24 Hu BT, Chen QC, Zhu FZ. LCSTS: A large scale Chinese short text summarization dataset. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon: ACL, 2015. 1967–1972. [doi: 10.18653/v1/D15-1229]

25 Ye JJ, Chen XT, Xu N, *et al*. A comprehensive capability analysis of GPT-3 and GPT-3.5 series models. arXiv:2303.10420, 2023.

26 Gemini Team Google, Anil R, Borgeaud S, *et al*. Gemini: A family of highly capable multimodal models. arXiv:2312.11805, 2024.

27 Glm T, Zeng A, Xu B, *et al*. ChatGLM: A family of large language models from GLM-130B to GLM-4 all tools. arXiv:2406.12793v2, 2024.

(校对责编: 王欣欣)