基于异构图和关键词的抽取式文本摘要模型

朱颀林[1]，王 羽[2,3]，徐 建[1*]

（1. 南京理工大学 计算机科学与工程学院，南京 210094；2. 国防科技大学 信息系统工程重点实验室，长沙 410003；
3. 中国电子科技集团公司第二十八研究所，南京 210007）

摘要 抽取式文本摘要使用一定的策略从冗长的文本中选择一些句子组成摘要，其关键在于要尽可能多地利用文本的语义信息和结构信息。为了更好地挖掘这些信息，进而利用它们指导摘要的抽取，提出了一种基于异构图和关键词的抽取式文本摘要模型（HGKSum）。该模型首先将文本建模为由句子节点和词语节点构成的异构图，在异构图上使用图注意力网络学习节点的特征，之后将关键词抽取任务作为文本摘要任务的辅助任务，使用多任务学习的方式进行训练，得到候选摘要，最后对候选摘要进行精炼以降低冗余度，得到最终摘要。在基准数据集上的对比实验表明，该模型性能优于基准模型，此外，消融实验也证明了引入异构节点和关键词的必要性。

关键词 抽取式文本摘要；异构图；关键词；图注意力网络；多任务学习
中图分类号 TP391 **文献标志码** A DOI 10.12178/1001-0548.2023019

Extractive Document Summarization Model Based on Heterogeneous Graph and Keywords

ZHU Qilin[1], WANG Yu[2,3], and XU Jian[1*]

(1. School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China;
2. Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410003, China;
3. The 28th Research Institute of China Electronics Technology Group Corporation, Nanjing 210007, China)

Abstract Extractive document summarization uses certain strategies to select some sentences from lengthy texts to form a summary, whose key is to use as much semantic and structural information of the text as possible. In order to better mine such information and then use it to guide the summarization, an extractive document summarization model based on heterogeneous graph and keywords (HGKSum) is proposed, which models the text as a heterogeneous graph composed of sentence nodes and word nodes. The model uses the graph attention networks to learn the features of the nodes in the graph. The multi-task learning is applied to the model, which considers the keywords extraction task as an auxiliary task of the document summarization task. The candidate summary which derived from the prediction of the neural networks in the model is often highly redundant, so the model refines it to create the final summary of low redundancy. The comparative experiment on the document summarization benchmark shows that the proposed model outperforms the baselines. Besides, ablation studies also demonstrate the necessity of introducing heterogeneous nodes and keywords.

Key words extractive document summarization; heterogeneous graph; keywords; graph attention network; multi-task learning

信息时代信息容量呈指数级增长，人们通常需要一份好的摘要来帮助自己处理如此庞大的信息，然而人工进行的摘要总结无论是从时间成本上还是经济成本上都已经变得越来越难以实施，因此依靠计算机的自动文本摘要技术已经成为了研究的热点。

自动文本摘要就是要在保留原文本主要思想的同时，对其进行压缩凝练，并最终产生一个简明流畅摘要的过程。一般来说，自动文本摘要按照其生成方式可以分为抽取式（extractive）摘要和生成式（abstractive）摘要[1]。生成式摘要对文本进行深度

收稿日期：2023−01−13；修回日期：2023−06−17
基金项目：国家自然科学基金（61872186）；国防基础科研计划国防科技重点实验室稳定支持项目（WDZC20225250405）
作者简介：朱颀林，主要从事自然语言处理方面的研究。
*通信作者 E-mail: dolphin.xu@njust.edu.cn

分析，产生一个由新生成的句子构成的摘要；而抽取式摘要则是选择原文本中重要的句子或段落，并把它们组合起来组成摘要。与生成式摘要相比，抽取式摘要具有语义正确性高、语法错误少、摘要速度快等优点，因此，本文关注的是抽取式摘要。

在抽取式文本摘要的研究中，最重要的是如何对句子的重要性进行评价。传统的方法通常是基于统计信息的方法，该类方法使用一些统计信息，如句子位置、词频等来识别文档中的重要句子。尽管这类方法具有实现简单、计算迅速等优点，但由于它们没有考虑句子和词语的语义信息，所以生成的摘要质量相对较差。

语义信息是指文本中的语义单元（如字、词、句子等）所携带的符合人类认知的有意义的信息。人类正是通过感知语义信息，才能理解文本所表达的含义。而结构信息则更侧重于语义单元在文本中的布局信息，如开头和结尾的语义单元通常更重要，再如两个语义单元之间的距离通常能反映二者之间的联系。

为了更好地获得句子的语义信息和结构信息，基于深度学习的抽取式文本摘要方法开始变得流行起来。这类方法通常采用的是编码器–解码器框架，并使用循环神经网络（Recurrent Neural Network, RNN）对句子进行编码。然而，基于循环神经网络的模型通常难以捕获句子级的长程依赖，进而导致模型表示的跨句关系不够丰富，影响摘要的质量。

一种直观的捕获跨句关系的方法是将文本表示为图。基于图的抽取式文本摘要方法将文本单元（如词语、句子等）建模为图中的节点，根据其在文本中的关系（如共现关系、语法关系等）进行连边，之后在图上进行节点的排序算法得到摘要。大多数基于图的抽取式文本摘要方法构建的是同构图，只考虑句子一种节点，而忽视了词语等其他类型的文本单元，而少数基于异构图的方法又严重依赖外部工具，存在着错误传播的问题。

与摘要类似，关键词也能表示文本的主要信息，而且从某种程度上来讲，关键词可以看作是一种更细粒度的摘要。容易发现，关键词常常贯穿于整个文本中，且集中出现在参考摘要中，因此关键词对于跨句关系的捕获以及摘要的抽取有着至关重要的指导作用。然而目前的抽取式文本摘要方法却常常忽视这一重要信息，对其利用程度还远远不够。

基于此，本文研究了基于异构图和关键词的抽取式文本摘要模型 HGKSum（Summarization Based on Heterogeneous Graph and Keywords）。为了丰富句子间的关系，此模型不仅依靠句子节点构图，还引入了词语节点，构造一种异构图。词语节点可以看作是句子节点之间的桥梁，让没有直接相邻的句子也间接联系在了一起。在如何对文本图进行学习的方面，模型采用图注意力网络来学习节点特征，可以很好地捕获图的结构信息。在如何利用关键词信息方面，模型使用关键词信息来缓解噪声词语节点对文本结构的影响，此外，由于关键词抽取任务和文本摘要任务具有互补性，模型使用多任务学习的思想，将关键词抽取任务作为文本摘要任务的辅助任务，在训练阶段，不仅对句子节点进行预测，对词语节点也进行预测，二者联合训练，能够获得更佳的摘要。

1 相关工作

本文提出的模型在摘要过程中融入了关键词信息，因此相关工作主要包括文本摘要以及关键词抽取。

1.1 文本摘要

文献 [2] 开创了系统研究文本摘要的先河，提出在排除停止词后，包含高频词语越多的句子越有可能成为摘要。之后，越来越多的文本摘要方法被提出，主要包括抽取式和生成式两类方法。生成式文本摘要方法目前主流的框架为序列到序列（Sequence to Sequence, Seq2Seq）模型，如文献 [3] 在编码阶段引入了注意力机制，在解码阶段使用束搜索（Beam Search）来生成摘要。由于本文提出的模型为抽取式模型，因此将重点关注抽取式文本摘要方法。

抽取式文本摘要方法大致可以分为有监督的方法和无监督的方法两类，有监督的方法需要标注数据，而无监督的方法则不需要提前标注。传统的基于统计信息的方法和基于图的方法都属于无监督的方法。早期的方法通常是基于统计信息的方法，文献 [4] 在关键词频的基础上引入了线索词（Cue Words）、标题词、句子位置等因素来计算句子权重。由于图能够很好地表示结构信息，基于图的方法也很早就得到了研究人员的关注。文献 [5] 根据句子的余弦相似度进行构图，使用文献 [6] 的思想对图中节点进行排序，提出了 TextRank 算法。文献 [7] 研究了使用语义角色信息来改进基于图的多文档摘要排序算法。文献 [8] 引入大规模预训练语言模型来捕获语义信息，并根据句子的位置信息将传统的无向图转换为有向图来进行摘要。

基于机器学习的方法通常是有监督的方法，其中基于深度学习的方法成为了新的研究热点。文献 [9] 提出了 SummaRuNNer，这是一种基于顺序分类来提取文本摘要的两层双向门控循环单元（Gated Recurrent Unit, GRU）序列模型，其中每个句子按原始顺序依次访问，并判断是否使用它作为摘要。文献 [10] 研究了强化学习在抽取式文本摘要任务上的应用，提出了一种基于深度 Q 网络（Deep Q-Network, DQN）的摘要方法，该方法利用 Q 值对句子的显着性和冗余性进行建模。文献 [11] 针对案件舆情领域的摘要问题，提出使用图卷积网络（Graph Convolutional Network, GCN）来进行摘要。此外，基于 BERT[12] 等预训练模型的方法也得到了许多研究并取得了不错的效果。MatchSum[13] 将抽取式文本摘要任务视作语义匹配问题，使用孪生 BERT 网络在摘要粒度上进行训练，选择最佳的候选摘要成为最终摘要。BertSum[14] 通过修改 BERT 模型的输入并叠加不同的分类器来对其进行微调，并最终获得摘要。

1.2　关键词抽取

作为理解文本内容的最小单位，关键词表征了文本主题性和关键性的内容，并得到了研究人员的广泛关注。文献 [15] 提出了使用句子聚类和潜在狄利克雷分布的单文档关键短语提取算法。该算法利用句子向量之间的余弦相似度进行聚类，以突出显示语义相关的文本部分，之后分析反映文档主题的句子集群以获得文本的主要主题，这些主题中最重要的词被提取为关键词。YAKE 算法[16] 基于文档的统计特征进行关键词抽取。该算法充分考虑了文档中词语的一些属性，如大小写、位置、频率、停止词的相似度和包含该单词的句子数等，提出了统一的评分公式，并最终生成冗余度低的关键词。文献 [17] 提出引入瑞丽熵来对词语重要性进行评判，进而抽取关键词。

与抽取式文本摘要算法类似，基于图的关键词抽取算法也是一类重要的算法。sCAKE 算法[18] 关注词语在邻接句子之间的共现关系，利用图的 truss 分解导出文档中词语的语义关系，利用词语间的语义关系抽取关键词。文献 [19] 选择使用 k-core、k-truss、k-shell 等对语言网络图进行图收缩，抽取出影响力大的节点作为文档关键词。FLAKE 算法[20] 根据词语在文档中出现的位置和次数，依靠模糊逻辑将文档构建为一个模糊图，并根据模糊图中节点的中心度来确定关键词。MGRank 算法[21] 提出使用多重图来建模文档，使得两个节点之间的边数可以多于一条，提高了抽取关键词的质量。

2　HGKSum 模型

2.1　模型框架

为了捕获复杂的跨句关系，提出一种基于异构图和关键词的抽取式文本摘要模型，命名为 HGKSum，其框架如图 1 所示，其中圆形节点、矩形节点、五边形节点、菱形节点分别代表重要词语、重要句子、普通词语、普通句子，黄色网络、红色网络、绿色网络、蓝色网络分别代表词−词节点特征学习、句−句节点特征学习、句−词节点特征学习、词−句节点特征学习。HGKSum 主要包括异构图构建、文本向量化、异构图节点特征学习、多任务学习、摘要精炼 5 部分。首先，模型根据句子、词语之间的关系以及权重大小构建异构的文本结构图，之后对图的节点及边的特征进行向量化，再依靠图注意力网络对节点特征进行学习，依次进行词−词节点特征学习、句−句节点特征学习、句−词节点特征学习和词−句节点特征学习，整体训练方式为多任务学习，包括关键词抽取任务和文本摘要任务，最后进行精炼得到最终摘要。

2.2　异构图构建

2.2.1　图的结构

异构图即图中节点类型或边的类型不止一种，而同构图则是图中节点类型和边的类型都只有一种，可以在一定程度上认为是异构图的特殊情况。异构图可形式化表示如下。

记图$\mathcal{G}=(\mathcal{V},\mathcal{E})$，其中$\mathcal{V}$表示图中节点的集合，$\mathcal{E}$表示图中边的集合，若图$\mathcal{G}$还具有节点类型映射函数$\pi:\mathcal{V}\to\mathcal{A}$和边类型映射函数$\varpi:\mathcal{E}\to\mathcal{R}$，其中$\mathcal{A}$表示预定义的节点类型，$\mathcal{R}$表示预定义的边类型，且$|\mathcal{A}|+|\mathcal{R}|>2$，那么图$\mathcal{G}$就是一个异构图。

模型将文本建模为无向异构图$G=\{V,E\}$，其中V表示图中节点的集合，E表示图中边的集合。图中的节点包括两类，分别是句子节点和词语节点，即$V=V_{\mathrm{w}}\cup V_{\mathrm{s}}$，其中$V_{\mathrm{w}}=\{w_1,w_2,\cdots,w_m\}$表示的是文本经过分词得到的 m 个独特的词语组成的集合，$V_{\mathrm{s}}=\{s_1,s_2,\cdots,s_n\}$表示的是文本经过分句得到的 n 个独特的句子组成的集合。图中的边$E=\{(v_i,v_j)|v\in V\}$，具体构建方式如下。

1）当两个词语在一句话中共现时，二者对应的节点进行连边；

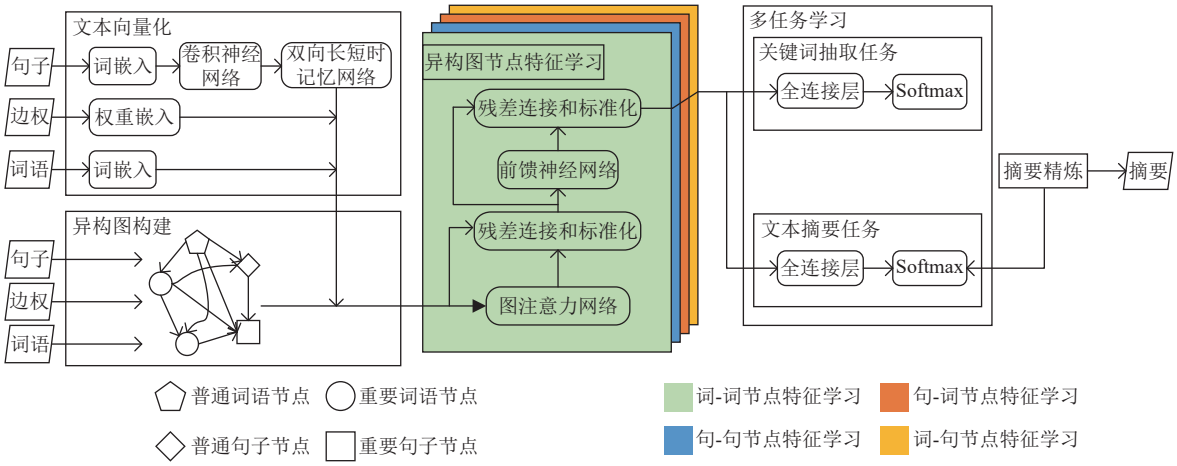

图 1 HGKSum 框架

2）当两个句子相邻时，二者对应的节点进行连边；

3）当一个词语出现在一个句子中时，二者对应的节点进行连边；

4）对每个节点添加自环。

记节点 $V=\{w_1,w_2,\cdots,w_m,s_1,s_2,\cdots,s_n\}$，异构图邻接矩阵为 $\boldsymbol{A}$，可知 $\boldsymbol{A}$ 为分块矩阵，包括词–词邻接矩阵 $\boldsymbol{A}_{\mathrm{ww}}$、句–句邻接矩阵 $\boldsymbol{A}_{\mathrm{ss}}$ 以及词–句邻接矩阵 $\boldsymbol{A}_{\mathrm{ws}}$，那么上述构建方式可形式化描述如下。

$$\boldsymbol{A}=\begin{bmatrix}\boldsymbol{A}_{\mathrm{ww}} & \boldsymbol{A}_{\mathrm{ws}}\\ \boldsymbol{A}_{\mathrm{ws}}^{\mathrm{T}} & \boldsymbol{A}_{\mathrm{ss}}\end{bmatrix} \tag{1}$$

$$A_{ij}^{\mathrm{ww}}=\begin{cases}1 & \text{if } w_i=w_j \text{ or } w_i \text{ and } w_j\\ & \text{co-occur in one sentence}\\ 0 & \text{otherwise}\end{cases} \tag{2}$$

$$A_{ij}^{\mathrm{ss}}=\begin{cases}1 & \text{if } s_i=s_j \text{ or } s_i \text{ and } s_j \text{ are consecutive}\\ 0 & \text{otherwise}\end{cases} \tag{3}$$

$$A_{ij}^{\mathrm{ws}}=\begin{cases}1 & \text{if } w_i \text{ occurs in } s_j\\ 0 & \text{otherwise}\end{cases} \tag{4}$$

2.2.2 图的边权

为了捕获文本更多的信息，模型构建的是带权图，权重的计算针对不同类型的边综合考虑了多种统计特征。具体来说，对于词–词边，主要考虑词语的共现次数（Co-occurrence Counts, CC）；对于句–句边，主要考虑句子的位置（Position, POS）；对于词–句边，主要考虑词语的词频–逆文档频率（Term Frequency–Inverse Document Frequency, TF-IDF）。

共现次数 CC 即两个词语在一句话中共同出现的次数，该特征表示了两个词语间的紧密程度。由于文本开头的句子通常较为重要，因此句子位置 POS 也能在一定程度上反映出句子的重要性。词频–逆文档频率 TF-IDF 表示了一个词对于一句话的重要程度，具体计算公式为：

$$\mathrm{TF\text{-}IDF}(w,s)=\mathrm{TF}(w,s)\times\log\frac{n}{1+\mathrm{DF}(w)} \tag{5}$$

式中，$\mathrm{TF}(w,s)$ 表示词语 w 在句子 s 中的词频；n 表示文本中句子的数目；$\mathrm{DF}(w)$ 表示文本中包含词语 w 的句子数目。对于自环的权重，模型可以将其设为一个特殊值，如 0。综上所述，权重的具体计算为：

$$e_{ij}=\begin{cases}\mathrm{TF\text{-}IDF}\\ \text{if } (v_i\in V_{\mathrm{s}} \text{ and } v_j\in V_{\mathrm{w}}) \text{ or}\\ \quad (v_i\in V_{\mathrm{w}} \text{ and } v_j\in V_{\mathrm{s}})\\ \mathrm{CC}\\ \text{if } v_i\neq v_j \text{ and } v_i\in V_{\mathrm{w}} \text{ and}\\ \quad v_j\in V_{\mathrm{w}}\\ \mathrm{POS}\\ \text{if } v_i\neq v_j \text{ and } v_i\in V_{\mathrm{s}} \text{ and}\\ \quad v_j\in V_{\mathrm{s}}\\ 0\\ \text{if } v_i=v_j\end{cases} \tag{6}$$

2.2.3 重要节点筛选

为了减轻噪声节点对于节点特征学习的影响，传统做法是使用固定的停止词列表进行停止词消除，而本模型则是引入关键信息来筛选重要节点。

TextRank 算法[5] 是一种高效的无监督的关键词抽取算法，同时它也能抽取摘要句，它将文本构建为图，在图上使用 PageRank 算法[6] 对节点打分以得到重要节点。本模型对待摘要原文使用 TextRank 算法分别获取 k 个关键词和 l 个摘要句作为重要节点，分别记作 V_{kw} 和 V_{ks}，可知 $V_{\mathrm{kw}}\subseteq V_{\mathrm{w}}$，$V_{\mathrm{ks}}\subseteq V_{\mathrm{s}}$。

重要节点将在节点特征学习中控制信息的传递方式，其具有更高的权重，能够参与更多的学习过程，本文将在后文对其进行详细介绍。

2.3 文本向量化

无论是词语、句子还是权重都需要将他们向量化才能进行学习。本文将词向量、句向量和权重向量分别表示为$\boldsymbol{x}_{\mathrm{w}} \in \mathbb{R}^{d_{\mathrm{w}}}$、$\boldsymbol{x}_{\mathrm{s}} \in \mathbb{R}^{d_{\mathrm{s}}}$和$\boldsymbol{x}_{\mathrm{e}} \in \mathbb{R}^{d_{\mathrm{e}}}$，其中$d_{\mathrm{w}}$表示词向量的维度，$d_{\mathrm{s}}$表示句向量的维度，$d_{\mathrm{e}}$表示权重向量的维度。

对于词向量$\boldsymbol{x}_{\mathrm{w}}$，为了在一开始就获得一个较好的语义表示，模型采用预训练的词向量模型进行初始化，如 GloVe[22]。

对于句向量$\boldsymbol{x}_{\mathrm{s}}$，由于句子是词语的集合，所以模型首先对句子中的每个词语进行初始化得到$\boldsymbol{x}_{\mathrm{w}}$，之后基于每个句子的词向量$\boldsymbol{x}_{\mathrm{w}}$集合，使用卷积神经网络（Convolutional Neural Network, CNN）[23]捕获句子的局部特征$\boldsymbol{x}'_{\mathrm{s}}$，最后为了获得句子的全局特征，模型将这些$\boldsymbol{x}'_{\mathrm{s}}$送入双向长短时记忆网络（Bi-Directional Long Short-Term Memory, BiLSTM）[24]中来得到$\boldsymbol{x}_{\mathrm{s}}$。

对于权重向量$\boldsymbol{x}_{\mathrm{e}}$，由于模型构建的异构图的权重为无界的连续值，所以需要先对其进行离散化编码，再将其向量化。模型采用了一种简单的编码方式，首先针对不同的权重类型，对权重值分别进行线性归一化，并将其放缩到适当的区间内，如[1,10]，而后对新的权重值进行四舍五入操作，将其离散化，如此便得到了 3×10+1=31 个码点，即权重向量的“词表”大小为 31。在对原始权重值编码后，模型采用随机初始化的方式初始化可学习的权重向量$\boldsymbol{x}_{\mathrm{e}}$。

2.4 异构图节点特征学习

2.4.1 模型结构

作为图神经网络的一种，图注意力网络（Graph Attention Network, GAT）[25]可以直接学习图结构，且可以应对归纳式任务。因此，模型采用图注意力网络来学习文本图中语义节点的特征。

设图中节点特征维度为d_{h}，则第 i 个节点的特征向量可表示为$\boldsymbol{h}_i \in \mathbb{R}^{d_{\mathrm{h}}}, i \in \{1,2,\cdots,(m+n)\}$，图注意力网络只考虑其邻居节点的特征，并使用 LeakyReLU[26]来增强模型的非线性表达能力，具体为：

$$\mathrm{LeakyReLU}(x) = \max(0,x) + \rho\min(0,x) \tag{7}$$

$$z_{ij} = \mathrm{LeakyReLU}\left(\boldsymbol{a}^{\mathrm{T}}\left[\boldsymbol{W}\boldsymbol{h}_i \| \boldsymbol{W}\boldsymbol{h}_j\right]\right) \tag{8}$$

$$\alpha_{ij} = \frac{\exp(z_{ij})}{\sum\limits_{k\in\mathcal{N}_i}\exp(z_{ik})} \tag{9}$$

$$\boldsymbol{u}_i = \phi\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}\boldsymbol{W}\boldsymbol{h}_j\right) \tag{10}$$

式中，ρ为 LeakyReLU 的超参数；线性变换矩阵$\boldsymbol{W} \in \mathbb{R}^{d'_{\mathrm{h}}\times d_{\mathrm{h}}}$用作特征增强；注意力向量$\boldsymbol{a} \in \mathbb{R}^{2d'_{\mathrm{h}}}$用于实现注意力机制；节点 j 为节点 i 的某一邻居节点；α_{ij}表示节点 i 与 j 之间的注意力系数；$\mathcal{N}_i$表示节点 i 的邻居节点集合；ϕ表示任意激活函数，如 ELU 函数等。拥有 P 个头的多头图注意力网络可以表示为：

$$\boldsymbol{u}_i = \|_{p=1}^{P}\phi\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}^{p}\boldsymbol{W}^{p}\boldsymbol{h}_j\right) \tag{11}$$

式中，$(\cdot)^p$表示在第 p 个头中相应的参数；||表示向量拼接。

文献 [25] 提出的图注意力网络中的注意力机制已经成为事实上的标准，如式 (8) 所示。然而，文献 [27] 研究发现由于$\boldsymbol{W}$和$\boldsymbol{a}$连在一块，二者退化为一个可学习参数，因此式 (8) 实现的是“静态注意力”，即给定一组固定的键（节点 i 的邻居节点集），不同的查询（节点 i）得到的注意力分数排名不变。毫无疑问，“静态注意力”使得模型的表征能力大打折扣。此外，原始的图注意力网络没有考虑边权信息，为了有效利用这些信息，模型在文献 [27] 工作的基础上，采用式 (12) 实现注意力机制。

$$z_{ij} = \boldsymbol{a}^{\mathrm{T}}\mathrm{LeakyReLU}\left(\boldsymbol{W}\left[\boldsymbol{h}_i \| \boldsymbol{h}_j \| \boldsymbol{e}_{ij}\right]\right) \tag{12}$$

式中，$\boldsymbol{e}_{ij} \in \mathbb{R}^{d_{\mathrm{e}}}$表示节点 i 与 j 之间的权重向量。综合式 (9)～式 (12) 将图注意力网络表示为GAT($\boldsymbol{H}_{\mathrm{q}}, \boldsymbol{H}_{\mathrm{k}}, \boldsymbol{H}_{\mathrm{v}}$)，其中$\boldsymbol{H}_{\mathrm{q}}, \boldsymbol{H}_{\mathrm{k}}, \boldsymbol{H}_{\mathrm{v}}$分别表示注意力机制中的查询节点向量、键节点向量和值节点向量。

Transformer[28]模型在许多自然语言处理任务中都取得了优异的成绩。为了提高表达能力，模型借鉴了 Transformer 模型的结构，在每个图注意力网络后面都引入一个全连接的前馈神经网络（Feed-Forward Network, FFN），并引入了残差连接[29]和层标准化（Layer Normalization, LayerNorm）[30]。设输入特征为$\boldsymbol{h} \in \mathbb{R}^{d_{\mathrm{h}}}$，前馈神经网络和层标准化可具体表示如下：

$$\mathrm{ReLU}(x) = \max(0,x) \tag{13}$$

$$\mathrm{FFN}(\boldsymbol{h}) = \boldsymbol{W}_{\mathrm{f2}}\mathrm{ReLU}(\boldsymbol{W}_{\mathrm{f1}}\boldsymbol{h} + \boldsymbol{b}_{\mathrm{f1}}) + \boldsymbol{b}_{\mathrm{f2}} \tag{14}$$

$$\mathrm{LayerNorm}(\boldsymbol{h}) = \boldsymbol{g} \odot \frac{\boldsymbol{h} - \mu_h}{\sigma_h} + \boldsymbol{b} \tag{15}$$

式中，ReLU[31] 为线性整流函数；$\boldsymbol{W}_{\mathrm{f1}} \in \mathbb{R}^{d_{\mathrm{f}} \times d_{\mathrm{h}}}$；$\boldsymbol{W}_{\mathrm{f2}} \in \mathbb{R}^{d_{\mathrm{h}} \times d_{\mathrm{f}}}$；$\boldsymbol{b}_{\mathrm{f1}} \in \mathbb{R}^{d_{\mathrm{f}}}$，$\boldsymbol{b}_{\mathrm{f2}} \in \mathbb{R}^{d_{\mathrm{h}}}$，$\boldsymbol{g} \in \mathbb{R}^{d_{\mathrm{h}}}$，$\boldsymbol{b} \in \mathbb{R}^{d_{\mathrm{h}}}$均为可学习参数；$\mu_h$为 $\boldsymbol{h}$ 的总体均值；σ_h为 $\boldsymbol{h}$ 的总体标准差；⊙为矩阵的哈达玛积（两个同型矩阵对应位置元素相乘）。

2.4.2 节点特征学习过程

模型构建的异构图的节点类型可粗粒度地分为词语节点和句子节点两种，每种节点均可以作为图注意力网络的查询节点和键节点，因此一共有 4 种学习方式，分别是词−词节点特征学习、句−句节点特征学习、句−词节点特征学习、词−句节点特征学习。为了更好地捕获节点之间的关系，不同学习方式的学习次序显得尤为重要。

由于同构节点之间的特征相似性较高易于学习，而异构节点之间的特征差异性较高需要更好的节点特征表示才能获得好的学习效果。因此模型先进行同构节点特征的学习，而后进行异构节点特征的学习。此外，由于主任务是文本摘要任务，因此在异构节点特征的学习中，应先进行句−词节点特征学习以获得更好的词特征，最后进行词−句节点特征学习。

基于图注意力网络的节点特征学习可以朴素地理解为节点根据其邻居节点的特征来更新自己的特征，因此噪声节点的存在将影响学习效果。假如一个句子中大部分都是停止词，那么其进行词−句节点特征学习时，其邻居节点大部分都是停止词对应的词语节点，最终其学习到的特征将趋向于意义不大的停止词的特征。为了缓解这一问题，模型引入重要节点，在异构节点特征学习过程中，只有重要节点才能成为键节点和值节点，使得查询节点特征更加关注关键信息。

综上所述，节点特征经过初始化后，首先进行词−词节点特征学习和句−句节点特征学习，而后进行句−词节点特征学习，最后进行词−句节点特征学习，学习过程如式 (16)～式 (25) 所示。

$$\boldsymbol{H}_{\mathrm{w}}^{0} = \boldsymbol{X}_{\mathrm{w}} \tag{16}$$

$$\boldsymbol{H}_{\mathrm{s}}^{0} = \boldsymbol{X}_{\mathrm{s}} \tag{17}$$

$$\boldsymbol{U}_{\mathrm{w}\leftarrow\mathrm{w}} = \mathrm{LayerNorm}\left(\mathrm{GAT}\left(\boldsymbol{H}_{\mathrm{w}}^{0}, \boldsymbol{H}_{\mathrm{w}}^{0}, \boldsymbol{H}_{\mathrm{w}}^{0}\right) + \boldsymbol{H}_{\mathrm{w}}^{0}\right) \tag{18}$$

$$\boldsymbol{H}_{\mathrm{w}}^{1} = \mathrm{LayerNorm}\left(\mathrm{FFN}\left(\boldsymbol{U}_{\mathrm{w}\leftarrow\mathrm{w}}\right) + \boldsymbol{U}_{\mathrm{w}\leftarrow\mathrm{w}}\right) \tag{19}$$

$$\boldsymbol{U}_{\mathrm{s}\leftarrow\mathrm{s}} = \mathrm{LayerNorm}\left(\mathrm{GAT}\left(\boldsymbol{H}_{\mathrm{s}}^{0}, \boldsymbol{H}_{\mathrm{s}}^{0}, \boldsymbol{H}_{\mathrm{s}}^{0}\right) + \boldsymbol{H}_{\mathrm{s}}^{0}\right) \tag{20}$$

$$\boldsymbol{H}_{\mathrm{s}}^{1} = \mathrm{LayerNorm}\left(\mathrm{FFN}\left(\boldsymbol{U}_{\mathrm{s}\leftarrow\mathrm{s}}\right) + \boldsymbol{U}_{\mathrm{s}\leftarrow\mathrm{s}}\right) \tag{21}$$

$$\boldsymbol{U}_{\mathrm{w}\leftarrow\mathrm{s}} = \mathrm{LayerNorm}\left(\mathrm{GAT}\left(\boldsymbol{H}_{\mathrm{w}}^{1}, \boldsymbol{H}_{\mathrm{ks}}^{1}, \boldsymbol{H}_{\mathrm{ks}}^{1}\right) + \boldsymbol{H}_{\mathrm{w}}^{1}\right) \tag{22}$$

$$\boldsymbol{H}_{\mathrm{w}}^{2} = \mathrm{LayerNorm}\left(\mathrm{FFN}\left(\boldsymbol{U}_{\mathrm{w}\leftarrow\mathrm{s}}\right) + \boldsymbol{U}_{\mathrm{w}\leftarrow\mathrm{s}}\right) \tag{23}$$

$$\boldsymbol{U}_{\mathrm{s}\leftarrow\mathrm{w}} = \mathrm{LayerNorm}\left(\mathrm{GAT}\left(\boldsymbol{H}_{\mathrm{s}}^{1}, \boldsymbol{H}_{\mathrm{kw}}^{2}, \boldsymbol{H}_{\mathrm{kw}}^{2}\right) + \boldsymbol{H}_{\mathrm{s}}^{1}\right) \tag{24}$$

$$\boldsymbol{H}_{\mathrm{s}}^{2} = \mathrm{LayerNorm}\left(\mathrm{FFN}\left(\boldsymbol{U}_{\mathrm{s}\leftarrow\mathrm{w}}\right) + \boldsymbol{U}_{\mathrm{s}\leftarrow\mathrm{w}}\right) \tag{25}$$

式中，$\boldsymbol{H}_{\mathrm{w}}^{i} \in \mathbb{R}^{d_{\mathrm{h}} \times m}, \boldsymbol{H}_{\mathrm{s}}^{i} \in \mathbb{R}^{d_{\mathrm{h}} \times n}, \boldsymbol{H}_{\mathrm{kw}}^{i} \in \mathbb{R}^{d_{\mathrm{h}} \times k}, \boldsymbol{H}_{\mathrm{ks}}^{i} \in \mathbb{R}^{d_{\mathrm{h}} \times l}$，$i \in \{0,1,2\}$分别代表时间步为 i 时的词向量、句向量、重要词语向量、重要句子向量；$\boldsymbol{X}_{\mathrm{w}}$和$\boldsymbol{X}_{\mathrm{s}}$分别代表初始化的词向量和句向量。

2.5 多任务学习

抽取式文本摘要任务和关键词抽取任务均可被视作一个文本分类任务。给定一篇具有 n 个句子的文档 $D = \{s_1, s_2, \cdots, s_n\}$，其具有大小为 m 的词表 $\mathrm{Vocab} = \{w_1, w_2, \cdots, w_m\}$，抽取式文本摘要的任务是对每个句子$s_i$预测其是否为摘要句，关键词抽取任务则是对每个词语$w_i$预测其是否为关键词。不难发现，二者在任务定义上具有高度的相似性。此外，从优化角度来看，二者都是从文本中抽取重要的文本单元，且通常使用交叉熵作为损失函数，具有相同的优化方向。因此为了提高性能，模型使用多任务学习的方式进行训练，将抽取式文本摘要任务作为主任务，将关键词抽取任务作为辅助任务，二者联合训练，希望二者能够互相补充和促进，进而获得更好的摘要效果。

模型对异构图中的每个节点使用简单的全连接层进行二分类，句子节点将被分为摘要句和非摘要句，词语节点将被分为关键词和非关键词，分别为：

$$\boldsymbol{Y}_{\mathrm{s}} = \boldsymbol{W}_{\mathrm{fc}} \boldsymbol{H}_{\mathrm{s}}^{2} + \boldsymbol{b}_{\mathrm{fc}} \tag{26}$$

$$\boldsymbol{Y}_{\mathrm{w}} = \boldsymbol{W}_{\mathrm{fc}} \boldsymbol{H}_{\mathrm{w}}^{2} + \boldsymbol{b}_{\mathrm{fc}} \tag{27}$$

式中，$\boldsymbol{W}_{\mathrm{fc}} \in \mathbb{R}^{2 \times d_{\mathrm{h}}}$；$\boldsymbol{b}_{\mathrm{fc}} \in \mathbb{R}^{2}$为全连接层的权重和偏置。

在一篇文章中大部分句子和词语都是普通句子和普通词语，摘要句以及关键词占比通常很小，因此样本不平衡问题广泛存在于抽取式文本摘要任务以及关键词抽取任务中，导致模型难以关注文本的关键部分。为了缓解这一问题，模型使用带权的交叉熵作为损失函数，对于样本数少的类别将赋予其较大的权重。带权的交叉熵损失函数定义为：

$$\mathrm{CE}(p, q) = -\sum_{n=1}^{N} \sum_{c=1}^{C} w_c q_{n,c} \log\left(p_{n,c}\right) \tag{28}$$

式中，C 为类别数目；N 为样本数目；w_c为第 c 类的权重；$q_{n,c} \in \{0,1\}$为样本 n 在第 c 类的标签；

$p_{n,c}$为模型预测的样本 n 属于第 c 类的概率。

最终的损失函数由两部分组成，分别用于抽取式文本摘要任务和关键词抽取任务，定义如下：

$$L=\lambda\mathcal{L}_{\mathrm{s}}+(1-\lambda)\mathcal{L}_{\mathrm{w}}=\lambda\mathrm{CE}(\mathrm{Softmax}(\boldsymbol{Y}_{\mathrm{s}}),\boldsymbol{T}_{\mathrm{s}})+(1-\lambda)\mathrm{CE}(\mathrm{Softmax}(\boldsymbol{Y}_{\mathrm{w}}),\boldsymbol{T}_{\mathrm{w}}) \tag{29}$$

式中，$0\leqslant\lambda\leqslant1$为调节系数；$\boldsymbol{T}_{\mathrm{s}}$和$\boldsymbol{T}_{\mathrm{w}}$分别为句子标签和词语标签。

上面仅是从理论上讨论了模型的结构和训练过程，然而在实际应用中，大部分文本摘要领域的数据集并未给出模型训练所需的关键词标签。因此，针对在实际训练过程中数据集标签缺失的问题，模型根据数据集提供的参考摘要，使用启发式的方法生成关键词标签。为了减小待摘要原文中无关词的影响，模型认为关键词出现在参考摘要中的概率比出现在待摘要原文中的概率高，因此选择对参考摘要使用无监督的 TextRank 算法来抽取关键词作为标签。

2.6　摘要精炼

一份好的摘要不仅要尽可能多地覆盖原文大意，还要具有尽可能低的冗余信息。通过前面的步骤，模型得到了每个句子的得分$\boldsymbol{Y}_{\mathrm{s}}$。给定摘要预期长度 K，传统的解码方式是选择得分最高的 K 个句子形成摘要。然而，这样做会引入大量的冗余信息，导致最后的摘要质量不佳。其原因在于人们通常会使用多个不同的句子描述同一件事情，特别是在描述重要事情时，这就导致许多得分高的句子其实是冗余句子。

三元语法（trigram）是指文本中连续出现的 3 个语词，如“This is a sentence”这句话中所有的三元语法为“This is a”和“is a sentence”。三元语法作为一种文本语义单元，可以在一定程度上度量两篇文本的冗余程度。本文启发式地认为，如果两个句子具有相同的三元语法，那么这两个句子是冗余的。基于此，为了获得低冗余的摘要，模型采用文献 [32] 提出的方法，在解码阶段根据三元语法的重叠情况，使用最大边缘相关[33] 的思想进行摘要精炼来得到最终摘要。

具体流程如算法 1 所示，首先初始化已选择句子的三元语法集合为空集，并对每个句子按照其得分由高到低进行排序，之后依次遍历每个句子直到已选择句子长度达到预期摘要长度，在遍历时，计算当前句子的三元语法集合和已选择句子的三元语法集合，如果二者不相交，那么选择该句子作为摘要句。

算法 1　摘要精炼算法

输入：预期摘要长度 K，所有句子 Sents，句子得分 $\boldsymbol{Y}_{\mathrm{s}}$

输出：最终摘要 Summary

```
Summary ← [], trigrams_set ← ∅, sents_num ← 0
sorted_sent ← sort(Sents, Y_s)
for sent in sorted_sent do
    if sents_num = K
        return Summary
    end if
    sent_trigrams ← get_trigrams(sent)
    if is_disjoint(sent_trigrams, trigrams_set)
        trigrams_set ← trigrams_set | sent_trigrams
        Summary.append(sent)
        sent_num++
    end if
end for
return Summary
```

3　实验与分析

3.1　数据集

目前文本摘要领域的大部分数据集每篇文档篇幅相对较短，且并不提供适用于抽取式文本摘要训练的标签，难以直接将其应用在有监督的抽取式文本摘要训练中来。此外许多数据集并非开源数据集，需要付费获得授权才能使用，门槛较高。为了确保实验的公平性和可靠性，本文选择使用 CNN/DailyMail 数据集[34-35] 作为实验数据集。

CNN/DailyMail 数据集是目前单文档文本摘要领域中最常用的测试基准，其主要内容为摘自 CNN 和 DailyMail 上的新闻，平均文档长度为 781 个单词，平均摘要长度为 3 个句子。本文使用该数据集的非匿名版本[36]，其中训练集、验证集和测试集的样本数分别为 287 226、13 368 和 11 490。

3.2　评价标准

本文采用自动文本摘要领域中最常用的 ROUGE[37] 指标作为摘要质量的评价指标。ROUGE 指标中常用的有 ROUGE-N 和 ROUGE-L。ROUGE-N 基于 n 元语法（n-gram）在系统摘要和参考摘要间的重叠程度来评价系统摘要的质量，ROUGE-L 则依据的是最长公共子序列的重叠程度，二者计算公式分别如下：

$$\mathrm{ROUGE-N}=\frac{\sum_{S\in\{Ref\}}\sum_{gram_n\in S}\mathrm{Count_{match}}(gram_n)}{\sum_{S\in\{Ref\}}\sum_{gram_n\in S}\mathrm{Count}(gram_n)} \tag{30}$$

$$\mathrm{ROUGE-L}=\frac{\sum_{S\in\{Ref\}}\mathrm{LCS}(S,Sys)}{\sum_{S\in\{Ref\}}\mathrm{Length}(S)} \tag{31}$$

式中，Ref代表参考摘要组成的集合；Sys 代表系统摘要；$gram_n$代表 n 元语法；$\mathrm{Count_{match}}(gram_n)$代表系统摘要和参考摘要重叠的 n 元语法个数；$\mathrm{Count}(gram_n)$代表参考摘要中 n 元语法个数；LCS 表示以词语为单位的最长公共子序列的长度；Length 表示以词语为单位的摘要长度。

3.3 实验设置

实验中模型的超参数设置如下：采用 6 头图注意力网络，初始化的词向量维度d_{w}、句向量维度d_{s}以及图注意力网络中的节点向量维度d_{h}均设为 300，前馈神经网络的隐层节点维度d_{f}设为 1 200，权重向量维度d_{e}设为 54。在训练过程中，本文使用 Adam[38] 作为优化器，学习率设为 5×10^{-5}，批尺寸设为 32，损失函数摘要句类别权重设为 9.98，非摘要句类别权重设为 0.55，关键词类别权重设为 13.47，非关键词类别权重设为 0.53。为了防止过拟合，本文采用早停策略，如果连续 3 轮训练中验证集上的指标都未下降，就提前停止训练。在测试阶段，摘要预期长度设为 3 个句子，并使用版本为 1.5.5 的官方 ROUGE 脚本进行测评，测评选项为“-m -n 2”。每个实验进行 3 次，实验结果取平均值。

3.4 敏感性分析

首先，评估重要节点数目对 HGKSum 性能的影响。为了探索重要词语节点数目 k 和重要句子节点数目 l 的影响，固定任务权重 λ 为 0.7，将 k 和 l 分别表示为γm 和γn，其中 m 和 n 分别为词语节点数目和句子节点数目，设置γ为 0～1 进行实验，实验结果如表 1 所示。当γ=0，即不考虑重要节点，模型的效果最差；当γ=0.7，模型的效果最好，两种情形在 ROUGE-1、ROUGE-2、ROUGE-L 的差距分别为 1.49、1.31 和 1.56。为了更清楚地观察重要节点数目与模型性能之间的关系，进一步地绘制了重要节点数目与平均的 ROUGE 指标的变化曲线，如图 2 所示。结合图 2 与表 1，可以发现，随着重要节点数目的增加，模型的性能先快速上升，当$\gamma\geqslant0.3$，模型性能缓慢上升；当在$\gamma=0.7$时，模型性能最好而后缓慢下降。主要原因在于重要节点数目过少会导致异构节点特征之间的学习机会变少，在$\gamma=0$的极端情况下模型完全不进行异构节点特征之间的学习，严重制约模型的表达能力；而重要节点数目过多则会引入大量噪声节点，在$\gamma=1$的极端情况下所有节点包括噪声节点均被视作重要节点，同样会干扰节点特征的学习过程。

表 1 重要节点数目的影响

指标	γ=0.0	γ=0.1	γ=0.2	γ=0.3	γ=0.4	γ=0.5	γ=0.6	γ=0.7	γ=0.8	γ=0.9	γ=1.0
R-1	41.26	42.08	41.91	42.41	42.54	42.59	42.71	**42.75**	42.78	42.67	42.60
R-2	18.33	19.06	18.98	19.27	19.43	19.39	19.58	**19.64**	19.62	19.40	19.52
R-L	37.57	38.39	38.31	38.73	38.86	38.87	39.03	**39.13**	39.12	39.03	38.92

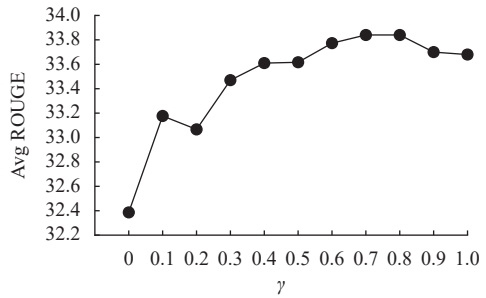

图 2 重要节点数目与模型性能之间的关系

其次，探索任务权重 λ 对 HGKSum 性能的影响，将重要节点数目 k 和 l 分别固定为 0.7m 和 0.7n，λ 设置为 0.5～1 进行实验。之所以仅选择 $\lambda\geqslant0.5$ 来进行实验，是因为本文将摘要任务定义为主任务，其权重理应最大。实验结果如表 2 所示。当 λ=0.7 时，模型效果最好；当 λ=1 时，模型效果最差，二者在 ROUGE-1、ROUGE-2 和 ROUGE-L 上分别相差 0.51、0.49 和 0.61。主要原因在于当 λ=1 时，完全不考虑关键词抽取任务，多任务学习退化为单任务学习，最终使得模型性能下降。与实验 1 类似，同样绘制了任务权重与平均 ROUGE 指标的变化曲线，如图 3 所示。由图 3 和表 2 可以发现，在不考虑 λ=1 的极端情况时，随着任务权重 λ 的

增加，模型性能的起伏不大。具体来说，ROUGE-1、ROUGE-2 和 ROUGE-L 的极差分别为 0.02、0.09 和 0.1，表明模型对于该参数不敏感，有鲁棒性。

表 2　任务权重的影响

指标	λ=0.5	λ=0.6	λ=0.7	λ=0.8	λ=0.9	λ=1.0
R-1	42.76	42.75	**42.75**	42.74	42.74	42.24
R-2	19.58	19.55	**19.64**	19.57	19.59	19.15
R-L	39.10	39.08	**39.13**	39.03	39.13	38.52

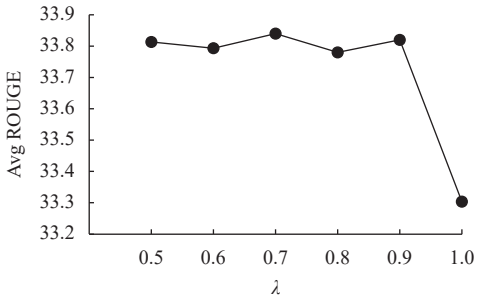

图 3　任务权重对于模型性能的影响

3.5　整体评价和对比分析

为了验证使用 HGKSum 进行摘要的有效性，选择 8 种基准模型与 HGKSum 进行对比实验。需要说明的是，受限于计算资源，HGKSum 并未引入大规模预训练模型，为了确保公平性，选择的基准模型同样不是基于大规模预训练模型的抽取式文本摘要模型。基准模型包括：

1） ORACLE[14] 表示选择数据集的标签作为摘要，可以在一定程度上看作是抽取式文本摘要在此数据集上的性能上限；

2） LEAD-3[36] 模型简单地选择文章开头的几个句子作为摘要；

3） REFRESH[39] 将抽取式文本摘要任务视作句子排序任务，使用强化学习的方式生成摘要；

4） BanditSum[40] 将抽取式文本摘要任务视作上下文老虎机（contextual bandit）问题，使用强化学习的方式生成摘要；

5） NeuSUM[41] 将抽取式文本摘要中的句子评分和句子选择整合为一个步骤进行整体学习；

6） PNBERT[42] 使用 BERT 作为编码器，指针网络作为解码器来生成摘要；

7） MatchSum[13] 将抽取式文本摘要任务视作语义匹配问题，使用孪生 BERT 网络选择最佳的候选摘要成为最终摘要；

8） HSG[43] 与本文模型类似，也是基于异构图的模型，主要不同点在于该模型没有引入关键词信息，也没有进行同构节点之间的特征学习。

实验结果如表 3 所示。表 3 中模型可以分为 3 类，分别是无监督的模型（LEAD-3）、基于 BERT 的深度学习模型（PNBERT、MatchSum）和不基于 BERT 的深度学习模型（REFRESH、BanditSum、NeuSUM、HSG 和 HGKSum），可以得出如下观测结果。

1）与 ORACLE 相比，所有模型的性能仍有较大提升空间，这说明抽取式文本摘要任务远未达到研究瓶颈，尚有广阔的探索空间。

2）与基于 BERT 的深度学习基准模型相比，HGKSum 的性能强于 PNBERT，表明 HGKSum 在不依赖预训练语言模型的情形下有较好的性能表现。

3）HGKSum 的性能弱于 MatchSum，原因在于 MatchSum 模型复杂度高于 HGKSum，MatchSum 有 1.7 亿参数，而 HGKSum 仅有 360 万个。计算方式如下：假设输入特征维度数为 H，从模型参数量角度进行对比，HGKSum 的参数量可近似表示为$4(2H^2+H+8H^2)$，其中 4 表示进行了 4 次节点特征学习，$2H^2$是变换矩阵 $\boldsymbol{W}$ 的参数量，H 是注意力向量 $\boldsymbol{a}$ 的参数量，$8H^2$是前馈神经网络的参数量；而 MatchSum 的参数量可近似表示为$2\times12(H^2+3H^2+8H^2)$，其中 2 表示孪生 BERT 网络中包含两个 BERT 网络，12 表示一个 BERT 网络堆叠了 12 层 Transformer 块，H^2是变换矩阵的参数量，$3H^2$是 3 个注意力矩阵 QKV 的参数量，$8H^2$是前馈神经网络的参数量。特征维度 H 在 HGKSum 中取值为 300，在 MatchSum 中取值为 768。

4）与无监督的基准模型和不采用基于 BERT 的基准模型相比，HSG 模型是表现最好的，而与 HSG 相比，HGKSum 在 ROUGE-1、ROUGE-2 和 ROUGE-L 上分别提升了 0.72、0.20 和 0.62，表明 HGKSum 使用异构图建模文本以及引入关键词信息的有效性。

表 3　HGKSum 与基准模型对比结果

模型	R-1	R-2	R-L
ORACLE	52.59	31.24	48.87
LEAD-3	40.34	17.70	36.57
REFRESH	40.00	18.20	36.60
BanditSum	41.50	18.70	37.60
NeuSUM	41.59	19.01	37.98
PNBERT	42.69	19.60	38.85
MatchSum	**44.41**	**20.86**	**40.55**
HSG	42.03	19.44	38.51
HGKSum	42.75	19.64	39.13

3.6 消融实验

为了探索 HGKSum 模型中不同组件对摘要的贡献程度，本文设计了以下消融实验。首先，在文本向量化阶段，实验移除了预训练的 GloVe 词向量，采用随机初始化的方式初始化词向量，记为 M1。其次，在异构图构建过程中，实验不再进行重要节点的筛选，即所有的节点均设为重要节点，记为 M2。再次，在节点特征学习过程中，实验分别移除了异构节点特征之间的学习和同构节点特征之间的学习过程，分别记为 M3 和 M4，在训练阶段，实验移除了多任务学习中的关键词抽取任务，采用单任务学习的方式进行训练，记为 M5。实验还移除摘要精炼模块，直接使用得分最高的 3 个句子构成摘要，记为 M6。最后，为了验证节点学习次序的正确性，实验选择先进行异构节点之间的学习，再进行同构节点之间的学习，记为 M7，之后，实验选择先进行词–句节点特征学习，再进行句–词节点特征学习，记为 M8。

实验结果如表 4 所示。值得说明的是，不进行重要节点筛选（M2）和不进行异构节点特征学习（M3）实际上分别对应了实验 1 中的 γ 设置为 1 和 0，同样地，不进行关键词抽取任务（M5）实际上对应了实验 2 中的 λ 设置为 1。观察表 4，可以发现不论移除哪一组件，HGKSum 的性能都有所下降，说明了所有组件均具有正向效用，同时也验证了前面提及的节点学习次序的正确性。为了进一步探索哪一组件贡献最大，本文绘制了柱状图来表示不同消融模型对应组件的贡献，横坐标为消融模型，纵坐标为 HGKSum 与消融模型在指标上的差值 ΔROUGE，如图 4 所示。由图 4 可以清楚地发现，M3 对应的组件（异构节点特征之间的学习）贡献最大，其在 ROUGE-1、ROUGE-2 和 ROUGE-L 上分别贡献了 1.49、1.31 和 1.56，这说明文本摘要关注的句子关系更多地为复杂的跨句关系，难以通过简单的同构节点来学习，体现了 HGKSum 引入异构节点的必要性。

表 4 HGKSum 上的消融实验

模型	R-1	R-2	R-L
HGKSum	42.75	19.64	39.13
M1	42.12	19.04	38.43
M2	42.60	19.52	38.92
M3	**41.26**	**18.33**	**37.57**
M4	42.51	19.40	38.81
M5	42.24	19.15	38.52
M6	41.65	19.08	38.07
M7	42.30	19.20	38.58
M8	42.46	19.35	38.74

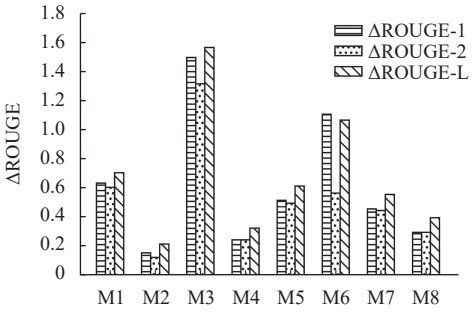

图 4 HGKSum 中不同组件的贡献

4 结束语

本文提出了一种基于异构图和关键词的抽取式文本摘要模型，构建的异构图能够很好地捕获复杂的句子关系，引入的关键词信息通过多任务学习的方式能够有效地指导摘要的抽取。在基准数据集上的实验表明，本文模型优于不依赖预训练模型的摘要模型。为了获取更好的文本表示，下一步的工作拟探索预训练模型在本模型中的应用。

参 考 文 献

[1] 李金鹏, 张闯, 陈小军, 等. 自动文本摘要研究综述[J]. 计算机研究与发展, 2021, 58(1): 1-21.
LI J P, ZHANG C, CHEN X J, et al. Survey on automatic text summarization[J]. Journal of Computer Research and Development, 2021, 58(1): 1-21.

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

[3] RUSH A M, CHOPRA S, WESTON J. A neural attention model for abstractive sentence summarization[C]//Proc of the 2015 Conf on Empirical Methods in Natural Language Processing. Stroudsburg: ACL, 2015: 379-389.

[4] EDMUNDSON H P. New methods in automatic extracting[J]. Journal of the ACM, 1969, 16(2): 264-285.

[5] MIHALCEA R, TARAU P. TextRank: Bringing order into text[C]//Proc of the 2004 Conf on Empirical Methods in Natural Language Processing. Stroudsburg: ACL, 2004: 404-411.

[6] PAGE L, BRIN S, MOTWANI R, et al. The PageRank citation ranking: Bringing order to the web[R]. Palo Alto: Stanford InfoLab, 1999.

[7] YAN S, WAN X J. SRRank: Leveraging semantic roles for extractive multi-document summarization[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2014, 22(12): 2048-2058.

[8] ZHENG H, LAPATA M. Sentence centrality revisited for unsupervised summarization[C]//Proc of the 57th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: ACL, 2019: 6236-6247.

[9] NALLAPATI R, ZHAI F F, ZHOU B W. SummaRuNNer:

A recurrent neural network based sequence model for extractive summarization of documents[C]//Proc of the 31st AAAI Conf on Artificial Intelligence. Palo Alto: AAAI, 2017: 3075-3081.

[10] YAO K C, ZHANG L B, LUO T J, et al. Deep reinforcement learning for extractive document summarization[J]. Neurocomputing, 2018, 284: 52-62.

[11] 韩鹏宇, 余正涛, 高盛祥, 等. 案件要素句子关联图卷积的案件與情摘要方法[J]. 软件学报, 2021, 32(12): 3829-3838.
HAN P Y, YU Z T, GAO S X, et al. Case-related public opinion summarization method based on graph convolution of sentence association graph with case elements[J]. Journal of Software, 2021, 32(12): 3829-3838.

[12] DEVLIN J, CHANG M W, LEE K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding[C]//Proc of the 2019 Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Stroudsburg: ACL, 2019: 4171-4186.

[13] ZHONG M, LIU P F, CHEN Y R, et al. Extractive summarization as text matching[C]//Proc of the 58th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: ACL, 2020: 6197-6208.

[14] LIU Y, LAPATA M. Text summarization with pretrained encoders[C]//Proc of the 2019 Conf on Empirical Methods in Natural Language Processing and the 9th Int Joint Conf on Natural Language Processing (EMNLP-IJCNLP). Stroudsburg: ACL, 2019: 3730-3740.

[15] PASQUIER C. Single document keyphrase extraction using sentence clustering and latent Dirichlet allocation[C]//Proc of the 5th Int Workshop on Semantic Evaluation. Stroudsburg: ACL, 2010: 154-157.

[16] CAMPOS R, MANGARAVITE V, PASQUALI A, et al. YAKE! Keyword extraction from single documents using multiple local features[J]. Information Sciences, 2020, 509: 257-289.

[17] SINGHAL A, SHARMA D K. Keyword extraction using Renyi entropy: A statistical and domain independent method[C]//2021 7th Int Conf on Advanced Computing and Communication Systems (ICACCS). Piscataway, NJ: IEEE, 2021: 1970-1975.

[18] DUARI S, BHATNAGAR V. sCAKE: Semantic connectivity aware keyword extraction[J]. Information Sciences, 2019, 477: 100-117.

[19] TIXIER A, MALLIAROS F, VAZIRGIANNIS M. A graph degeneracy-based approach to keyword extraction[C]//Proc of the 2016 Conf on Empirical Methods in Natural Language Processing. Stroudsburg: ACL, 2016: 1860-1870.

[20] JAIN A, MITTAL K, VAISLA K S. FLAKE: Fuzzy graph centrality-based automatic keyword extraction[J]. The Computer Journal, 2022, 65(4): 926-939.

[21] GOZ F, MUTLU A. MGRank: A keyword extraction system based on multigraph GoW model and novel edge weighting procedure[J]. Knowledge-Based Systems, 2022, 251: 109292.

[22] PENNINGTON J, SOCHER R, MANNING C D. GloVe: Global vectors for word representation[C]//Proc of the 2014 Conf on Empirical Methods in Natural Language Processing (EMNLP). Stroudsburg: ACL, 2014: 1532-1543.

[23] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[24] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735-1780.

[25] VELIČKOVIĆ P, CUCURULL G, CASANOVA A, et al. Graph attention networks[EB/OL]. [2023-01-03]. https://arxiv.org/abs/1710.10903.

[26] MAAS A L, HANNUN A Y, NG A Y. Rectifier nonlinearities improve neural network acoustic models[C]//Proc of the 30th Int Conf on Machine Learning. Brookline, MA: Microtome Publishing, 2013: 2104-2109.

[27] BRODY S, ALON U, YAHAV E. How attentive are graph attention networks?[EB/OL]. [2023-01-03]. https://arxiv.org/abs/2105.14491.

[28] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Proc of the 31st Int Conf on Neural Information Processing Systems. New York: Curran Associates Inc, 2017: 6000-6010.

[29] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition[C]//Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society, 2016: 770-778.

[30] BA J L, KIROS J R, HINTON G E. Layer normalization[EB/OL]. [2023-01-05]. https://arxiv.org/abs/1607.06450.

[31] NAIR V, HINTON G E. Rectified linear units improve restricted Boltzmann machines[C]//Proc of the 27th Int Conf on Machine Learning. Madison, WI: Omnipress, 2010: 807-814.

[32] PAULUS R, XIONG C M, SOCHER R. A deep reinforced model for abstractive summarization[EB/OL]. [2023-01-03]. https://arxiv.org/abs/1705.04304.

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

[34] HERMANN K M, KOČISKÝ T, GREFENSTETTE E, et al. Teaching machines to read and comprehend[C]//Proc of the 28th Int Conf on Neural Information Processing Systems-Volume 1. Cambridge: MIT Press, 2015: 1693-1701.

[35] NALLAPATI R, ZHOU B W, SANTOS C, et al. Abstractive text summarization using sequence-to-

sequence RNNs and beyond[C]//Proc of the 20th SIGNLL Conf on Computational Natural Language Learning. Stroudsburg: ACL, 2016: 280-290.

[36] SEE A, LIU P J, MANNING C D. Get to the point: Summarization with pointer-generator networks[C]//Proc of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Stroudsburg: ACL, 2017: 1073-1083.

[37] LIN C Y, HOVY E. Automatic evaluation of summaries using n-gram co-occurrence statistics[C]//Proc of the 2003 Conf of the North American Chapter of the Association for Computational Linguistics on Human Language Technology. Stroudsburg: ACL, 2003: 71-78.

[38] KINGMA D P, BA J. Adam: A method for stochastic optimization[EB/OL]. [2023-01-03]. https://arxiv.org/abs/1412.6980.

[39] NARAYAN S, COHEN S B, LAPATA M. Ranking sentences for extractive summarization with reinforcement learning[C]//Proc of the 2018 Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg: ACL, 2018: 1747-1759.

[40] DONG Y, SHEN Y K, CRAWFORD E, et al. BanditSum: Extractive summarization as a contextual bandit[C]//Proc of the 2018 Conf on Empirical Methods in Natural Language Processing. Stroudsburg: ACL, 2018: 3739-3748.

[41] ZHOU Q Y, YANG N, WEI F R, et al. Neural document summarization by jointly learning to score and select sentences[C]//Proc of the 56th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: ACL, 2018: 654-663.

[42] ZHONG M, LIU P F, WANG D Q, et al. Searching for effective neural extractive summarization: What works and what's next[C]//Proc of the 57th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: ACL, 2019: 1049-1058.

[43] WANG D Q, LIU P F, ZHENG Y N, et al. Heterogeneous graph neural networks for extractive document summarization[C]//Proc of the 58th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: ACL, 2020: 6209-6219.

编 辑 税 红