结合邻域知识的文档级关键词抽取方法

李晨亮[1,2*]，龙俊辉[1,2]，唐作立[1,2]，周　涛[1,2]

(1. 武汉大学空天信息安全与可信计算教育部重点实验室　武汉　430072；2. 武汉大学国家网络安全学院　武汉　430072)

【摘要】基于编码器–解码器 (encoder-decoder) 框架的生成式方法在关键词抽取任务上得到了广泛应用并取得了较好的性能，然而该方法面临的主要挑战为建模有效的文档向量表示，及生成覆盖整个文档主题的关键词集合，这些挑战都会直接影响关键词抽取的结果。该文提出了结合邻域知识的文档级关键词抽取模型以应对这些挑战。具体来说，通过给指定文档添加少量的最近邻样本，原文档被扩展为一个文档集合。基于单词之间的距离将文档集合中的每个文档构建成词图，合并集合中的所有词图形成一个大图，然后利用图卷积网络进行编码。解码端引入了上下文修改机制和覆盖机制，使模型能够生成更加多样化的关键词来覆盖文档包含的所有主题。最后在 4 种数据集上分别与现有的基准模型进行对比，实验结果表明该方法能够有效提升关键词抽取的性能。

关　键　词　深度学习；　编码器–解码器框架；　图卷积网络；　图神经网络；　关键词抽取；　邻域知识

中图分类号　TP391　　**文献标志码**　A　　doi:10.12178/1001-0548.2021095

Document-level Keyphrase Extraction Approach using Neighborhood Knowledge

LI Chen-liang[1,2*], LONG Jun-hui[1,2], TANG Zuo-li[1,2], and ZHOU Tao[1,2]

(1. Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, Wuhan University　Wuhan　430072;
2. School of Cyber Science and Engineering, Wuhan University　Wuhan　430072)

Abstract　Encoder-decoder-based generative approaches have been widely used and achieved good performance for keyphrase extraction tasks. However, the main challenges of the encoder-decoder-based approach are modeling an effective document vector representation and generating a set of keyphrases covering the entire document topic, which can directly affect the keyphrase extraction results. In this paper, a document-level keyphrase extraction model incorporating neighborhood knowledge is proposed to address the challenges mentioned above simultaneously. Specifically, the original document is extending to a document set by adding some nearest-neighbor documents. Then, each document in the set is constructed into a word graph based on the distance between words, and all the word graphs in the set are merged into a large graph, which is then encoded using graph convolutional networks. Besides, to fully cover diverse keyphrases and topics, the context modification mechanism and coverage mechanism are introduced at the decoding step. Finally, by comparing with the existing baseline model on four benchmark datasets, the experimental results show that the method can effectively improve the performance of extracting keyphrases.

Key words　deep learning;　encoder-decoder framework;　graph convolutional network;　graph neural networks;　keyphrase extraction;　neighborhood knowledge

关键词抽取技术是一种从文本中抽取主题和一些重要短语的技术，可以帮助阅读者快速了解文本中最有价值的信息[1]。一篇文档的关键词通常是几个单词或者短语，可以作为该文档主要内容的提要。近几十年来，自动抽取关键词的任务受到了广泛关注[2]。另外，由于关键词具有表达简洁、精准的特点，被广泛用于自然语言处理领域中的下游任务，比如文本摘要、情感分析以及文本聚类等[3]。

最传统的关键词抽取方法基于 TF-IDF，它能识别出在当前文档中频繁出现，但在整个文档集合中不会频繁出现的单词，即该文档的重点单词。还有基于图的排序方法[4]，它首先从文档构造一个词

收稿日期：2021－03－28；修回日期：2021－05－06
基金项目：国家自然科学基金面上项目 (61872278)
作者简介：李晨亮 (1983－)，男，博士，教授，主要从事信息检索、数据挖掘、机器学习、社交网络分析及自然语言处理方面的研究. E-mail：cllee@whu.edu.cn

图，然后用基于随机游走的方法 (如 PageRank) 确定关键词的重要性。通过构建词图，这些方法可以有效地识别最显著的关键词。近年来，基于端到端神经网络的关键词抽取方法受到越来越多的关注。文献 [2] 将关键词抽取视为一项序列标注任务，并提出了一种联合层 RNN 的模型。文献 [5] 首先提出了一种用于关键词抽取的编码器–解码器框架，并在框架中引入了注意力机制，使模型基于输入的文档来输出对应的关键词。文献 [6] 进一步提出了一个基于 CNN 的模型来完成这项任务。文献 [7] 提出了一种标题指导的 seq2seq 网络来加强隐式文档表示。另外，为了考虑生成的关键词之间的关系，文献 [8] 使用了自适应奖励的强化学习方法。文献 [9] 指出了关键词生成中第一个词的重要性。文献 [10] 利用了聚类的方法来过滤掉无意义的候选关键词。

基于端到端的神经网络方法还面临两个重要挑战：1) 建模有效的文档向量表示。文档向量表示能够直接决定解码器的输出结果，影响整个模型的效果；2) 生成全面并准确覆盖整个输入文档主题的关键词集合。为了解决以上问题，本文提出了结合邻域知识的文档级关键词抽取方法。在编码端给定一个文档，基于该文档与相似文档的相关信息以及文档内部单词之间的距离构建出文档的图结构，将相同的单词聚合到一起用同一个节点表示。在解码端设计了一种多样化指针网络，从原文档的词图中动态地选择相关的单词构成强概括性且不重复的关键词，极大优化了解码器端的性能。实验结果表明，相比传统非监督学习方法和经典的深度学习方法，结合邻域知识的文档级关键词抽取模型可以有效提升抽取结果的相关指标。

1 研究方法

本文模型由检索器、编码器及解码器组成。编码器采用了图卷积神经网络 (graph convolutional networks, GCN)[11] 进行编码，解码器以关键词为单位逐个生成关键词。编码器和解码器的结构示意图分别如图 1 和图 2 所示。

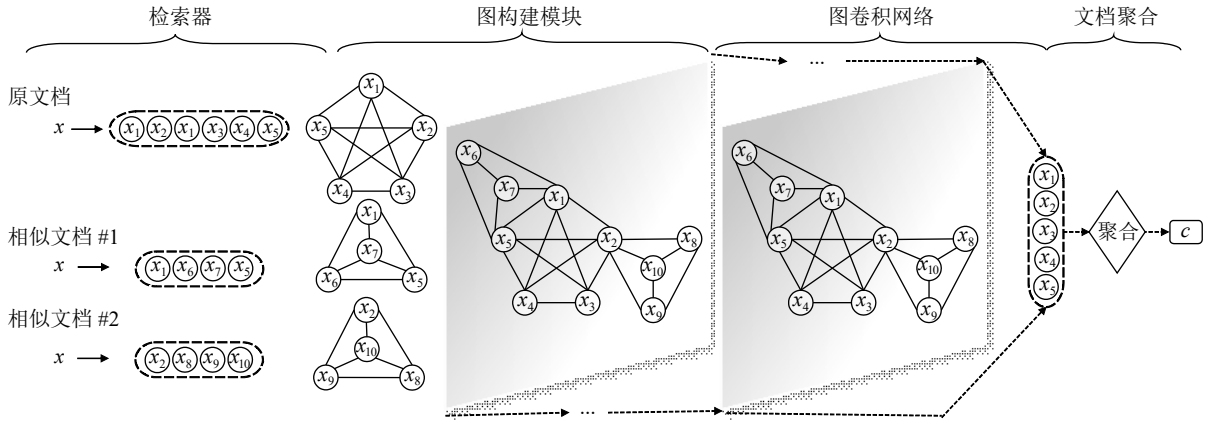

图 1 编码器结构的示意图

1.1 检索器

输入指定文档，找到与该文档相似的邻域文档，邻域文档表示在局部空间中与指定文档靠近的文档。邻域文档可以给原文档提供额外的领域知识上下文信息。即给定文档 x，利用相似文档检索技术检索出 Top-K 个邻域文档，原文档 x 被扩充为文档集合 $\chi=\{x,x_1,\cdots,x_k\}$，作为输入文档的增强上下文知识数据。相似文档检索技术的核心依赖于两个文档相似度的评估功能。本文使用余弦相似度方法来度量文档 x_i 和文档 x_j 的相似度，并引入 TF-IDF 指标来度量文档中单词的重要性，最后计算出基于加权的文档词项向量的余弦相似度分数，其计算公式为：

$$\mathrm{sim}_{\mathrm{doc}}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\frac{\boldsymbol{x}_i\cdot\boldsymbol{x}_j}{|\boldsymbol{x}_i|\times|\boldsymbol{x}_j|} \tag{1}$$

1.2 编码器

1.2.1 图构建

不同于传统的基于图排序的方法如 TextRank 和其他图卷积神经网络方法如 GAT[12]。模型采用一个有向完全图来表示文档的结构关系[13]，把文档中多次出现的同一个词用唯一节点表示，然后利用节点在文档中的结构关系强度，加权得到节点之间的边。通常，设 $G=(V,E)$ 为有向完全图，其中

V 和 E 分别表示节点集合和边集合，邻接矩阵表示为$\boldsymbol{A} \in \{\overleftarrow{\boldsymbol{A}}, \overrightarrow{\boldsymbol{A}}\}$。假设文档中的单词经常出现在一起即两个单词彼此越靠近，关系就越强，具有更强的语义关系。该方法扩展了传统的邻接矩阵，使其具有更丰富灵活的节点关系信息。邻接矩阵$\boldsymbol{A} \in \{\overleftarrow{\boldsymbol{A}}, \overrightarrow{\boldsymbol{A}}\}$中的两个词节点$w_i$和$w_j$的边权重大小定义为：

$$\overleftarrow{A}_{ij} = \sum_{p_i \in \mathcal{P}(w_i)} \sum_{p_j \in \mathcal{P}(w_j)} \mathrm{relu}\left(\frac{1}{p_i - p_j}\right) \tag{2}$$

$$\overrightarrow{A}_{ij} = \sum_{p_i \in \mathcal{P}(w_i)} \sum_{p_j \in \mathcal{P}(w_j)} \mathrm{relu}\left(\frac{1}{p_j - p_i}\right) \tag{3}$$

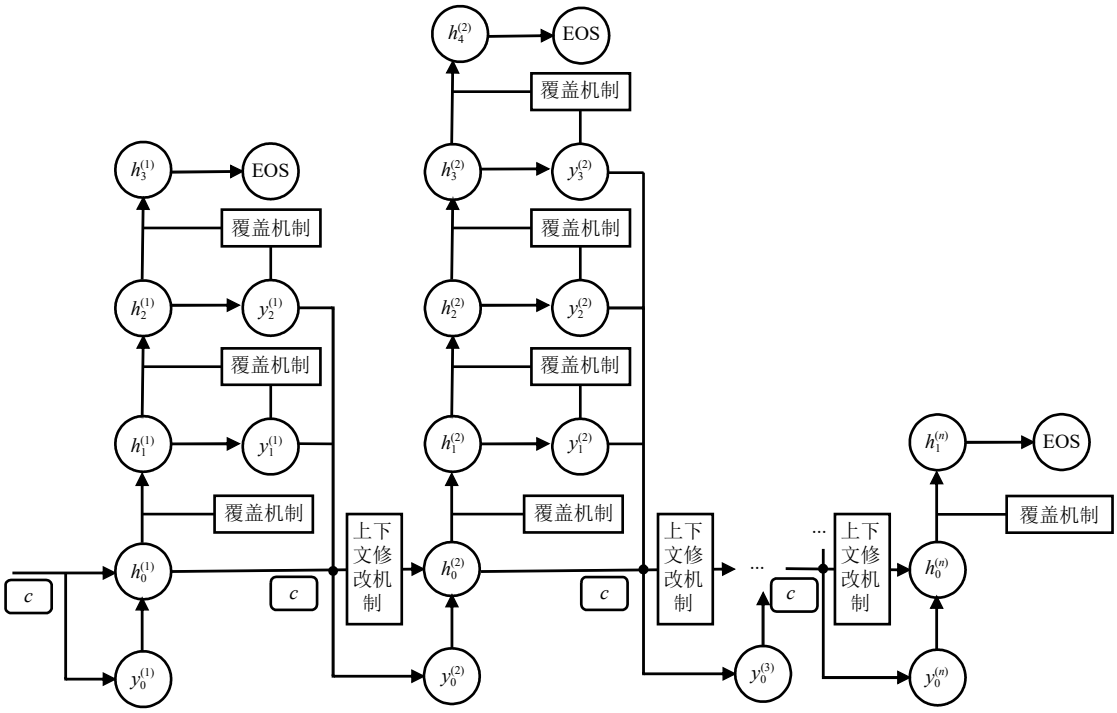

图 2　解码器结构的示意图

式中，$\mathcal{P}(w_i)$表示文档中单词 w_i 的位置偏移量集合，ReLU 激活函数的目的是选择出有效的单向信息，帮助编码器识别出节点序列的顺序信息。完全图的特性确保文档中单词之间都是相互关联的，因此给每个单词都提供了全局视角的上下文信息。为了使模型在迭代过程中信息传输更加稳定有效，将邻接矩阵$\boldsymbol{A}$的对角元素设置为 1，给每个节点添加了自环，即$\hat{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$。然后引入度矩阵$\hat{\boldsymbol{D}}$，其中$\hat{D}_{ii} = \sum_j \hat{A}_{ij}$，最后计算$\tilde{\boldsymbol{A}} = \hat{\boldsymbol{D}}^{-1/2}\hat{\boldsymbol{A}}\hat{\boldsymbol{D}}^{-1/2}$，得到归一化的邻接矩阵$\tilde{\boldsymbol{A}}$。

1.2.2　图卷积神经网络

图卷积神经网络 (GCN) 是一个直接对图进行操作的多层神经网络，能够根据图内部节点中邻居节点的属性来归纳学习当前节点的嵌入向量表示。图卷积神经网络中一个卷积层能够捕获到每个节点的直接邻居信息，如果堆叠多层卷积网络就能够聚集到更大范围的邻域节点信息。因此一旦完成了文档图构建操作，便把整个图输入到一个多层图卷积神经网络中。每个卷积层通常包含两个阶段，第一个阶段聚合每个节点的所有邻居节点信息，在第二个阶段中，每个节点根据当前节点表示和聚合信息表示来更新自身的节点表示，两个阶段共同实现了一次完整的信息传输和聚合流程。给定节点矩阵表示$\boldsymbol{H}_l$，l 表示当前卷积层的下标，图中所有节点的聚合和更新过程用函数$f_l(\boldsymbol{H}_l)$表示，则有：

$$f_l(\boldsymbol{H}_l) = \overleftarrow{\tilde{\boldsymbol{A}}}\boldsymbol{H}_l\overleftarrow{\boldsymbol{W}}_l + \overrightarrow{\tilde{\boldsymbol{A}}}\boldsymbol{H}_l\overrightarrow{\boldsymbol{W}}_l + \boldsymbol{H}_l\boldsymbol{W}_l \tag{4}$$

式中，$\overleftarrow{\boldsymbol{W}}_l$、$\overrightarrow{\boldsymbol{W}}_l$、$\boldsymbol{W}_l$为第 l 层的可训练参数，两个有向矩阵$\overleftarrow{\boldsymbol{W}}_l$、$\overrightarrow{\boldsymbol{W}}_l$捕获了图中双向序列信息。因此通过两个有向邻接矩阵聚合了每个节点中具有序列属性的邻域信息。节点表示更新完成之后，受到文献 [14] 的启发，引入卷积网络门控单元和残差学习框架，其中卷积网络门控单元用公式$h_l(\boldsymbol{X}) = (\boldsymbol{X} * \boldsymbol{W} + \boldsymbol{b}) \otimes \sigma(\boldsymbol{X} * \boldsymbol{V} + \boldsymbol{c})$表示，$\sigma$表示 sigmoid函数，$\otimes$表示矩阵中的点乘操作。结合残差学习框架后，扩展了

节点表示的更新结果，即：

$$\boldsymbol{H}_{l+1} = \boldsymbol{H}_l + f_l(\boldsymbol{H}_l) \otimes \sigma(g_l(\boldsymbol{H}_l)) \tag{5}$$

门控单元内部的函数g_l具有和f_l相似的网络结构。$\boldsymbol{H}_0$初始化为单词的嵌入表示，作为模型的初始输入。

1.2.3 文档聚合

针对编码器端最后一层 H_L 中属于原文档的节点，模型采用多头自注意力机制[15]线性转换原文档中出现的单词表示，计算均值并拼接得到文档的向量表示 $\boldsymbol{c}$，则有：

$$\boldsymbol{c} = \mathop{\|}_{t=1}^{T} \overline{\mathrm{Attention}(Q,K,V)H_L} \tag{6}$$

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{7}$$

实验中设置多头的数量为 4，即 $T=4$。

1.3 解码器

解码器是该方法的重要组成部分，文档通过编码器得到文档的向量表示。基于文档表示，解码器将连续生成多个具有多样性且完整覆盖文档主题的多个关键词。本节将详细介绍用指针网络来实现解码器的方法，并说明引入的上下文修改机制以及覆盖机制的原理和作用。

1.3.1 指针网络

指针网络[16]是一种学习输出序列的条件概率的网络结构，能够解决可变大小输入字典的问题。因此输入文档表示，解码器就能从输入文档的字典范围中生成多个高质量关键词，满足关键词抽取任务的要求。和大多数 seq2seq 框架一样，解码器首先使用 RNN 框架来依次输出每一时刻t的网络隐藏状态，则有：

$$\boldsymbol{h}_t^{(i)} = \mathrm{RNN}\left(\boldsymbol{y}_{t-1}^{(i)}, \boldsymbol{h}_{t-1}^{(i)}\right) \tag{8}$$

式中，i 和 t 分别表示第 i 个关键词和第 i 个关键词中的第 t 个单词。用文档表示 $\boldsymbol{c}$ 来表示$\boldsymbol{h}_0^{(0)}$作为解码器的初始输入。然而在解码的过程中$\boldsymbol{h}_0^{(i)}$应该会随着关键词 i 动态变化，下个小节将重点解决这个问题。另外，$\boldsymbol{y}_{t-1}^{(i)}$表示上一时刻预测出单词$y_{t-1}^{(i)}$的节点表示，$\boldsymbol{h}_{t-1}^{(i)}$表示上一时刻 RNN 的隐藏状态。因此利用$t$时刻 RNN 的隐藏状态$\boldsymbol{h}_t^{(i)}$，原文档中出现过的单词节点表示$\boldsymbol{x}_j$，以及指针网络来选取单词$y_t^{(i)}$：

$$e_{t,j}^{(i)} = \boldsymbol{v}^{\mathrm{T}} \tanh\left(\boldsymbol{W}_h \boldsymbol{h}_t^{(i)} + \boldsymbol{W}_x \boldsymbol{x}_j + \boldsymbol{b}\right) \tag{9}$$

$$P\left(y_t^{(i)}\right) = \mathrm{softmax}\left(\boldsymbol{e}_t^{(i)}\right) \tag{10}$$

指针网络使用注意力机制来计算每个单词节点 x_j 的分数，$j \leqslant |V|$表示选择单词的范围只局限在原文档中。然后使用 softmax 函数归一化向量 $\boldsymbol{e}_t^{(i)} \in \mathcal{R}^{|V|}$，输出单词节点的概率分布。其中，$\boldsymbol{v}^{\mathrm{T}}$、$\boldsymbol{W}_h$、$\boldsymbol{W}_x$、$\boldsymbol{b}$是模型中的可学习参数。选择概率分布中具有最大概率的单词节点作为模型t时刻生成的单词$y_t^{(i)}$。

1.3.2 上下文修改机制

文档表示 $\boldsymbol{c}$ 作为解码器端的唯一输入，对解码器的生成效果有至关重要的影响。如果文档表示始终保持不变，将会导致模型生成重复的关键词，并使模型生成的候选关键词集合的语义变得复杂冗余。因此，当模型生成第i个关键词之前，本文引入了上下文修改机制[13]动态地修改上下文表示$\boldsymbol{h}_0^{(i)}$：

$$\boldsymbol{y}_0^{(i)} = \boldsymbol{W}_y\left[\boldsymbol{c}, \overline{\boldsymbol{y}^{(1:i-1)}}\right] + \boldsymbol{b}_y \tag{11}$$

$$\boldsymbol{h}_0^{(i)} = \tanh\left(\boldsymbol{W}_h\left[\boldsymbol{c}, \overline{\boldsymbol{y}^{(1:i-1)}}\right] + \boldsymbol{b}_h\right) \tag{12}$$

式中，$\overline{\boldsymbol{y}^{(1:i-1)}}$表示过去生成的所有关键词表示的均值，初始化$\overline{\boldsymbol{y}^{(\cdot)}}$为向量 $\boldsymbol{0}$。因此，上下文修改机制能够促使模型生成多个拥有丰富语义的关键词来覆盖文档的所有主题，解决模型过度生成的问题。

1.3.3 覆盖机制

在关键词抽取任务中，多个关键词通常对应原文中不同的位置，已经被模型选择过的单词应该尽量避免多次被模型选中。为了避免这个问题，模型引入了覆盖机制[17]，灵活调节每个关键词的概率分布，确保原文中所有的重要区域都能够被模型重点关注并且被总结为关键词。

具体来说，该模型维护一个词汇级别的覆盖向量 $\boldsymbol{c}^{(i)}$[13]，首先记录所有单词的独热 (one-hot) 编码表示，即解码器生成单词的对应节点位置记为 1，其余位置为 0，然后计算过去生成关键词的独热编码总和来得到当前时刻的$\boldsymbol{c}_j^{(i)}$。直观而言可以把$\boldsymbol{c}^{(i)}$看作是目前为止原文档中所有单词从模型得到的一种关注程度，即覆盖程度。$\boldsymbol{c}_j^{(i)}$计算方式为：

$$\boldsymbol{c}_j^{(i)} = \boldsymbol{c}_j^{(i-1)} + \sum_t \boldsymbol{o}_{t,j}^{(i-1)} \tag{13}$$

式中，$\boldsymbol{c}_j^{(i)}$表示模型生成到第 i 个关键词为止，第 j 个候选单词的覆盖程度；$\sum_t \boldsymbol{o}_{t,j}^{(i-1)}$表示上一时刻生成的关键词中，第 j 个单词的覆盖值大小。最后更新指针网络为：

$$e_{t,j}^{(i)} = \boldsymbol{v}^{\mathrm{T}} \tanh\left(\boldsymbol{W}_h \boldsymbol{h}_t^{(i)} + \boldsymbol{W}_x \boldsymbol{x}_j + \boldsymbol{W}_c \boldsymbol{c}_j^{i} + \boldsymbol{b}\right) \tag{14}$$

式中，新增的$\boldsymbol{W}_c$是可学习参数。

1.4　训练

给定一个数据样本，其中包括原文本 x 和对应的关键词集合$\mathcal{P}=\{p_i\}_{i=0}^{M}$，模型的目标是最小化目标单词的负对数似然函数，损失函数的计算方法如下：

$$\text{loss}=-\frac{1}{M}\sum_{i=0}^{M}\sum_{t=0}^{T_i}\log\left(p\left(y_t^i\mid y_{1:t-1}^i,y^{(1:i-1)},\boldsymbol{c}\right)\right) \tag{15}$$

另外，训练和解码的过程中在关键词的末尾添加“EOS”符号来让模型停止生成多余的单词。

2　实验及结果分析

2.1　数据集

使用 4 个科学文献基准数据集来评估模型的效果，包括 KP20k、Inspec、Krapivin 以及 SemEval。对 KP20k 数据集做了预处理，删除了其中重复的数据，最终保留了 509 818 个训练集样本，20 000 个验证集样本以及 20 000 个测试集样本。并且有模型都是基于 KP20k 数据集进行训练。模型训练完成之后，在 4 个基准的测试集上测试了所有模型。数据集的统计结果如表 1 所示。

表 1　数据集统计表

数据集	总数量	验证集	测试集
Inspec	2 000	1 500	500
Krapivin	2 303	1 903	400
SemEval	244	144	100
KP20k	549 818	20 000	20 000

2.2　对比模型和评价指标

首先分别采用以下几个经典的关键词抽取算法作为本文模型的对比模型，其中包括 4 种无监督关键词抽取算法 (Tf-Idf、TextRank、SingleRank[18]、ExpandRank[18]) 和其他基于监督的关键词抽取算法 (Maui[19]、RNN[5]、CNN[6]、CopyRNN[5])。

为了评价每种模型的表现，采用 F_1 分数作为评价指标。F_1 指标由召回率 R、精确度 P 计算得到，计算方式如下：

$$F_1=\frac{2PR}{P+R} \tag{16}$$

最后，模型使用英文分词算法 (porter stemmer) 处理关键词，决定关键词之间是否匹配。

2.3　实验结果

表 2 总结了不同的关键词抽取算法在 4 个基准数据集上的表现情况，给出了前 5 位和前 10 位的 F_1 结果，并且使用粗体显示最好的分数。

表 2　不同模型在 4 种基准数据集上的关键词抽取性能

模型	Inspec		Krapivin		SemEval		KP20k	
	F_1@5	F_1@10	F_1@5	F_1@10	F_1@5	F_1@10	F_1@5	F_1@10
Tf-Idf	0.223	0.304	0.113	0.143	0.120	0.184	0.105	0.130
TextRank	0.229	0.275	0.172	0.147	0.172	0.181	0.180	0.150
SingleRank	0.214	0.297	0.096	0.137	0.132	0.169	0.099	0.124
ExpandRank	0.211	0.295	0.096	0.136	0.135	0.163	N/A	N/A
Maui	0.040	0.042	0.249	0.216	0.044	0.039	0.270	0.230
RNN	0.000	0.000	0.002	0.001	0.004	0.003	0.138	0.009
CNN	0.088	0.069	0.141	0.098	0.162	0.127	0.188	0.203
CopyRNN	0.292	0.336	0.302	0.252	0.291	0.296	0.328	0.255
ours	**0.357**	**0.401**	**0.360**	**0.292**	**0.364**	**0.342**	**0.358**	**0.279**

首先将本文的模型与传统的关键词抽取算法进行比较，结果显示本文的模型在所有测试集上都取得了明显的提升效果。另外，本文模型相较于常见的基于端到端的深度学习方法也有了更进一步提升，说明本文模型引入了文档邻域知识，图卷积方法以及多种多样性机制确实学习到了有效的文档向量表示，并使生成的关键词尽可能多地覆盖了文档主题。本文的模型相较于各对比模型都有着良好的关键词抽取效果，证明了该模型的有效性。

2.4　消融实验

为了验证上下文修改机制和覆盖机制对关键词抽取任务的性能影响，在 Inspec 数据集和 Krapivin 数据集上对模型进行消融实验。其中，“-ConModify”表示不使用上下文修改机制，即模型在解码的时候不参考之前生成的关键词内容来修改文档向量表示。“-CovMech”表示不使用覆盖机制即每当解码器生成关键词单词的时候，不再记录所有单词 one-hot 编码总和。“-Neither”表示两者都不使

用。从表3中能看到如果模型单纯添加了覆盖机制的情况下即“-ConModify”，相比于“-Neither”方法能够在Krapivin数据集上得到更好的效果，然而在Inspec数据集上的效果却降低了。如模型仅仅增加了上下文修改机制即“-CovMech”，结果表明上下文修改机制在两种数据集上都能给模型带来更好的关键词抽取效果。相较于“-Neither”方法，当同时加入上下文修改机制和覆盖机制，模型的关键词抽取效果得到了提升，说明当前两种机制存在很强的互补关系，并且进一步增强了模型的鲁棒性。

表3 消融实验

模型	Inspec		Krapivin	
	F_1@5	F_1@10	F_1@5	F_1@10
Our	**0.357**	**0.401**	**0.360**	**0.292**
-ConModify	0.331	0.360	0.347	0.279
-CovMech	0.343	0.376	0.355	0.284
-Neither	0.335	0.365	0.342	0.266

2.5 参数分析

编码器端使用了图方法来对所有文档进行建模和编码。为了深入研究模型的效果，分析不同GCN层数对模型效果的影响。实验结果如表4所示，结果显示当编码端只有一层GCN的时候，图中的节点无法从邻居节点得到足够的有效信息。在设置2层GCN的情况下，模型在Inspect数据集上取得了较好的效果。直到在实验中设置了6层GCN的情况下，模型在Inspec和SemEval数据集上同时达到了出色的效果。

表4 GCN层数对模型效果的影响

GCN层数	Inspec		SemEval	
	F_1@5	F_1@10	F_1@5	F_1@10
1-layer GCN	0.339	0.377	0.310	0.307
2-layer GCN	0.352	**0.402**	0.334	0.328
4-layer GCN	0.346	0.391	0.358	0.337
6-layer GCN	**0.357**	0.401	**0.364**	**0.342**
8-layer GCN	0.350	0.395	0.361	0.339

实验结果表明随着GCN层数的增加，模型效果会逐渐提升。然而过多的GCN层数会导致整个图中的节点出现同质化的负面效果，导致模型性能下降。另外，通常情况下2层GCN会得到较好的拟合效果，然而卷积网络门控单元和残差学习框架来缓解GCN的过拟合过程，并最终取得了更好的关键词抽取效果。

3 结束语

针对关键词抽取任务，本文提出了一种结合邻域知识的文档级关键词抽取方法。为了更好地建模输入文档的向量表示和生成覆盖文档所有主题的关键词集合，计算文档之间的相似度并选择部分最近邻文档作为原文档的邻域知识。然后根据每个文档中单词之间的长短程关系构建词图，合并输入文档和邻域文档的词图，利用图卷积网络完成编码过程。最后采用指针网络完成解码过程。模型引入上下文修改机制和覆盖机制，共同避免模型生成重复的关键词。在多种数据集上进行关键词抽取任务的对比实验，结果证明结合邻域知识的文档级关键词抽取模型能够有效提升关键词抽取任务的效果。

参考文献

[1] 杨丹浩, 吴岳辛, 范春晓. 一种基于注意力机制的中文短文本关键词提取模型[J]. 计算机科学, 2020, 47(1): 193-198.
YANG Dan-hao, WU Yue-xin, FAN Chun-xiao. Chinese short text keyphrase extraction model based on attention[J]. Computer Science, 2020, 47(1): 193-198.

[2] ZHANG Qi, WANG Yang, GONG Ye-yun, et al. Keyphrase extraction using deep recurrent neural networks on twitter[C]//Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2016: 836-845.

[3] CHEN Wang, CHAN Hou-peng, LI Pi-ji, et al. An integrated approach for keyphrase generation via exploring the power of retrieval and extraction[C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg, PA: ACL, 2019: 2846-2856.

[4] MIHALCEA R, TARAU P. Textrank: Bringing order into text[C]//Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2004: 404-411.

[5] MENG Rui, ZHAO San-qiang, HAN Shu-guang, et al. Deep keyphrase generation[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2017: 582-592.

[6] ZHANG Yong, FANG Yang, XIAO Wei-dong. Deep keyphrase generation with a convolutional sequence to sequence model[C]//The 4th International Conference on Systems and Informatics (ICSAI). New York: IEEE, 2017: 1477-1485.

[7] CHEN Wang, GAO Yi-fan, ZHANG Jia-ni, et al. Title-guided encoding for keyphrase generation[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Menlo Park, CA: AAAI, 2019: 6268-6275.

[8] CHAN Hou-pong, CHEN Wang, WANG Lu, et al. Neural keyphrase generation via reinforcement learning with

adaptive rewards[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2019: 2163-2174.

[9] CHEN Wang, CHAN Hou-pong, LI Pi-ji, et al. Exclusive hierarchical decoding for deep keyphrase generation[C]// Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2020: 1095-1105.

[10] TAM Y C. Cluster-based beam search for pointer-generator chatbot grounded by knowledge[J]. Computer Speech & Language, 2020, 64: 101094.

[11] KIPF T N, WELLING M. Semi-supervised classification with graph convolutional networks[C]//International Conference on Learning Representations(ICLR). La Jolla, CA: ICLR, 2016: arXiv: 1609.02907.

[12] VELICKOVIC P, CUCURULL G, CASANOVA A, et al. Graph attention networks[C]//International Conference on Learning Representations (ICLR). La Jolla, CA: ICLR, 2018: arXiv: 1710.10903.

[13] SUN Zhi-qing, TANG Jian, DU Pan, et al. Divgraphpointer: A graph pointer network for extracting diverse keyphrases[C]//Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 2019: 755-764.

[14] DAUPHIN Y N, FAN A, AULI M, et al. Language modeling with gated convolutional networks[C]// International Conference on Machine Learning. New York: ACM, 2017: 933-941.

[15] VASWANI A, SHAZZER N, PARMAR N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems. Cambridge, MA: MIT Press, 2017: 6000-6010.

[16] VINYALS O, FORTUNATO M, JAITLY N. Pointer networks[C]//Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems. Cambridge, MA: MIT Press, 2015: 2692-2700.

[17] TU Zhao-peng, LU Zheng-dong, LIU Yang, et al. Modeling coverage for neural machine translation[C]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2016: 76-85.

[18] WAN Xiao-jun, XIAO Jian-guo. Single document keyphrase extraction using neighborhood knowledge[C]// Proceedings of the 23rd AAAI Conference on Artificial Intelligence. Menlo Park, CA: AAAI, 2008: 855-860.

[19] MEDELYAN O, FRANK E, WITTEN I H. Human-competitive tagging using automatic keyphrase extraction[C]//Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2009: 1318-1327.

编　辑　蒋　晓