软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software [doi: 10.13328/j.cnki.jos.006406]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于主题交互图的案件话题摘要*

黄于欣[1,2], 余正涛[1,2], 郭军军[1,2], 于志强[1,2], 高凡雅[1,2]

[1](昆明理工大学 信息工程与自动化学院, 云南 昆明 650504)
[2](云南省人工智能重点实验室 (昆明理工大学), 云南 昆明 650504)
通信作者: 余正涛, E-mail:ztyu@hotmail.com

摘　要: 从案件相关的话题评论中生成简短的话题描述对于快速了解案件舆情有着重要作用, 其可以看做是基于用户评论的多文档摘要任务. 然而用户评论中含有较多噪声且生成摘要所需的重要信息分散在不同的评论句中, 直接基于序列模型容易生成错误或不相关的摘要. 为了缓解上述问题, 提出一种基于主题交互图的案件话题摘要方法, 将嘈杂的用户评论组织为主题交互图, 利用图来表达不同用户评论之间的关联关系, 从而过滤重要的用户评论信息. 具体来说, 首先从评论句中抽取案件要素, 然后构造以案件要素为节点, 包含案件要素的句子为内容的主题交互图; 然后利用图 Transformer 网络生成图中节点的表征, 最后生成简短的话题描述. 在收集的案件话题摘要数据集上的实验结果表明, 所提方法是一种有效的数据选择方法, 能够生成连贯、事实正确的话题摘要.

关键词: 案件话题摘要; 用户评论; 主题交互图; 案件要素; 图 Transformer 网络
中图法分类号: TP18

中文引用格式: 黄于欣, 余正涛, 郭军军, 于志强, 高凡雅. 基于主题交互图的案件话题摘要. 软件学报. http://www.jos.org.cn/1000-9825/6406.htm
英文引用格式: Huang YX, Yu ZT, Guo JJ, Yu ZQ, Gao FY. Case-related Topic Summarization Based on Topic Interaction Graph. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/6406.htm

Case-related Topic Summarization Based on Topic Interaction Graph

HUANG Yu-Xin[1,2], YU Zheng-Tao[1,2], GUO Jun-Jun[1,2], YU Zhi-Qiang[1,2], GAO Fan-Ya[1,2]

[1](Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650504, China)
[2](Yunnan Key Laboratory of Artificial Intelligence (Kunming University of Science and Technology), Kunming 650504, China)

Abstract: Generating coherent topic descriptions from the user comments of case-related topics plays a significant role in quickly understanding the case-related news, which can be regarded as a multi-document summarization task based on user comments. However, these comments contain lots of noise, the crucial information for generating summaries is scattered in different comments, the sequence-to-sequence model tends to generate irrelevant and incorrect summaries. Based on these observations, this paper presents a case-related topic summarization method based on the topic interaction graph, which reconstructs the user comments into a topic interaction graph. The motivation is that the graph can express the correlation between different user comments, which is useful to filter the key information in user comments. Specifically, the case elements are first extracted from the user comments, and then the topic interaction graph is constructed, which takes the case elements as the nodes and uses the sentences including these case elements as the node's contents; then the graph transformer network is introduced to produce the representation of the graph. Finally, the summary is generated by using a standard transformer-based decoder. The experimental results on the collected case-related topic summarization corpus show that the proposed method effectively selects useful content and can generate coherent and factual topic summaries.

Key words: case-related topic summarization; user comments; topic interaction graph; case elements; graph Transformer network

---

* 基金项目: 国家重点研发计划 (2018YFC0830105, 2018YFC0830100); 国家自然科学基金 (61972186, 61732005, 61762056); 云南省重大科技专项 (202002AD080001); 云南省基础研究专项面上项目 (202001AT070047, 202001AT070046); 云南省高新技术产业专项 (201606)
收稿时间: 2021-02-10; 修改时间: 2021-05-21; 采用时间: 2021-06-30;

1 引 言

案件相关的舆情新闻在社交媒体上快速传播会形成大量的话题和评论. 从同一话题的多个用户评论中生成简短的话题摘要对于快速了解案件的发生、发展过程有着重要的意义. 案件话题摘要任务可以看做一个面向特定领域的多文档摘要任务 (multi-document summarization, MDS), 旨在将同一主题下的多个文档压缩、重写为简短流畅的摘要. 目前针对多文档摘要任务主要的研究方法包括抽取式 (extractive) 和生成式 (abstractive) 两类: 抽取式摘要方法直接从源文档中抽取重要的句子作为摘要, 但是摘要句必须来自于原文, 这在一定程度上限制了生成摘要的新颖性. 而生成式摘要方法则更符合人类的摘要习惯, 首先编码多文档输入, 理解其核心思想, 在此基础上利用解码器逐词生成摘要. 随着深度学习的发展, 生成式摘要方法逐渐成为了研究的热点.

近年来, 基于序列到序列框架 (sequence-to-sequence, seq2seq) 的生成式摘要方法虽然在单文档摘要 (single document summarization, SDS) 任务上取得了很好的效果[1−3], 然而直接将其迁移到多文档摘要场景下仍然面临一些难题: 一是多文档摘要输入一般较长, 传统的编码器无法处理过长的输入序列; 另外多文档输入中含有较多的冗余信息, 从多文档中过滤噪声数据, 选择有用的信息较为困难. 为了缓解上述问题, 目前的研究主要从以下两个方面进行改进: 一是通过改进编码器来适应长的输入序列. Celikyilmaz 等人[4]提出引入多个编码器来处理长序列的不同部分, 通过建立编码器之间的交互关系产生更好的上下文表示. 另外也有学者提出将长序列分解为词、句子、段落、篇章等不同层次, 然后利用分层编码器来生成长文档表示[5−7]. 另一类解决方法是采用先抽取再生成的两阶段方法来缓解输入序列过长的问题, 即首先从多文档中抽取重要的内容, 再基于序列到序列模型生成摘要. Gehrmann 等人[8]提出利用标准摘要作为内容选择的监督信号来判断输入序列中的单词是否应该被选择为摘要, 从而实现输入序列自底向上的数据选择. 另外也有学者提出首先从输入长序列中选择重要的句子[9]或者段落[10]表示原文的重要信息, 然后再基于序列到序列框架生成摘要. 为了克服两阶段方法带来的误差累积问题, 也有学者提出基于强化学习来联合优化内容选择和摘要生成过程提升摘要的生成质量[11]. 然而虽然这些方法可以有效的缩短输入序列的长度, 降低摘要生成模型的复杂度, 但是其摘要的生成质量仍依赖于前一步内容选择模型的性能. 因此也有部分学者提出将内容选择融入摘要生成过程中, 通过引入不同的外部知识来实现重要内容的自动选择. 外部知识包括从原文抽取的关键词[12,13], 从语料库中学习的全局主题信息[14,15]和从语料库中检索的软模板[16,17]等不同形式, 然后再基于注意力机制或者指针网络将外部知识融入到序列到序列模型中, 改进模型从原文中选择重要信息的能力, 提升模型性能.

然而, 虽然目前多文档摘要生成的方法已在维基百科[10]、新闻文本[18]、医学文本[19]等领域上取得了很好的性能, 但是在处理用户评论等不规范、表达方式随意的文本时仍然面临许多新的挑战[20]. 本文聚焦从案件相关话题的用户评论中生成话题摘要的任务.表 1 给出了案件话题摘要任务示例, 包括案件话题标题、话题导语以及相关的用户评论. 如表 1 所示虽然大部分的用户都围绕“新郎”“新娘”“继承”等案件要素展开讨论, 但是也存在以下几个问题: (1) 用户评论比较发散, 很多评论和案件相关性较低, 如表 1 中的评论 5 虽然也是案件相关的评论, 但是实际上信息量较小. 因此案件话题摘要任务需要模型具有较好的噪声数据过滤能力, 能够有效过滤不相关的用户评论. (2) 生成话题摘要所需要的重要信息分散在不同的评论句中, 如表 1 中粗体部分为案件相关的要素信息, 可以看出这些信息分散在不同评论的不同句子中, 这也要求摘要模型能够准确的从用户评论中筛选重要信息. (3) 不同评论的长度差异较大, 在收集的案件话题摘要数据集中, 17% 的样本的评论句数量超过 50 句, 26% 的评论超过 1 000 词, 这也要求摘要生成模型能够适应不同长度文本的建模.

针对以上问题, 本文在两阶段的多文档摘要框架下, 提出一种基于主题交互图的数据选择方法, 把文本到文本的生成任务转换为图到文本的生成任务. 具体来讲, 首先从用户评论中抽取案件要素, 并以这些要素为图的节点构造主题交互图. 为了丰富节点的表示, 图的节点不仅包含词粒度的案件要素信息, 并且还包括和要素高度相关的评论句, 每个包含要素和关键句的节点簇可以看做是用户评论中的不同的子主题. 为了进一步描述这些子主题之间的关联关系, 本文通过计算任意两个节点之间重叠的句子数来表示其关联强度. 相比直接对多文档输入进行层次化建模[4]或先抽取重要信息再生成摘要[11]的方法, 直接将嘈杂的用户评论表征为案件要素相关的主题交互图不仅

能够更好的控制数据中的噪声和冗余信息, 实现重要信息筛选, 并且能够进一步建模不同要素之间的关联关系, 从而让模型更好的理解用户评论. 基于构造的主题交互图, 本文提出一种图到序列的生成框架, 首先利用预训练模型对主题交互图中的节点进行初始化, 然后利用图 Transformer 模型对其进行编码得到图中节点的向量表示, 最后基于序列解码器逐词生成话题摘要.

表 1　案件话题摘要示例

话题标题	新郎婚礼当日坠亡新娘继承183万
话题导语	近日, 辽宁锦州. 王某婚礼当天坠亡, 刚领证半个月新娘张某诉求继承183万元被公婆起诉.
评论1	并没有形成夫妻事实, 所以这个**继承**有点不合理, 这个**新郎**的死疑点重重
评论2	争夺遗产的还有前妻, **公婆败诉**…更大疑点为何在**婚礼**两小时前**坠亡**
评论3	没有哪一个**新娘**想要这**183万**. 对于这样的**继承**, 就是祸从天降. 本应该到来的幸福时刻, 等来的却是**新郎的坠亡**. 如果可以选择, 俺相信**新娘**会选**新郎**的复活.
评论4	王某婚礼当天**坠亡**, 刚领证半个月**新娘**张某诉求**继承183万**元被公婆起诉…**法院一二审**均认定王父供证不足, 不能证明存款系其名下所有, 张某胜诉. …一审二审法院都不支持男方父母诉求…**新娘**却不善罢甘休, 既然结了婚领了证, 就有**继承**遗产的权利, 所以两家开始打官司. 法院讲求的是证据, 判父母败诉.
评论5	这钱拿得伤天害理[鄙视]. 现在越来越相信, 一个人如果伤天害理的事情做尽, 法律最终也救不了你, 不是不报, 时候不到.

2　相关工作

2.1　基于图的抽取式摘要方法

早期基于图的文本摘要方法多为抽取式摘要, 其核心思想是利用文档中的句子作为图的节点, 通过图优化算法得到句子在文档中的重要性得分. Mihalcea 等人[21]提出一种 TextRank 算法, 利用句子的余弦相似度构造句子关联图模型中各节点之间边的权重, 然后迭代生成句子的重要性得分. Wan 等人[22]在此基础上提出在摘要过程中不仅需要考虑句子和句子之间的关联关系, 同时需要在构建图的过程中引入文档节点, 构造文档和句子之间关系, 该方法在 DUC 摘要数据集上取得了很好的性能. Zhao 等人[23]提出一种基于子图聚类的无监督多文档摘要方法, 通过词嵌入和词共现等信息来构造句子间的关联关系, 然后基于图聚类算法获得不同的关联子图, 最后从子图中选择摘要句. 基于图的抽取式摘要另一个研究方向是将摘要任务转换为基于深度学习框架的句子二分类任务, 核心在于如何利用图结构来更好建模句子、文档之间的交互关系, 得到更好的句子表示. Yasunaga 等人[24]提出一种基于图卷积网络 (graph convolutional networks, GCNs)[25]的多文档摘要方法, 首先构造句子关联图, 利用循环神经网络 (recurrent neural networks, RNNs) 初始化图的节点, 然后基于 GCN 编码句子特征并基于此实现摘要句分类. 相比传统方法, 该模型在 DUC 数据集上取得了很好的性能. Antognini 等人[26]进一步提出一种改进的语义关联图构建方法, 利用预训练模型来表征句子并基于此计算句子的相似度, 生成句子的语义关联图. 该方法通过引入预训练的通用知识, 能够更准确表达不同句子的关联关系, 因此取得了更好的性能. 受到以上方法的启发, Wang 等人[27]提出一种将输入文本构造为包含词、句子和文档的异构图来表示文档之间、文档句子、句子词等不同层次的关联关系, 最后基于 GCN 进行编码生成句子表示.

2.2　基于图的生成式摘要方法

近年来基于图的生成式摘要方法获得了广泛的关注. 其核心是利用图来改进序列到序列框架的编码器以适应多文档输入. Tan 等人[28]基于序列到序列框架提出一种基于图的改进注意力机制, 在每个解码时刻利用图模型来建模句子的重要性得分, 从而让模型能够显式的关注最重要的句子, 提升了模型的性能. Liu 等人[5]提出一种层次化的 Transformer 模型, 利用注意力机制隐式的建模多文档间的关联关系. Jin 等人[6]进一步提出一种基于注意力机制的词、句子和文档的多粒度结构信息交互式建模方法, 基于词的表示解码生成摘要, 基于句子表示抽取相应的摘要句, 从而联合优化句子抽取过程和摘要生成过程. Li 等人[7]在此基础上提出一种利用段落级的语义相似图和语篇结构图作为外部知识来增强文档表征的生成式摘要方法, 通过引入图结构信息作为自注意力机制的正则

项, 引导模型更好的建模文档间的关联关系. 以上研究主要关注如何利用图结构来捕捉文档、句子、词之间的层次关系, 帮助解码器更好的理解复杂的多文档输入. 但是和之前的研究工作不同, 本文更关注如何利用图结构实现噪声数据的过滤和重要信息选择, 探讨如何将嘈杂的多文档输入分解为图结构表示以及如何更好建模图节点及其关联关系.

3 主题交互图构造

本节主要介绍如何从用户评论中构造主题交互图. 设给定案件相关话题的用户评论集 $\mathcal{D}=\{d_1,d_2,\ldots,d_{|D|}\}$, 其中 $d_i=\{s_{i,1},s_{i,2},\ldots,s_{i,|S|}\}$ 表示第 i 个评论, $|D|$ 和 $|S|$ 分别表示评论的数量和每个评论的句子数量, $s_{i,j}$ 表示第 i 个评论中的第 j 个句子. 基于此构造话题关联图 $G=(\mathcal{V},E)$, 其中顶点 $v_i\in\mathcal{V}$ 包含案件要素集合 k_i 及相关评论句集合, 边 $e_{i,j}=(v_i,v_j)\in E$ 表示顶点 v_i 和 v_j 之间存在关联关系, 关联强度为边的权重 $w_{i,j}$. 如算法 1 所示, 本文首先从用户评论中抽取案件要素集合 K, 然后将包含要素 k 的评论句 s 分配给节点 v, 最后通过计算不同节点间重叠的句子数得到其关联权重 $w_{i,j}$.

算法 1. 主题交互图构造算法.

输入: 用户评论集 $\mathcal{D}$;
输出: 主题交互图 $G=(\mathcal{V},E)$.

1. 从用户评论中抽取案件要素集合 K
2. 初始化顶点 $\mathcal{V}$, 令 $v_k=k$
3. FOR d_i IN $\mathcal{D}$ DO
4. FOR $s_{i,j}$ IN d_i DO
5. IF $s_{i,j}$ CONTAINS $k\in K$ THEN
6. $v_k=v_k\cup s_{i,j}$
7. END FOR
8. END FOR
9. FOR d_i IN $\mathcal{D}$ DO
10. FOR $s_{i,j}$ IN d_i DO
11. IF v_i AND v_j CONTAINS $s_{i,j}$ THEN
12. $e_{i,j}=(v_i,v_j)$
13. $w_{i,j}=w_{i,j}+1$
14. END FOR
15. END FOR

具体来讲, 本文首先利用 TextRank 算法[21]从用户评论中抽取关键词作为案件要素, 其可以看做是案件中用户所关注的不同方面, 即用户评论中的子主题. 每个案件要素 k 都被当做是主题交互图的节点 v. 但是仅利用案件要素作为主题交互图节点可能导致用户评论信息损失过多, 影响模型的性能. 因此为了丰富节点的表征, 本文进一步将包含案件要素的句子 s 并入节点 v. 即如果句子 s 包含关键词 k 则将该句子分配到当前节点 v, 对于不包含任何关键词的句子则直接丢弃. 最后本文提出一种简单有效的方法来构造不同顶点之间的关联关系, 即如果节点 v_i 和 v_j 中包含相同的句子则为其构建关联边 $e_{i,j}$, 边的权重 $w_{i,j}$ 为两个顶点中相同句子的数量.

图 1 给出了表 1 中给出例子的主题交互图构建结果, 其中案件要素集合为“婚礼、新郎、坠亡、新娘、继承、183 万”等, 每个案件要素簇还包括包含该要素的句子集合. 要素之间的关系通过统计要素相关的句子集合中重叠句子的数量来构建. 如在“新娘”和“继承”两个要素之间构建关联关系, 是因为“新娘”和“继承”两个要素均出现在“刚领证半个月新娘张某诉求继承 183 万元被公婆起诉”这个句子中. 从图 1 中也可以看出, 相比基于句子相似度[7]

的方法, 本文提出的方法能够更直接表示不同案件要素之间的关联关系, 避免因句子表示不准确而带来的误差传递问题. 本文也给出了不同的 TIG 图构造方法的对比结果, 详细分析见第 6.2 节.

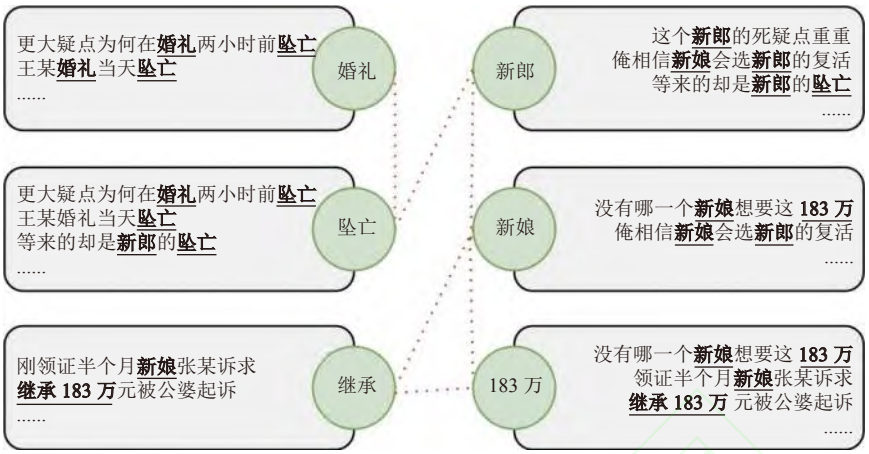


图 1　主题交互图示例

4　基于主题交互图的案件话题摘要

基于构建的主题交互图, 本文提出一种图到序列的案件话题摘要生成方法. 如图 2 所示, 提出的模型包括图编码器和序列解码器两部分, 其中图编码器负责对生成主题交互图进行编码, 生成每个节点的语义表征, 序列解码器通过读取图编码器的表示, 基于注意力机制逐词生成摘要.

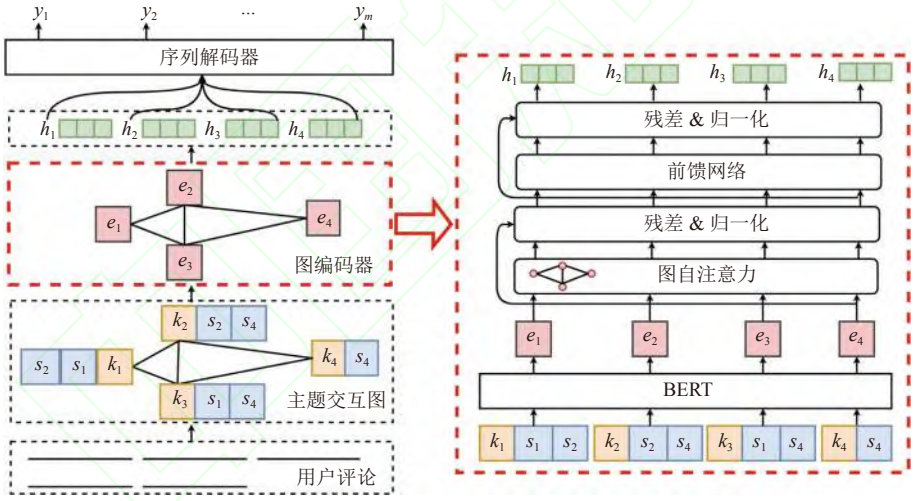


图 2　基于主题交互图的案件话题摘要模型

4.1　图编码器

近年来, 以图卷积 (graph convolutional networks, GCNs) 和图注意力 (graph attention networks, GATs) 为代表的图编码器在知识图谱[29]、概念图[30]和文档异构图[31]等图结构数据上取得了很好的效果, 特别是基于 GATs 的图 Transformer 模型将图注意力机制和 Transformer 框架结合可以实现更好的图结构数据建模. 因此本文采用图 Transformer 作为主题交互图编码器来生成节点的表示.

具体来讲, 给定图 $G=(\mathcal{V},E)$, 首先需要对图节点 v_i 进行初始化, 得到每个节点的高纬向量表示. 近年来基于自监督优化目标的预训练模型能够从大规模无标记语料中学习隐含的语义规则和通用的知识, 在文本分类、智能问答以及机器翻译等任务上取得了很好的性能. 因此本文为了生成更好的节点 v_i 的表示, 提出一种基于预训练模型 (bidirectional encoder representations from transformers, BERT)[32]的图节点嵌入方法. 具体来讲, 首先基于节点 $v_i=\{k_i,S_i\}$ 得到 BERT 模型的输入序列, 即 [CLS];[k_i];[s_0];[SEP];[s_1];[SEP]$\cdots$, 其中 [CLS] 为 BERT 模型的分类标签, [SEP] 为不同句子的分隔标签. 节点的初始化过程如公式 (1) 所示:

$$\mathbf{H}_i=\mathrm{BERT}(v_i) \tag{1}$$

其中, $\mathbf{H}_i$ 表示 BERT 模型的最后一层输出向量序列, $\mathbf{H}_{i,cls}$ 则表示 $\mathbf{H}_i$ 序列中 [CLS] 标签对应的向量. 特别说明, 尽管 BERT 模型能够为下游任务提供有用的知识, 但是参数规模较大 (约 110M), 微调整个 BERT 模型可能会带来过拟合和灾难性遗忘等问题. 因此为了避免以上问题, 本文采用基于适配器网络的 BERT 模型微调方法[33], 即首先固定 BERT 模型参数, 然后引入一个多层感知机 (multi-layer perceptron, MLP) 适配器网络来匹配 BERT 模型和 Transformer 模型的维度, 生成节点的初始特征表示 e_i:

$$e_i=\tanh(\mathbf{W}_e\mathbf{H}_{i,cls}+b_e) \tag{2}$$

其中, tanh 为激活函数, $\mathbf{W}_e\in\mathbb{R}^{d\times l}$ 和 $b_e\in\mathbb{R}^{d\times 1}$ 为训练参数, d 和 l 分别表示图 Transformer 模型的嵌入维度 (本文设置为 512) 和 BERT 模型的输出维度 (本文设置为 768).

在得到节点的初始特征表示的基础上, 本文引入图注意力机制来生成当前节点的表示. 具体来讲, 针对节点 v_i, 通过汇聚节点 v_i 及其所有的邻居节点 $\mathcal{N}_i$ 的表征, 生成 v_i 的表征. 公式如下:

$$\hat{h}_i^l=h_i^{l-1}+\sum_{j\in\mathcal{N}_i}\alpha_{i,j}\mathbf{W}_V\mathrm{h}_j^{l-1} \tag{3}$$

其中, $\hat{h}_i^l$ 为第 l 层节点 v_i 的表示, $\mathcal{N}_i$ 表示节点 v_i 在图 G 中的邻居节点, $\mathbf{W}_V$ 为训练参数. 特别说明 $h_i^0=e_i$. $\alpha_{i,j}$ 为节点 v_i 和 v_j 之间的注意力权重, 其计算过程如下:

$$\begin{cases}\alpha_{i,j}=softmax\left(u_{i,j}\right)=\dfrac{\exp\left(u_{i,j}\right)}{\sum\limits_{j\in\mathcal{N}_i}\exp\left(u_{i,j}\right)}\\ u_{i,j}=\dfrac{\left(\mathbf{W}_K h_j^{l-1}\right)^{\top}\mathbf{W}_Q h_i^{l-1}}{\sqrt{d_k}}\end{cases} \tag{4}$$

其中, $\mathbf{W}_*$ 为训练参数. 模型首先计算 h_i^{l-1} 和 h_j^{l-1} 的内积, 然后利用 $softmax$ 函数实现权值归一化. 特别说明, 为了避免 h_i^{l-1} 和 h_j^{l-1} 的内积过大, 通常除以 $\sqrt{d_k}$ 对输出进行缩放, d_k 为图 Transformer 的隐状态维度.

图 Transformer 模型在 GATs 模型的基础上, 引入层归一化和残差连接机制, 能够有效的缓解梯度消失问题, 取得了更好的性能. 因此本文在图注意力模块基础上, 引入残差连接和层归一化来生成节点 v_i 的最终表征 h_i:

$$\begin{cases}h_i=LayerNorm\left(h_i'+LayerNorm\left(\hat{h}_i\right)\right)\\ h_i'=FFN\left(LayerNorm\left(\hat{h}_i\right)\right)\end{cases} \tag{5}$$

其中, FFN 表示以 ReLU 作为激活函数的前馈神经网络.

4.2 解码器

本文采用基于 Transformer 的序列解码器. 在解码时刻 t, 解码器首先计算 t–1 时刻解码器隐状态 s_{t-1} 和编码端隐状态序列 $\mathbf{H}=(h_1,h_2,\ldots,h_N)$ 之间的注意力权重 $\alpha_{t,i}$, 公式如下:

$$\begin{cases}e_{t,i}=\mathbf{v}_a^{\mathrm{T}}\tanh\left(\mathbf{W}_a\left[h_i;s_{t-1}\right]\right)\\ \alpha_{t,i}=softmax\left(e_{t,i}\right)=\dfrac{\exp\left(e_{t,i}\right)}{\sum\limits_{i=1}^{N}\exp\left(e_{t,i}\right)}\end{cases} \tag{6}$$

其中, $\mathbf{v}_a^{\mathrm{T}}$ 和 $\mathbf{W}_a$ 为训练参数, [;] 表示拼接操作, tanh 为激活函数. 然后基于生成的注意力权重, 对编码端隐状态序列 $\mathbf{H}$ 进行加权求和得到 t 时刻的上下文向量 c_t, 如公式 (7) 所示:

$$c_t = \sum_{i=1}^{N} \alpha_{t,i} h_i \tag{7}$$

最后解码器通过融合 t 时刻解码器的隐状态 s_t, 上下文向量 c_t, 计算生成 t 时刻 y_t 的在目标词表上的分布概率, 公式如下:

$$p(y_t \mid y_1, \ldots, y_{t-1}) = softmax\left(\mathbf{W}_p \tanh(\mathbf{W}_t [s_t; c_t])\right) \tag{8}$$

其中, $\mathbf{W}_p$ 和 $\mathbf{W}_t$ 为训练参数. 最终模型通过最小化生成摘要 y 和标准摘要 $\hat{y}$ 之间的交叉熵损失来优化整个模型.

$$\mathcal{L}(\theta) = -\sum \log p(y|\mathcal{D}, G) \tag{9}$$

5 实验设置

5.1 数据集

为了验证提出方法的有效性, 本文从新浪微博中收集案件相关的话题及相关的评论信息来构造案件话题摘要数据集. 具体来讲, 本文从澎湃新闻、头条新闻等微博账号中抓取新闻约 23 万条左右, 然后利用相关性分析方法[34]筛选出与司法领域相关的新闻并从涉案新闻中抽取话题 (以##包括内容), 并对抽取得到的话题其进行人工标注得到案件相关话题. 最后基于此抓取话题对应的导语和用户评论, 去除话题评论中重复的内容形成话题标题-导语-评论形式的数据对. 在此基础上本文对收集的数据进行预处理, 过滤标题少于 5 个字, 导语为空或评论为空的数据, 最终得到包含 48 135 个话题的案件话题摘要数据集. 本文进一步对数据集进行随机划分, 得到包含 46 135 个样本的训练集、1 000 测试集和 1 000 验证集. 具体统计信息如表 2 所示.

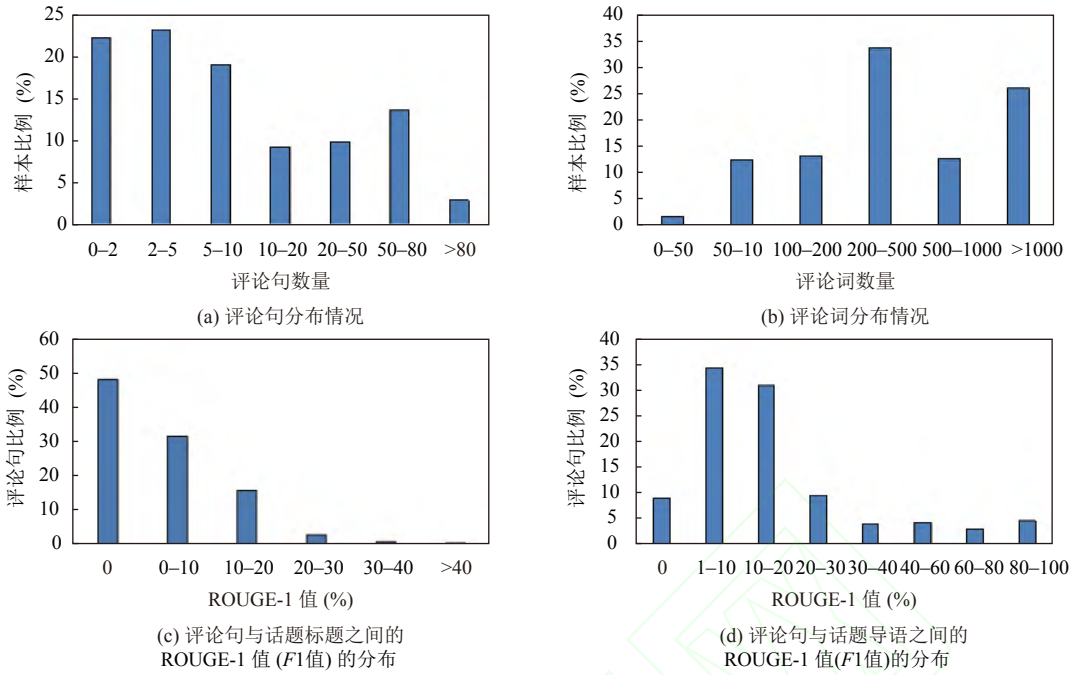
表 2 案件话题摘要数据集统计信息

统计项	训练集	测试集	验证集
样本数	46 135	1 000	1 000
标题平均词数	6.28	6.38	6.28
导语平均词数	52.72	51.26	51.45
评论平均条数	18.57	20.44	19.98
评论平均词数	873.15	950.97	925.53

从表 2 可以看出, 每个话题平均包含 19 条左右的评论句, 但是评论句的分布极不均衡, 部分话题仅包含 1 条评论, 而最多的则包含 96 条评论, 数量差异较大. 另外也可以看出, 每个话题平均包含 916 个词, 这个长度超过了 Transformer 模型的处理能力. 本文进一步从定量的角度给出了案件话题摘要数据集的评论句数量和评论句词数的分布情况, 如图 3 所示. 从图 3(a) 可以看出, 虽然数据集中评论句数量少于 10 的样本较多 (64%), 但是仍然有 17% 的样本的评论句数量超过 50 句, 这也要求本文提出的模型能够从大量的评论中过滤噪声数据. 从图 3(b) 中也能看出, 评论词超过 500 的样本数占总样本的比例超过了 39%, 且长度超过 1 000 的样本超过了 26%, 这些超长的样本也给提出的摘要模型带来了挑战.

本文进一步从内容分析角度, 通过计算评论句和话题标题及导语之间 ROUGE 值来评价评论句和话题标题及导语之间的内容相关程度.ROUGE 值是一种通过统计生成摘要和标准摘要之间的 n 元组重叠度来评价摘要的质量 (详细介绍见第 5.2 节). 从图 3(c) 中可以看出, 有 48% 的评论句和话题标题之间不存在任何重叠的词, 9% 的评论句和导语完全不相关, 这也说明用户的评论内容比较发散, 因此噪声数据较多. 尤其是评论句和话题标题之间不相关的比例较高, 可能的原因是话题标题是内容的高度浓缩, 一般较短 (6 个词左右), 因此 ROUGE 值也较相对较低.

通过以上分析可知案件话题摘要数据集中含有较多的噪声数据, 提出的模型需要有很强的噪声数据过滤能力. 另外数据集中评论句平均长度较长且长度差异大, 提出的模型要具有处理长数据的能力和适应不同长度的输入.



(a) 评论句分布情况　(b) 评论词分布情况

(c) 评论句与话题标题之间的 ROUGE-1 值 ($F1$值) 的分布　(d) 评论句与话题导语之间的 ROUGE-1 值($F1$值)的分布

图 3　案件话题摘要数据集分布情况

5.2　评价指标

本文采用通用的 ROUGE (recall-oriented understudy for gisting evaluation)[35]值作为案件话题摘要任务的评价指标, 其通过计算标准摘要和生成摘要之间 n 元组 (n-gram) 的共现信息来评价摘要的生成质量, 和之前的工作相同[1,2], 本文使用 Pyrouge 脚本计算生成摘要和标准摘要之间的 ROUGE-1, ROUGE-2 和 ROUGE-L 的 $F1$ 值作为评价结果, 分别表示词 (word)、二元组 (bigram) 和最长子序列之间 (longest common sequence) 的重叠关系. 为了简便本文分别将其简写为 RG-1, RG-2 和 RG-L.

5.3　实现细节

本文基于 Fairseq[36]框架实现话题摘要模型. 在模型训练方面, 批次大小设置为 4 096 词, 训练轮次设置为 50 轮. 采用 Adam 作为模型优化器, 学习率设置为 7e−4. 训练前 4 000 步采用 Warm-up 策略, 最小学习率设置为 1e−9. 在训练过程中采用标签平滑 (label smoothing) 策略, 平滑因子设置为 0.1. 训练采用梯度累加策略, 累加次数设置为 4. 所有模型训练过程中 dropout 设置为 0.1. 在模型词表方面, 采用 BPE 策略生成词表, 共享编解码器词表和词嵌入层. 在模型结构方面, 编解码器均采用 6 层 Transformer, 词嵌入的维度和模型的隐状态维度设置为 512, 多头注意力设置为 8, 前馈神经网络设置为 1 024. 解码推断阶段, 采用标准的集束搜索 (beam search) 策略, beam-size 设置为 5. 当模型解码长度超过预设最大长度 (标题生成设置为 10, 导语生成设置为 50) 或者生成 EOS 标签时停止解码.

5.4　基线模型

为了验证提出方法的有效性, 本文选择以下基线系统, 包括标准 Transformer 编解码模型、基于两阶段的摘要方法和基于图的摘要方法:

(1) Flat Transformer (F-Trans)[37]: 编解码器均采用标准的 6 层 Transformer 模型. 案件话题评论被拼接为一个长文本, 然后截取前 500 个词作为输入.

(2) Hierarchical Transformer (H-Trans)[38]: 为了处理长序列, 首先利用词级 Transformer 对每条评论单独编码,

然后对所有评论词的隐状态进行平均池化 (average-pooling) 得到每条评论的向量, 然后基于句子级 Transformer 得到每个评论句的编码向量. 解码器在解码时同时关注词级的上下文和句子级的上下文.

(3) KeySents Transformer (KS-Trans): 采用两阶段的摘要生成策略, 首先利用 Textrank 算法[21]从话题评论中抽取关键句, 然后基于标准的 Transformer 生成摘要, 关键句的数量设置为 50.

(4) Keyword Transformer (KW-Trans): 采用两阶段的摘要生成策略, 首先利用 Textrank 算法从话题评论中抽取关键词, 然后基于标准的 Transformer 生成摘要, 关键词的数量设置为 100.

(5) Keyword graph Transformer (KW-GraphTrans): 基于图的摘要生成方法, 首先从话题评论中抽取关键词并基于共现关系构建关键词图, 然后基于图 Transformer 对关键词图进行编码得到关键词表征并生成摘要, 关键词数量设置为 100.

(6) Topic interaction graph Transformer (TIG-GraphTrans): 本文提出方法, 基于关键词和包含关键词的句子构建主题交互图, 然后基于图 Transformer 编码得到节点特征并生成摘要, 关键词数量设置为 100.

6 实验结果分析

6.1 基线模型结果对比

本文在表 3 中给出了不同基线模型在标题生成任务和导语生成任务上的对比实验结果, 其中 Δ 表示本文提出的 TIG-GraphTrans 相比其他基线模型在 RG-1 指标上的性能提升.

表 3　不同基线模型结果对比

任务	模型	RG-1	RG-2	RG-L	Δ (%)
标题生成	F-Trans	88.73	84.83	88.63	4.79
	H-Trans	86.42	81.66	86.09	7.59
	KS-Trans	91.17	88.26	89.87	1.99
	KW-Trans	89.31	86.08	89.31	4.11
	KW-GraphTrans	91.73	88.95	91.80	1.36
	TIG-GraphTrans	92.98	90.54	93.03	—
导语生成	F-Trans	50.77	35.10	49.36	5.16
	H-Trans	50.92	36.46	49.22	4.85
	KS-Trans	51.57	36.60	50.18	3.53
	KW-Trans	45.01	32.38	43.53	18.62
	KW-GraphTrans	49.80	34.47	48.29	7.21
	TIG-GraphTrans	53.39	38.50	51.90	—

从表 3 可以看出, 在标题生成任务上所有的基线模型均取得了较好的性能. 如 F-Trans 模型直接对话题评论进行截取仍然取得了 88.73 的 RG-1 得分. 这也说明标题生成任务相对简单 (话题标题平均长度仅 7 个词), Transformer 模型很容易就能学习到编码器和解码器之间映射关系. 另外也可以发现 H-Trans 模型性能相比其他模型有一定的下降, 可能的原因是 H-Trans 引入了词和句子级编码器对用户评论进行分层编码, 模型复杂度高, 参数量大, 从而导致模型出现了过拟合问题. 另外可以看出两阶段模型 KW-Trans 和 KS-Trans 模型首先抽取关键词和关键句, 精炼了摘要生成模型的输入, 因此相比 F-Trans 模型取得了更好的性能. 另外相比 KW-Trans 模型, KS-Trans 模型在 RG-1, RG-2 和 RG-L 这 3 个指标上分别提高了 1.86, 2.01 和 3.01. 本文分析原因是: 关键句相比关键词包含的信息量更丰富, 流畅性也更好, 因此在摘要生成时能够更好的覆盖原文的重要信息. 另外 KW-GraphTrans 模型通过将关键词组织为共现关键词图, 基于 Graph Transformer 模型能够生成更好的关键词表示, 因此相比 KW-Trans 模型, 在 ROUGE-1 指标上也取得了 2.42 的提升. 从表 3 还可以看出, TIG-GraphTrans 模型能够联合利用关键词和关键句的信息, 并且将其组织为主题交互图, 因此相比其他模型取得了更好的性能. 相比 F-Trans 模型, TIG-GraphTrans 模型在 RG-1, RG-2 和 RG-L 分别提高了 4.25, 5.71 和 4.4, 相比 KW-GraphTrans 模型

也提升了 1.25, 1.59 和 1.23.

相比话题标题生成任务, 话题导语生成任务需要生成平均长度 51 个词左右的导语作为话题摘要. 从表 3 可以看出, 相比其他的基线模型, TIG-GraphTrans 模型同样取得了最优的性能, 与 F-Trans、KW-Trans 和 KW-GraphTrans 模型相比在 RG-1 指标上分别取得了 5.16%、3.53% 和 7.21% 的提升. 这也说明在复杂的导语生成任务中, TIG 能够同时利用关键词和关键句来表达原文的重要信息, 并通过建立不同节点之间的交互关系帮助模型更好的理解话题, 有效的实现重要信息的选择和无关信息的过滤. 从表 3 还可以看出, 导语生成任务与标题生成任务表现出了一些不同的特点: (1) 导语生成相比标题生成难度更大, 不同模型的性能差异较为显著. 可能的原因是话题标题长度较短, 解码过程简单, 因此解码器对编码器的输入并不敏感, 即便输入中存在噪声, 模型仍然能够很好的生成标题. 但是导语生成任务则更依赖编码器的输入, 要求模型具有更精细化的输入筛选能力. (2) KW-Trans 和 KW-GraphTrans 两个基于关键词的方法相比其他基于句子级输入的模型能差距较大, 即便在标题生成任务中取得较好性能的 KW-GraphTrans 模型相比 F-Trans 和 TIG-GraphTrans 在 RG-1 指标上仍相差 0.97 和 3.59. 本文认为可能的原因是关键词虽然能够表征用户评论的要点, 但是也同样存在较多的信息损失, 在生成导语这样语义复杂的摘要时无法很好的提供用户评论得到全局信息.

通过以上分析可以得出结论, TIG 能够同时利用关键词及关键句的信息, 是一种有效的数据选择方法, 在标题生成和导语生成两类摘要生成任务上都取得了良好的性能.

6.2 不同 TIG 图结果对比

本文提出的 TIG 图以抽取的关键词和包含关键词的句子作为节点, 以不同节点是否包含相同的句子来构建不同节点之间的边. 为了进一步验证不同的 TIG 图构造方法对模型性能的影响, 本文提出了 3 种不同的 TIG 图构造方法: (1) PMI-TIG: 首先计算关键词之间的点互信息 (point-wise mutual information, PMI), PMI 信息越大表明关键词之间的相关性较强, 反之亦然. 通过在 PMI>0 的关键词之间构建关联关系得到 TIG 图, PMI 计算窗口大小设置为 5; (2) TFIDF-TIG: 基于 TF-IDF 算法首先生成每个顶点的表示, 然后基于此计算不同节点之间的余弦相似度 (cosine similarity), 相似度超过阈值的节点之间则构建关联关系, 阈值设置为 0.5; (3)LDA-TIG: 和模型 TFIDF-TIG 相似, 不同在于每个顶点的表示为基于 LDA (latent dirichlet allocation) 主题模型[39]训练得到的文档主题概率分布, 其中主题数设置为 100. 不同模型在导语生成任务上的实验结果如表 4 所示.

表 4 不同 TIG 图在导语生成任务上结果对比

模型	RG-1	RG-2	RG-L
PMI-TIG	53.22	37.16	50.07
TFIDF-TIG	50.31	35.91	49.68
LDA-TIG	51.63	37.39	50.44
TIG	53.39	38.50	51.90

从表 4 可以看出, 本文提出的基于句子共现的 TIG 图构造方法相比其他基于相似度方法取得了更好的性能. 可能的原因是句子共现关系实际上也可以看做关键词在句子大小窗口内的共现关系, 能够直接描述不同案件要素之间的关联关系. 基于 PMI 的方法则是统计关键词的局部共现关系, 与 TIG 模型取得了相近的性能. 基于 TF-IDF 和 LDA 的方法相比 TIG 图虽然有一定的性能下降, 但是总体上来讲性能仍优于表 3 中基于 Transformer 的方法和两阶段的方法. 这也说明基于 TF-IDF 和 LDA 表征文本能够一定程度上反映不同节点的关联程度, 尤其是 LDA 模型直接通过主题建模的方式来构造不同节点之间的关联关系, 因此相比 TF-IDF 模型取得了一定的性能提升. 但是通过先表征节点再构建 TIG 图的方法也会带来因为表征误差而导致的误差累加的问题, 其性能依赖于节点的表示质量. 因此本文认为直接计算不同的节点之间的共现关系是一种简单且有效的构图方法.

6.3 不同图节点初始化方法结果对比

本文提出一种基于 BERT 模型和适配器网络的节点初始化方法. 为了进一步验证不同初始化方法对模型性能

的影响, 表 5 中列举了不同的节点初始化方法在话题导语生成任务上性能的对比结果, 其中 Random 表示每个节点根据词表进行随机初始化, 然后基于平均池化得到节点的表征; Bi-LSTM 表示将每个节点看做一个输入序列, 基于双向长短期记忆网络 (bi-directional long short term memory, Bi-LSTM) 进行编码, 以模型最后时刻的输出作为节点的表征, 隐状态的维度设置为 512; Transformer 模型与 Bi-LSTM 相似, 利用 Transformer 对图节点进行初始化, 基于平均池化得到节点的表示, 隐状态的维度同样设置为 512; BERT 为本文提出的方法.

表 5　不同图节点初始化方法在导语生成任务结果对比

模型	RG-1	RG-2	RG-L
Random	46.75	33.74	44.91
BI-LSTM	51.13	37.26	50.47
Transformer	50.66	36.82	49.58
BERT	53.39	38.50	51.90

从表 5 可以看出, BERT 模型相比 Random、Bi-LSTM 和 Transformer 等方法取得了更好的性能, 尤其是相比 Random 在 RG-1, RG-2 和 RG-L 指标上分别提升了 6.64, 4.76 和 6.99. 这也说明引入 BERT 模型能够获得更好的初始化表征, 对于下游的摘要任务能够提供很好的支撑. 另外一个原因可能是相比 BERT 模型, 其他模型引入了更多的参数, 包括 Embedding 层和 Bi-LSTM 或 Transformer 编码层, 在一定程度上可能也会存在过拟合的风险, 这点在标题生成任务上更为明显. 另外 Bi-LSTM 和 Transformer 模型相比 Random 模型取得了更好的效果, 这也说明在随机词嵌入的基础上, 利用 Bi-LSTM 或者 Transformer 继续学习不同词之间的依赖关系能够生成更好的初始化表征. 因此本文认为, 将 BERT 等预训练模型引入到词嵌入层能够生成更好的节点初始化特征.

6.4　人工评价

本文进一步引入人工评价, 从生成摘要的流畅度、信息量以及真实性 3 个方面更精确的评价生成的话题摘要质量, 其中流畅度主要评价生成的话题摘要是否存在逻辑错误和语法错误; 信息量则主要关注生成摘要是否覆盖了原文的中心思想, 是否关注了原文中不重要的细节; 而真实性则反映生成摘要是否存在事实性错误以及是否和原文表达存在冲突. 为了进行评测, 本文随机从导语生成任务测试集中选择 100 条样本作为评测数据, 并选择 5 名研究生作为评测人员对不同模型生成的话题导语进行打分, 分值从 1–5, 其中 1 分表示得分最低, 而 5 分表示最优, 最终采用所有评测者的平均分作为样本的最终得分, 评价结果如图 4 所示.

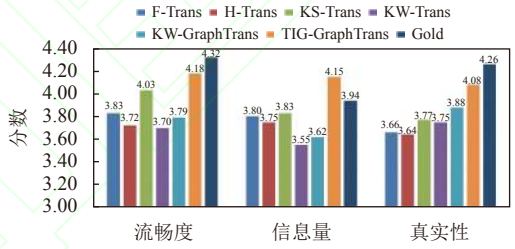


图 4　人工导语生成任务评价结果

从图 4 中可以看出, 在流畅度、信息量和真实性 3 个指标上, TIG-GraphTrans 模型均取得了较好的性能, 尤其是在信息量指标上取得了 4.15 分, 比标准摘要 (gold) 高出 0.21. 本文分析可能的原因有两个: 一是 TIG 模型能够选择重要信息并有效的过滤噪声数据, 因此生成的摘要能够反映案件话题的重要信息, 另外就是构造的案件话题摘要数据集存在噪声, 部分标准摘要包含有一些噪声信息, 如部分摘要可能含有“据网友微博爆料”“网页链接”等与案件话题无关的内容. 另外还可以看出, 基于关键词 KW-Trans 和 KW-GraphTrans 模型在流畅度、信息量指标上性能较差, 也可以反映出, 如果模型仅依赖于关键词来生成摘要容易忽视事件的整体信息, 因此导致无法生成高质量的话题导语. 在真实性方面, 基于两阶段的 KW-Trans 和 KS-Trans 以及基于图的 KW-GraphTrans 和 TIG-

GraphTrans 方法性能普遍优于 F-Trans 和 H-Trans, 这也说明在有噪数据集中, 先对数据进行选择和过滤是一种有效的摘要方法, 通过选择话题中的重要信息能够很好的避免生成事实错误的摘要.

6.5 实例分析

本文在表 6 中列举了不同的模型生成案件话题摘要的示例, 其中斜体加粗部分为抽取的关键词和关键句. 从表 6 可以看出, 该话题主要围绕“吴春红误判故意杀人罪”案件开展讨论, 重点聚焦在“河南省高院宣判撤销吴春红故意杀人罪判决改判无罪”. 不同的模型在话题标题生成任务中均生成了语义正确的话题标题, 本文提出的 TIG-Trans 将标准摘要中的“河南 16 年前投毒杀人案”替换为了“吴春红”, 从语义理解上更加准确. 在话题导语生成任务上, Flat-Trans 模型关注了“吴春红判处死缓”这一不重要的细节, 生成了事实错误的摘要, 认为“河南高院对吴春红的判决结果为死缓”. 出现这种原因可能原因是 Flat-Trans 直接截取话题评论前 500 个词作为生成摘要模型的输入, 仅包含话题评论的部分信息, 因此无法很好的了解整个话题的重要信息. 另外 KW-Trans 生成了不连贯、事实错误的导语, 在话题标题生成上却较为准确, 这也说明生成摘要模型仅依赖于抽取关键词, 失去了上下文语境, 因此模型无法很好的生成复杂的话题导语. 本文提出的 TIG-Graph 方法则通过引入关键词来发现话题评论的重要信息, 同时利用关键句来增强关键词的语义表征, 因此能够生成了语义正确、连贯、事实正确的话题导语.

表 6　不同基线模型生成话题标题和话题导语示例

话题评论(部分)	(1)2004年11月15日, 河南省民权县周岗村两名孩童因“毒鼠强”中毒, 一死一伤. 吴春红被认定为嫌疑人, 因琐事“投毒报复”.2005年6月23日至2007年10月30日, 商丘中院三次以故意杀人罪, 判处吴春红死缓. 河南高院三次以“事实不清”为由, 发回重审. 多年来, 吴春红及其家属持续申诉. 入狱以来, 吴春红一直拒绝认罪、拒绝减刑. (2)4月1日, 在河南高院开庭宣判, 原判决被撤销, 改判吴春红无罪, 当庭释放.2004年11月15日, 河南民权县周岗村两名孩童“毒鼠强”中毒, 一死一伤, 吴春红被认定为因琐事投毒报复的凶手, 因故意杀人罪3次被判死缓, 1次被判无期徒刑. (3)人生有几个16年, 今天终于是无罪释放, 可是这16年走过的日日夜夜, 心中的滋味只有当事人最清楚…… 他在一部分的经历留给我们九个字的思考: 一件事, 一挥间, 一辈子. (4)16年前, 河南省民权县两幼童鼠药中毒, 一死一伤, 吴春红被认定为凶手, 羁押至今…… 本判决为终审判决, 撤销此前的刑事裁定和刑事附带民事判决. 吴春红代理律师李长青对判决结果满意, 他表示吴春红或申请国家赔偿. (5)16年的青春无论怎样都不可能重来了, 多少补偿金都换不回一个人的人生…… 让一个无罪的人在牢狱中度过了16年, 如果不还受害人一个公道, 如何让人相信法律的公正？法律又该如何保护受害人的权益？ (6)有几个16年可以等待, 16年的青春年华啊！谁来买单已经不重要了, 还了清白！被冤枉了, 自己和亲人承受着巨大的心理压力. (7)三次被判死缓, 三次被发回重审, 一次被判无期, 一次被维持原判, 吴春红终于在今天的再审中被宣判无罪, 他的儿子表示后续会申请国家赔偿.
标准摘要	[话题标题] 河南16年前投毒杀人案再审改判无罪 [话题导语] 4月1日, “吴春红投毒案”今日在河南高院开庭宣判. 河南省高级人民法院的判决书显示, 吴春红被宣判无罪, 当庭释放.
Flat-Trans	[话题标题] 河南高院撤销宣判“吴春红投毒案” [话题导语] 4月15日, 河南省高院吴春红以故意杀人罪判决, 再审改判嫌疑人吴春红死缓.
KW-Trans	[话题标题] 河南16年前吴春红投毒案被判无罪 [话题导语] 河南省民权孩子死亡, 河南高院吴春红投毒案因故意杀人罪三次被判16年死缓.
KS-Trans	[话题标题] 河南16年前投毒杀人案当庭释放 [话题导语] 吴春红投毒案河南高院宣判, 改判无罪释放, 他表示冤枉他的人应道歉.
TIG-Trans	[话题标题] 吴春红再审被判无罪 [话题导语] 16年前“吴春红投毒案”河南高院宣判, 吴春红改判无罪.

本文进一步在表 7 中列举了 2 个缺陷样例来说明本文方法存在的局限性, 话题导语表示该例子的标准摘要, TIG-Trans 表示本文提出方法生成的导语. 从第 1 个例子可以看出, 在某些情况下, TIG-Trans 容易生成流畅度较低的摘要, 如重复出现“刘女士家暴跳楼”等内容. 可能的原因是该案件话题的用户讨论比较发散, 在构建的主题交互图上过分的强调“刘女士”“家暴”“跳楼”等相关的案件要素, 且由于不相关的内容过多, 案件要素之间的关系构建

比较稀疏, 模型退化为基于关键词的摘要生成方法, 模型更容易复制案件要素等重要内容, 导致流畅度降低. 另外从第 2 个例子中还可以看出, 本文模型有时会生成一些事实错误的摘要, 如在原文的评论中从未出现“吴军豹被判 2 年 7 个月”这样的内容, 而基于主题交互图的摘要模型生成了错误的内容. 鉴于此, 在后续的研究中可以进一步考虑更好的利用案件要素以及相关评论句等信息, 构建更好的关联图构造方法, 提升模型的性能.

表 7　本文模型生成的缺陷样例

话题导语	7月24日, 河南省商丘市柘城县人民检察院负责该案件的李姓负责人称, 刘女士因遭到丈夫殴打被迫从楼上跳下, 因此跳楼行为和丈夫殴打行为之间存在间接联系. 该院已于6月15日以故意伤害罪对窦某提起公诉.
TIG-Trans	近日, 河南商丘, 刘女士遭刘女士家暴跳楼女子遭刘女士家暴跳楼, 并在公诉中把刘女士跳楼行为列为该案件的加重情节, 建议法院从重量刑.
话题导语	7月3日, 江西南昌, 豫章书院非法拘禁案第二次开庭. 受害人罗伟称, 等这天已等了七年, 要吴军豹公开道歉, 给受害者公正的真相.2017年, 豫章书院被曝虐待学生后停办. 此后, 罗伟等学生提出刑事附带民事诉讼.
TIG-Trans	7月29日, 江西南昌, “豫章书院”非法拘禁案一审宣判, 吴军豹被判2年7个月.

7 结　论

本文提出一种案件相关话题的文本摘要任务并从新浪微博抓取相关数据, 构建了案件话题摘要数据集. 针对该任务中用户评论长度差异大、噪声数据多等问题, 提出一种基于图的两阶段话题摘要生成方法, 通过利用案件要素及相关的关键句来构建主题交互图, 实现噪声数据的过滤和重要信息的选择. 相较传统的多文档摘要方法, 主题交互图能够有效的建模评论中重要的案件要素之间的关联关系, 从而让模型更好的理解案件话题. 在话题标题生成和话题导语生成两个任务上的实验结果也表明, 在案件话题摘要任务中引入主题交互图能够在保留重要信息的基础上有效降低复杂文档的处理难度, 从而有效的改善摘要的生成质量.

在下一步研究中, 本文拟继续提升案件话题摘要数据集的质量, 探索构建包含案件要素、评论句及评论文档的异构主题交互图来更好的表达案件要素和关键句之间的关联关系, 进一步探讨如何引入更好的主题交互图编码方法来提升摘要的生成质量.

References:

[1] See A, Liu PJ, Manning CD. Get to the point: Summarization with pointer-generator networks. In: Proc. of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver: Association for Computational Linguistics, 2017. 1073–1083. [doi: 10.18653/v1/P17-1099]

[2] Zhou QY, Yang N, Wei FR, Zhou M. Selective encoding for abstractive sentence summarization. In: Proc. of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver: Association for Computational Linguistics, 2017. 1095–1104. [doi: 10.18653/v1/P17-1101]

[3] Zhang JQ, Zhao Y, Saleh M, Liu PJ. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In: Proc. of the 37th Int'l Conf. on Machine Learning. New York: PMLR, 2020. 11328–11339.

[4] Celikyilmaz A, Bosselut A, He XD, Choi Y. Deep communicating agents for abstractive summarization. In: Proc. of the 2018 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. New Orleans: Association for Computational Linguistics, 2018. 1662–1675. [doi: 10.18653/v1/N18-1150]

[5] Liu Y, Lapata M. Hierarchical transformers for multi-document summarization. In: Proc. of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: Association for Computational Linguistics, 2019. 5070–5081. [doi: 10.18653/v1/P19-1500]

[6] Jin HQ, Wang TM, Wan XJ. Multi-granularity interaction network for extractive and abstractive multi-document summarization. In: Proc. of the 58th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2020. 6244–6254. [doi: 10.18653/v1/2020.acl-main.556]

[7] Li W, Xiao XY, Liu JC, Wu H, Wang HF, Du JP. Leveraging graph to improve abstractive multi-document summarization. In: Proc. of the 58th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2020. 6232–6243. [doi: 10.18653/v1/2020.acl-main.555]

[8] Gehrmann S, Deng YT, Rush A. Bottom-up abstractive summarization. In: Proc. of the 2018 Conf. on Empirical Methods in Natural

Language Processing. Brussels: Association for Computational Linguistics, 2018. 4098–4109. [doi: 10.18653/v1/D18-1443]

[9] Nallapati R, Zhai FF, Zhou BW. SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. In: Proc. of the 31st AAAI Conf. on Artificial Intelligence. San Francisco: AAAI Press, 2017. 3075–3081.

[10] Liu PJ, Saleh M, Pot E, Goodrich B, Sepassi R, Kaiser L, Shazeer N. Generating wikipedia by summarizing long sequences. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: OpenReview.net, 2018.

[11] Chen YC, Bansal M. Fast abstractive summarization with reinforce-selected sentence rewriting. In: Proc. of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne: Association for Computational Linguistics, 2018. 675–686. [doi: 10.18653/v1/P18-1063]

[12] Li CL, Xu WR, Li S, Gao S. Guiding generation for abstractive text summarization based on key information guide network. In: Proc. of the 2018 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. New Orleans: Association for Computational Linguistics, 2018. 55–60. [doi: 10.18653/v1/N18-2009]

[13] Hou LW, Hu P, Cao WL. Automatic Chinese abstractive summarization with topical keywords fusion. Acta Automatica Sinica, 2019, 45(3): 530–539 [doi: 10.16383/j.aas.c170617]

[14] Narayan S, Cohen SB, Lapata M. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In: Proc. of the 2018 Conf. on Empirical Methods in Natural Language Processing. Brussels: Association for Computational Linguistics, 2018. 1797–1807. [doi: 10.18653/v1/D18-1206]

[15] Huang YX, Yu ZT, Guo JJ, Yu ZQ, Xian YT. Legal public opinion news abstractive summarization by incorporating topic information. International Journal of Machine Learning and Cybernetics, 2020, 11(9): 2039–2050. [doi: 10.1007/s13042-020-01093-8]

[16] Cao ZQ, Li WJ, Li SJ, Wei FR. Retrieve, rerank and rewrite: Soft template based neural summarization. In: Proc. of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne: Association for Computational Linguistics, 2018. 152–161. [doi: 10.18653/v1/P18-1015]

[17] Wang K, Quan XJ, Wang R. BiSET: Bi-directional selective encoding with template for abstractive summarization. In: Proc. of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: Association for Computational Linguistics, 2019. 2153–2162. [doi: 10.18653/v1/P19-1207]

[18] Wu RS, Wang HL, Wang ZQ, Zhou GD. Short text summary generation with global self-matching mechanism. Ruan Jian Xue Bao/Journal of Software, 2019, 30(9): 2705−2717. http://www.jos.org.cn/1000-9825/5850.htm [doi: 10.13328/j.cnki.jos.005850]

[19] Johnson AEW, Pollard TJ, Greenbaum NR, Lungren MP, Deng CY, Peng YF, Lu ZY, Mark RG, Berkowitz SJ, Horng S. MIMIC-CXR-JPG, a large publicly available database of labeled chest radiographs. arXiv: 1901.07042, 2019.

[20] He RF, Duan XY, Zhang XF, Zhao WL. Topic oriented twitter summarization based on social media content and network topology. Chinese Journal of Computers, 2019, 42(6): 1174–1189 [doi: 10.11897/SP.J.1016.2019.01174]

[21] Mihalcea R, Tarau P. Textrank: Bringing order into text. In: Proc. of 2004 Conf. on Empirical Methods in Natural Language Processing. Barcelona: Association for Computational Linguistics, 2004. 404–411.

[22] Wan XJ. Towards a unified approach to simultaneous single-document and multi-document summarizations. In: Proc. of the 23rd Int'l Conf. on Computational Linguistics. Beijing: Tsinghua University Press, 2010. 1137–1145.

[23] Zhao JM, Liu M, Gao LX, Jin Y, Du L, Zhao H, Zhang H, Haffari G. SummPip: Unsupervised multi-document summarization with sentence graph compression. In: Proc. of the 43rd Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. New York: ACM, 2020. 1949–1952. [doi: 10.1145/3397271.3401327]

[24] Yasunaga M, Zhang R, Meelu K, Pareek A, Srinivasan K, Radev D. Graph-based neural multi-document summarization. In: Proc. of the 21st Conf. on Computational Natural Language Learning. Vancouver: Association for Computational Linguistics, 2017. 452–462. [doi: 10.18653/v1/K17-1045]

[25] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. In: Proc. of the 5th Int'l Conf. on Learning Representations. Toulon: OpenReview.net, 2017.

[26] Antognini D, Faltings B. Learning to create sentence semantic relation graphs for multi-document summarization. In: Proc. of the 2nd Workshop on New Frontiers in Summarization. Hong Kong, China: Association for Computational Linguistics, 2019. 32–41. [doi: 10.18653/v1/D19-5404]

[27] Wang DQ, Liu PF, Zheng YN, Qiu XP, Huang XJ. Heterogeneous graph neural networks for extractive document summarization. In: Proc. of the 58th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2020. 6209–6219. [doi: 10.18653/v1/2020.acl-main.553]

[28] Tan JW, Wan XJ, Xiao JG. Abstractive document summarization with a graph-based attentional neural model. In: Proc. of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver: Association for Computational Linguistics, 2017.

1171–1181. [doi: 10.18653/v1/P17-1108]

[29] Huang LY, Wu LF, Wang L. Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. In: Proc. of the 58th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2020. 5094–5107. [doi: 10.18653/v1/2020.acl-main.457]

[30] Li W, Xu JJ, He YC, Yan SL, Wu YF, Sun X. Coherent comments generation for Chinese articles with a graph-to-sequence model. In: Proc. of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: Association for Computational Linguistics, 2019. 4843–4852. [doi: 10.18653/v1/P19-1479]

[31] Yao SW, Wang TM, Wan XJ. Heterogeneous graph transformer for graph-to-sequence learning. In: Proc. of the 58th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2020. 7145–7154. [doi: 10.18653/v1/2020.acl-main.640]

[32] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis: Association for Computational Linguistics, 2019. 4171–4186. [doi: 10.18653/v1/N19-1423]

[33] Houlsby N, Giurgiu A, Jastrzebski S, Morrone B, de Laroussilhe Q, Gesmundo A, Attariyan M, Gelly S. Parameter-efficient transfer learning for NLP. In: Proc. of the 36th Int'l Conf. on Machine Learning. Long Beach: PMLR, 2019. 2790–2799.

[34] Zhang Y, Yu ZT, Mao CL, Huang YX, Gao SX. Correlation analysis of law-related news combining bidirectional attention flow of news title and body. Journal of Intelligent & Fuzzy Systems, 2021, 40(3): 5623–5635. [doi: 10.3233/JIFS-201162]

[35] Lin CY. Rouge: A package for automatic evaluation of summaries. In: Proc. of Post-Conf. Workshop of ACL 2004. Barcelona: Association for Computational Linguistics, 2004. 74–81.

[36] Ott M, Edunov S, Baevski A, Fan A, Gross S, Ng N, Grangier D, Auli M. Fairseq: A fast, extensible toolkit for sequence modeling. In: Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics. Minneapolis: Association for Computational Linguistics, 2019. 48–53. [doi: 10.18653/v1/N19-4009]

[37] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: MIT Press, 2017. 6000–6010. [doi: 10.5555/3295222.3295349]

[38] Nallapati R, Zhou BW, dos Santos C, Gulçehre Ç, Xiang B. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In: Proc. of the 20th SIGNLL Conf. on Computational Natural Language Learning. Berlin: Association for Computational Linguistics, 2016. 280–290. [doi: 10.18653/v1/K16-1028]

[39] Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. Journal of Machine Learning Research, 2003, 3: 993–1022.

附中文参考文献:

[13] 侯丽微, 胡珀, 曹雯琳. 主题关键词信息融合的中文生成式自动摘要研究. 自动化学报, 2019, 45(3): 530–539. [doi: 10.16383/j.aas.c170617]

[18] 吴仁守, 王红玲, 王中卿, 周国栋. 全局自匹配机制的短文本摘要生成方法. 软件学报, 2019, 30(9): 2705−2717. http://www.jos.org.cn/1000-9825/5850.htm [doi: 10.13328/j.cnki.jos.005850]

[20] 贺瑞芳, 段兴义, 张雪菲, 赵文丽. 基于社会媒体内容和网络拓扑的特定话题推特摘要研究. 计算机学报, 2019, 42(6): 1174–1189. [doi: 10.11897/SP.J.1016.2019.01174]

黄于欣(1983－), 男, 博士, 副教授, 主要研究领域为自然语言处理, 文本摘要.

余正涛(1970－), 男, 博士, 教授, CCF 高级会员. 主要研究领域为自然语言处理, 神经机器翻译, 信息检索.

郭军军(1987－), 男, 博士, 副教授, CCF 专业会员. 主要研究领域为自然语言处理, 神经机器翻译, 多模态情感分析.

于志强(1983－), 男, 博士生, 主要研究领域为自然语言处理, 神经机器翻译.

高凡雅(1997－), 女, 硕士生, 主要研究领域为自然语言处理.