一种采用关系异构图神经网络的实体引导文本摘要框架

Jingqiang Chen 计算机科学学院，南京邮电大学，中国南京
cjq@njupt.edu.cnORCID: 0000-0001-7242-0141

摘要

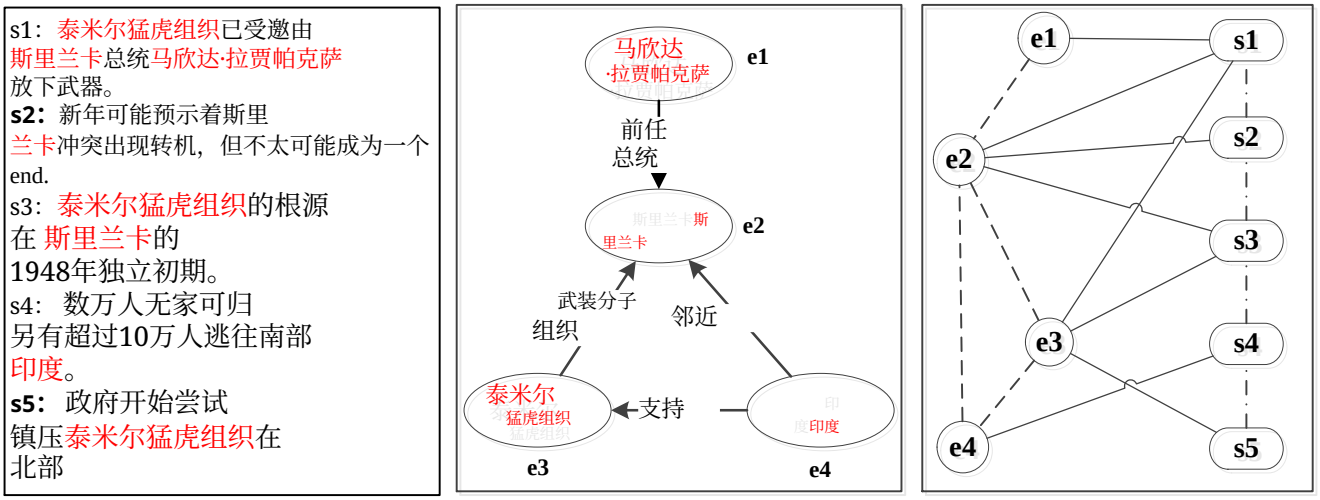
文本摘要要生成忠实的摘要，需要解决两个关键问题：利用文本之外的知识，以及利用文本中的跨句关系。针对这两个问题，直观的解决方式分别是知识图谱（KG）和图神经网络（GNN）。实体既是文本中的语义单元，也是知识图谱中的语义单元。本文利用文本中提及的实体连接GNN与KG，以同时解决这两个问题并实现文本摘要。首先，利用实体构建带权多类型边的句子-实体图，以建模句子关系，并提出一种用于摘要生成的关系异构图神经网络来计算节点编码。其次，利用实体将图连接到KG，以收集知识。第三，实体引导一个两步摘要框架，该框架定义多任务选择器以选择显著句子和实体，并使用实体导向抽象器压缩句子。通过构建句子-实体图将GNN与KG连接起来：基于KG构建实体-实体边，在KG上初始化实体嵌入，并利用实体-实体边训练实体嵌入。关系异构图神经网络同时利用GNN中的边权重和边类型，计算带权多类型边图。实验表明，在CNN/DM上，所提出的方法优于包括基于HGNN的HGNNSum在内的抽取式基线，以及包括实体驱动的SENECA在内的抽象式基线；在NYT50上，该方法优于大多数基线方法。子数据集上的实验表明，句子-实体边的密度会显著影响所提出方法的性能。密度越大，性能越好。消融实验表明了该方法的有效性。

关键词 摘要生成·图神经网络·知识图谱·实体

1 引言

自动文本摘要旨在将长文本提炼为简洁的摘要，以便快速获取信息 [1]。抽取式摘要直接从源文档中提取显著句子作为摘要。生成式摘要能够生成源文档中未出现的文本

。近年来，文本摘要的发展得益于深度学习技术的应用 [2-6]。这些方法主要采用编码器-解码器框架：编码器以不同形式对源文本进行编码，解码器则对句子进行标注或生成。



(a) 新闻文本的一部分　　(b) 知识图谱的一部分　　(c) 构建的句子-实体图 包含三种加权边

图 1　取自 CNN 的斯里兰卡泰米尔猛虎组织示例　　n

生成忠实摘要的摘要生成面临两个关键问题：如何利用文本之外的知识 [1], ，以及如何利用文本中的跨句关系 [2]。图 1 展示了一个示例。图 2(a) 中有五个句子。句子 s1 和 s5 相距较远，而 s1 和 s2 之间存在跨句关系，因为它们包含相同的实体——泰米尔猛虎组织。图 1(b) 是关于句子中识别出的实体的知识图谱的一部分。该知识图谱在新闻文本之外构建，包含文本之外的知识。图 1(a) 所示的跨句关系和图 1(b) 所示的实体知识对摘要生成都很重要。一种可行的利用实体知识和跨句关系的方法是构建如图 1(c) 所示的句子-实体图，并将该图与知识图谱关联起来。

对于知识问题，摘要生成所需的知识往往超出文本范围 [1, 7, 8], ，而现有研究大多聚焦于文本本身。知识图谱是文本之外知识的一种重要形式。文本中提及的实体可以与知识图谱中的实体建立关联，从而利用文本内和文本外的知识。文本中的实体包含关键词等知识以及文本中的结构化知识，可用于改进摘要生成。例如，使用实体作为关键词来生成实体连贯摘要 [3], ，以及使用实体作为桥梁对句子进行聚类，为较长的中文文章生成摘要 [9]。为了利用知识图谱中的实体来改进摘要生成，

一项初步研究只是将基于知识图谱训练的实体嵌入注入编码器-解码器架构[10]。近期研究则利用知识图谱中的实体来改进预训练语言模型 [11, 12]。这些既有研究利用文本中的实体或知识图谱中的实体进行摘要生成。如何有效地将文本中的结构化知识与知识图谱中的知识联系起来，以改进摘要生成，仍然是一项挑战。

针对跨句关系问题，对这些关系进行建模对于提取值得纳入摘要的句子至关重要，这些句子还可以进一步生成抽象式摘要。早期的传统工作，如 LexRank [13] 和 TextRank [14] ，利用句间余弦相似度构建文本图。近年来的研究基于图神经网络 [15] ，通过将跨句关系建模为图结构来捕捉长距离依赖。研究者可以构建各种类型的图结构，并提出图神经网络的不同变体来计算这些图。特别是在摘要生成中，由文本构建的图通常包含实值加权多类型边。近期一项研究构建了一个带有单一类型加权边的句子-词二部图，并提出异构图神经网络来计算节点编码 [2]。图神经网络还被用于对话摘要中的句内和句间关系建模 [16]。其他研究则依赖话语结构来构建摘要图 [17, 18]。对于具有无权多类型边的图，研究者提出了关系图神经网络（R-GNN）[19] 来计算这些图。边类型和边权重是基于图神经网络进行图计算时的重要信息，而文本中的实体则是用于图构建的丰富语义单元。当前的摘要生成研究尚未充分利用实体、边类型和边权重来构建和计算图，从而对跨句关系进行建模。

为综合解决摘要生成中的知识问题和跨句关系问题，本文提出了一种实体引导摘要生成框架，利用文本中提及的实体连接知识图谱和图神经网络，并利用图神经网络中的边权重和边类型计算带权多类型边图。

首先，利用文本中提及的实体构建并计算一个具有加权多类型边的句子-实体图，以对摘要生成中的句子关系进行建模。引入三种边类型，即句子-实体边、实体-实体边和句子-实体边，用于连接句子和实体。这些边具有不同的权重。本文提出用于摘要生成的关系异构图神经网络（R-HGNN），在传播过程中利用边类型和边权重对图进行计算。R-HGNN结合了传统图神经网络 [15] 利用边权重的优势，以及传统关系图神经网络[19] 利用边类型的优势。

其次，利用实体将句子-实体图与知识图谱连接起来，通过实体链接技术[20-22]利用文本之外的知识进行摘要生成。我们有理由认为，外部知识能够改进摘要生成。知识图谱在图神经网络中通过三种方式加以利用：根据维基百科网页中两个实体的共现次数构建实体-实体边；通过 RDF2Vec [23]，初始化知识库中的实体嵌入；以及利用实体-实体边对实体嵌入进行监督。

第三，实体引导两步式摘要生成方法。在句子—实体图网络上定义多任务目标，同时选择显著句子和实体。句子选择与实体选择可以相互促进，因为显著句子往往包含显著实体。随后，实体导向抽象器以显著实体为查询，对显著句子进行压缩。

本文的主要贡献总结如下：

- 据作者所知，本文首次利用实体连接图神经网络与知识图谱，以建模文本中的跨句关系以及文本之外的知识，从而实现摘要生成。通过构建句子-实体图（其中实体-实体边基于知识图谱构建）、在知识图谱上初始化实体嵌入，并利用实体-实体边训练实体嵌入，将图神经网络与知识图谱连接起来。

- 本文提出了用于摘要生成的关系异构图神经网络，通过在传播过程中同时利用边权重和边类型，计算带权多类型边图。

- 在不使用预训练语言模型的情况下，所提出的方法在 CNN/DM 数据集上优于所有现有基线方法，并在 NTY50 数据集上优于大多数基线方法。对子数据集进行的实验表明，句子-实体边的密度会显著影响所提方法的性能。句子-实体边的密度越高，方法的性能越好。消融研究表明了所提方法的有效性。

2 相关工作

用于摘要生成的实体和知识图谱 实体在摘要生成中发挥着重要作用，可用于选择句子 [3, 24, 25]。连接句子的实体可用于提取连贯句子[26]。近年来，研究者通过计算实体上下文向量，并将其与句子上下文向量进行比较，以选择适合纳入摘要的句子；此外，还利用实体压缩提取出的句子，生成连贯的抽象式摘要 [3]。此外，从输入文档中提取的命名实体被用于构建实体—谓词—实体图，随后采用 graph2seq 方法生成摘要 [27]。知识图谱是文本之外的一种重要知识形式

，可用于摘要生成。另一项类似的工作是语义链接网络，其思想可追溯至 1998 年构建网络模型的研究，近期又扩展至信息物理社会空间，以便更好地对信息物理社会系统建模 [28]。以往大多数研究通过基于知识图谱训练实体嵌入来利用知识图谱。例如，通过 TransE 方法学习实体嵌入 [29] ，再将其注入 Transformer [30]，从而将来自知识图谱的实体级知识融入用于摘要生成的编码器-解码器架构 [10] 。随着预训练语言模型的发展，实体级知识也可用于改进 BERT [31] ，例如 ERNIE [11]。K-Bert 模型通过将三元组与句子中的实体相连来构建句子树，从而将领域知识注入 BERT [12]。基于知识图谱训练得到的这些实体嵌入反映了知识图谱中实体和关系的结构。对于摘要生成而言，文档中的跨句结构也很重要。如何结合利用知识图谱中的实体关系知识与文本中的跨句关系知识，是摘要生成领域一个很有前景的研究方向。本文提出通过基于知识图谱构建实体-实体边、在知识图谱上初始化实体嵌入，并利用实体-实体边训练实体嵌入，将知识图谱中的知识注入句子-实体图。

用于摘要生成的图神经网络。图神经网络最初是为同质图设计的，这类图中的节点类型相同，且边带有权重 [32, 33]。传统图神经网络的消息传播算法利用加权边，从邻接节点传播消息，以计算节点编码 [33]。原始图神经网络不考虑边类型。因此，研究者提出了关系图神经网络（R-GNN），通过为每种边类型配备一个变换函数来利用边类型[19]。原始关系图神经网络不考虑边权重。图神经网络是对文档结构建模、捕捉文本中的长距离关系以进行摘要生成的有效方法。例如，有研究通过构建句子图进行抽取式摘要：句子由循环神经网络编码，边权重根据话语关系指标的数量计算，随后应用传统图神经网络计算该图 [17]。基于关系图神经网络的摘要方法通过构建包含句子、词语和实体的图，并使用 NEXT、包含和指代关系[34]，以捕捉长文本中的长距离关系。话语感知神经摘要模型利用图神经网络编码基于修辞结构理论树和共指提及构建的结构化话语图，从而捕捉长程依赖 [18]。最近，研究者提出了适用于摘要生成 [2] 或其他应用 [35-40]、包含不同类型节点和边的异构图神经网络。对于摘要生成，异质图神经网络（HGNN）

通过构建仅含一种边类型、用于对跨句关系建模的句子-词二部图[2], , 并应用图注意力网络[32] 计算节点编码。对于其他应用，异构图神经网络被用于推荐[34], 或程序再识别[37]。对于大规模图，HinSAGE 是 GraphSAGE [38] 针对异构网络的扩展[41] , 包含三种聚合算法，即均值聚合、LSTM 聚合和池化聚合。大多数现有异构图神经网络适用于边取值为 0/1 的图（无权边）。为了处理带权多类型边图，在传播过程中应同时利用边权重和边类型。本文提出了用于摘要生成的关系异构图神经网络，通过结合传统图神经网络和传统关系图神经网络的优势，同时利用边权重和边类型。

两步摘要方法 随着深度学习技术的发展，抽取式摘要和生成式摘要都取得了长足进步。大多数工作都聚焦于基于循环神经网络 [4, 5, 42] 或 Transformer [6, 43]的编码器-解码器模型。近年来，为了衔接抽取式摘要与生成式摘要，两步摘要方法逐渐流行起来 [3, 44]。抽取步骤会挑选适合作为摘要的句子，而生成步骤则根据所选句子生成生成式摘要 [45-47]。一些工作采用强化方法衔接这两个步骤，并提升 ROUGE 增益 [3, 9, 48]。在 [3], 中，利用实体挑选用于生成摘要的句子。在 [9], 中，使用图神经网络挑选用于生成摘要的句子簇，具体做法是先将包含相同关键词的句子聚类，再连接这些簇。本文提出多任务选择器，用于挑选显著句子和显著实体，并采用实体导向抽象器压缩句子。

3 所提出的框架

目标是利用文本之外的知识和跨句关系，为输入文档生成抽取式摘要和抽象式摘要。输入文档记作 D，包含 M 个句子和 N 个实体。每个实体可能在文档中多次出现，并采用不同的提及形式。例如，图 1 示例中的实体“泰米尔猛虎组织”可能被称为 Tigers、泰米尔伊拉姆猛虎解放组织、LTTE、武装组织等。因此，文档中的一个实体表示为 {EntityName, MentionSet}，其中 EntityName 是实体名称，MentionSet 是该实体在文档中的提及集合。每个句子都是一个词语序列，每个实体提及也都是一个词语序列。

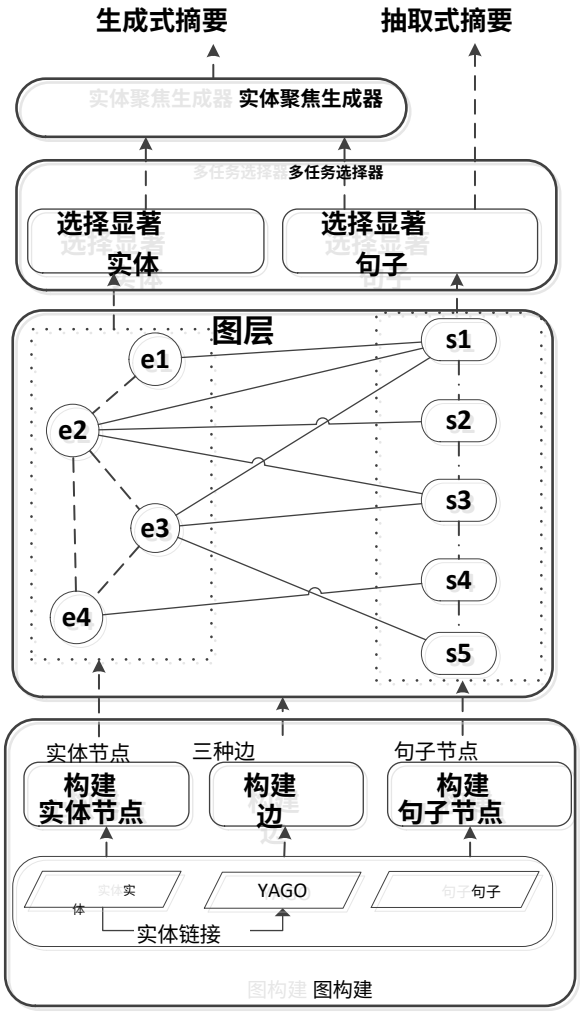


图 2 所提出的实体引导的摘要模型框架

为生成抽取式摘要和抽象式摘要，我们采用两步摘要方法：先选择显著句子和显著实体，再由生成器生成抽象式摘要。由于显著句子通常包含显著实体，因此我们为显著句子选择和显著实体选择定义了一个多任务目标。句子选择和实体选择均被表述为序列标注任务 [5], 。系统会预测一个标签序列 $y^S_1, y^S_2, ..., y^S_M$ ，其中 $y^S_i=1$ 表示第 i^{th} 个句子被选为摘要句子，$y^S_i=0$ 表示该句子未被选中。系统还会预测一个标签序列 $y^E_1, y^E_2, ..., y^E_N$ ，其中 $y^E_j=1$ 表示第 j^{th} 个实体被选为显著实体，$y^E_j=0$ 表示该实体未被选中。真实句子标签（称为 ORACLE）采用 [3], 中介绍的贪心方法获得；真实实体标签则通过收集真实人工生成式摘要中的实体获得。

抽象式摘要生成器是一个序列到序列网络，配备了注意力机制 [48] 和复制机制 [49], ，并使用选定的实体作为查询。

为利用外部知识和跨句关系，实体在所提出的框架中充当指导。实体指导句子-实体图的构建与计算，其中句子通过实体彼此连接，文本中提及的实体则与 YAGO2 [50] 连接，以收集用于摘要生成的外部知识。所提出的关系异构图神经网络（R-HGNN）用于计算句子-实体图的节点编码，以进行摘要生成。

图 2展示了所提出的框架。作为具体示例，图 2中的句子-实体图根据图 1中的示例构建而成。该图由实体节点和句子节点组成，这些节点通过三种类型的关系边相连，并在图 2中以不同形状表示。所提出的框架包括：1）图构建模块为输入文档构建并初始化句子-实体图（§3.1）；2）图层应用关系异构图神经网络来计算节点编码（§3.2）；3）多任务选择器通过使用多任务目标训练 R-HGNN，选择显著句子和实体（§3.3）；4）生成器以选定的句子和实体为输入，通过实体聚焦指针生成网络生成抽象式摘要（§3.4）。最后，强化学习连接器连接选择器和生成器（§3.5）。由于计算资源有限，未应用预训练编码器（即 BERT），这被视为未来的工作。此外，所提出的模型与基于BERT的模型正交。

3.1 为文档构建句子-实体图

给定一篇文档，构建异构句子-实体图。以往的工作 [2] 使用句子和词语作为节点，构建异构二部图，该图只有一种类型的边，即句子-词语边。本研究使用实体而非词语作为节点，原因有二：1）实体所包含的信息比词语更丰富；更重要的是，2）文本中提及的实体可以链接到知识图谱，从而利用文本之外的知识。然而，凡事都有两面性，实体在文本中的出现频率比词语低。因此，除了句子-实体边之外，还在句子-实体图中添加了句子-句子边和实体-实体边，如图 2 所示。每个句子在图中对应一个句子节点，文档中的每个实体在图中对应一个实体节点。节点和边的构建与编码如下所述。知识图谱在实体编码和实体-实体边的构建中发挥着重要作用。

构建实体节点 由于每个实体在文档中可能有多处提及，因此使用现成的命名实体识别工具识别输入文档中的实体提及，然后使用现成的共指消解工具对这些提及进行聚类。这两种工具均来自 Stanford CoreNLP [51]。具体而言，命名实体识别工具识别出的实体提及足够准确，但识别出的提及数量有限。因此，使用共指消解工具找出更多提及，并根据指称链对这些提及进行聚类。为平衡准确率和提及数量，仅使用包含命名实体识别工具所识别提及的指称链。一个提及簇对应一个实体。随后，构建实体节点并进行编码，如图 3 所示。

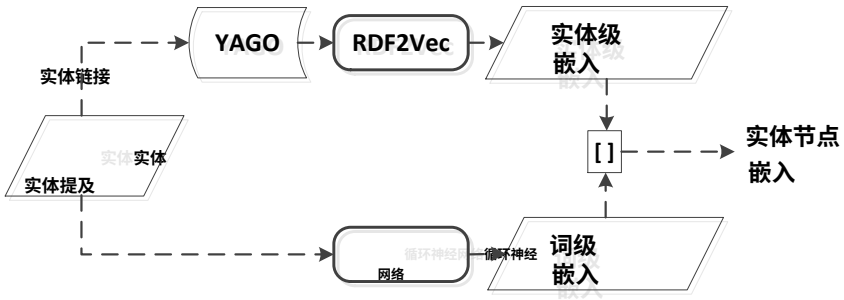


图 3 构建与编码实体节点

如图 3 所示，为了将句子-实体图与知识图谱连接起来，输入文档中的实体通过实体链接技术与 YAGO2 建立链接。YAGO2 是一个从维基百科提取的开源 知识库，包含大量实体和事实 [50]。实体链接旨在将文本中提及的实体链接到知识库中登记的实体。由于实体在文本中的提及具有歧义，实体链接是一项艰巨的任务。近年来，随着知识图谱和深度学习技术的发展，涌现出大量关于实体链接的研究 [20-22]。其中，本研究采用最先进的开源框架 AIDA-light [21] ，将每个实体提及簇链接到 YAGO2 中登记的实体。AIDA-light 是一个轻量级的集体实体链接框架，整合了三项特征：实体被提及的先验概率、提及上下文与候选实体之间的相似度，以及实体之间的连贯性。

在图 3 中，实体节点嵌入由词级嵌入和实体级嵌入拼接而成。1）词级实体嵌入反映实体在词语层面的字面含义。提及簇中的提及按照它们在输入文档中出现的顺序排列，然后使用特殊标记 <SEP> 作为分隔符，将它们拼接成一个序列。采用双向循环神经网络（双向循环神经网络）对每个提及序列进行编码。将最后一个前向隐藏状态和最后一个后向隐藏状态拼接，作为第 j^{th} 个实体的词级实体嵌入，记作

作为。2）实体级嵌入反映了知识图谱中的实体相关性。RDF2Vec [23] 用于从 YAGO2 学习实体级嵌入。RDF2Vec 首先通过图遍历将知识图谱转换为一组实体序列，然后使用这些序列训练神经语言模型，以估计实体序列在图中出现的可能性。习得的j^{th} 实体的实体级实体嵌入记为，而文档中所有实体的实体级嵌入记为矩阵E^E，其中第j^{th} 行等于 。由于 YAGO2 中的实体总数非常庞大，而且没有必要为 YAGO2 中的所有实体学习嵌入，因此本研究为固定大小的实体词汇表维护实体级实体嵌入，其中每个实体都在实验语料库中频繁出现。对于词汇表之外的实体，使用特殊实体 UNK。词级嵌入 和实体级嵌入 拼接后，作为句子-实体图中第j^{th} 个实体节点的初始编码，记为。以实体 Tamil Tigers 为例。它在文档中被提及为 Tigers、Liberation Tigers of Tamil Eelam、LTTE、The Militant Organization。这些提及通过<SEP> 连接起来，并使用双向循环神经网络进行编码，以获得词级嵌入。该实体也存在于 YAGO2 中，可通过 RDF2Vec 编码以获得实体级嵌入。词级嵌入和实体级嵌入拼接后，作为该实体的最终嵌入。通过这种方式，实体节点能够同时包含词级词汇信息和来自知识图谱的实体级信息。

构建句子节点 输入文档中的 M 个句子表示为 [s_1, s_2, s_3, ...，s_M]。句子 s_i 是由词语组成的序列，表示为 [s_{i1}, s_{i2}, s_{i3}, ..., $s_{i|si|}$]，其中 |si| 是第 i^{th} 个句子 s_i的长度。然后，采用双向循环神经网络对句子序列进行两级编码，如图4所示。公式 (1) 使用词级双向循环神经网络逐词编码每个句子

，公式 (2) 将最后一个前向隐藏状态与最后一个后向隐藏状态拼接，作为句子表示。公式 (3) 使用句子级双向循环神经网络，以句子表示为输入对句子序列进行编码，从而保留句子间的序列关系；随后，公式 (4) 将前向隐藏状态与后向隐藏状态拼接，作为初始句子节点表示。例如，图 1 (a) 中的五个句子首先分别通过公式 (1) 和 (2) 进行编码，然后依次通过公式 (3) 和 (4) 进行编码，以获得最终的句子表示。

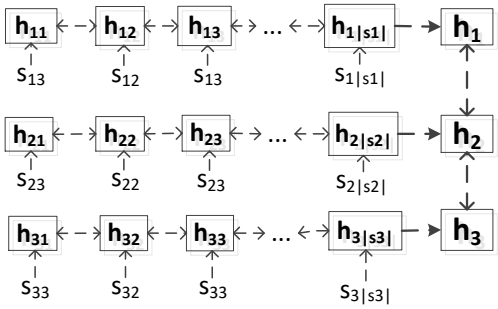


图4 句子编码

$$[\vec{h}_{i1},\vec{h}_{i2},...,\vec{h}_{i|si|};\overleftarrow{h}_{i1},\overleftarrow{h}_{i2},...,\overleftarrow{h}_{i|si|}]=BiRNN1([s_{i1},s_{i2},...,s_{i|si|}]) \quad (1)$$

$$s\_rep_i=[\vec{h}_{i|si|},\overleftarrow{h}_{i1}] \quad (2)$$

$$[\vec{h}_1,\vec{h}_2,...,\vec{h}_M;\overleftarrow{h}_1,\overleftarrow{h}_2,...,\overleftarrow{h}_M]=BiRNN2([s\_rep_1,s\_rep_2,...,s\_rep_M]) \quad (3)$$

$$S_i^{(0)}=[\vec{h}_i,\overleftarrow{h}_i] \quad (4)$$

构建边句子-实体图中有三种边。1）如果某个句子包含某实体的任何提及，则在句子节点和实体节点之间构建句子-实体（简称 SE）边。SE 边反映句子与实体之间的包含关系，其权重设为该实体在句子中的出现次数。2）在输入文档中每两个相邻句子之间构建句子-句子（简称 SS）边，因此，包含 M 个句子的文档的图中有M−1 条 SS 边。SS 边反映句子之间的序列关系，其权重设为 1。3）如果两个实体在同一个维基百科网页中共同出现，则在这两个实体之间构建实体-实体（简称 EE）边。EE 边基于外部知识库构建，与基于输入文档本身构建的另外两种边不同。由于在 YAGO2 中注册的实体均来自维基百科，因此每个实体都对应一个维基百科网页。EE 边的权重设为两个实体共同出现的维基百科网页数量。幸运的是，AIDA 提供了维基百科中的实体-实体共现数据。如图 2 所示，实体 e2 和 e3 之间存在一条 EE 边，因为这两个实体出现在同一个维基百科网页中。如图 1 (a) 所示，句子 s3 包含实体 e2，因此句子 s3 与实体 e2 之间存在一条 SE 边。句子 s2 和 s3 是相邻句子，因此二者之间存在一条 SS 边。三种边类型分别表示三种正交关系，且均为无向边。

实体-实体边基于知识图谱构建，实体级实体嵌入也在知识图谱上初始化。在接下来的章节中，将使用实体-实体边对实体编码的训练进行监督。

3.2 用于摘要生成的关系异构图神经网络

本节介绍关系异构图神经网络，用于计算带有加权多类型边的句子-实体图的节点编码，以进行摘要生成。边类型和边权重都是图神经网络中消息传播的重要信息。传统图神经网络 [15]利用边权重计算具有加权单类型边的图，而传统关系图神经网络 [19] 利用边类型计算具有未加权多类型边的图。对于带有加权多类型边的句子-实体图，R-HGNN结合两种图神经网络的优势，同时利用边类型和边权重。R-HGNN采用图神经网络的传播算法进行边类型内传播，以计算节点编码；定义了一个类似关系图神经网络的边类型特定函数，用于将编码转换为边类型特定编码，并对编码进行聚合。

R-HGNN 的定义：假设输入图包含 m 个节点和 c 种边类型。边类型表示为 {ET_1, ET_2, ..., ET_c}。节点通过不同类型的加权边相连，每种边类型都对应一个独立的节点邻接矩阵。符号说明如下：

- 表示 R-HGNN 第 l 层的节点编码，而 表示初始节点嵌入。 $X^{(0)}$
- 表示边类型 ET_k对应的特定类型节点邻接矩阵，矩阵 中的元素是相应边的权重。 A^{ET_k}

R-HGNN 的目标是学习一个函数 其中 表示节点的高层隐藏特征， 包含边类型、边权重和图结构的信息。R-HGNN 有 L 层，且 。 公式（5）至（7）给出了 R-HGNN 的传播过程方程，用于计算第 l^{th} 层的节点编码。

$$X^{ET_k^{(l)}} = D^{ET_k^{-\frac{1}{2}}} A^{ET_k} D^{ET_k^{-\frac{1}{2}}} g^{ET_k}(X^{(l-1)}) \quad (5)$$

$$X^{self^{(l)}} = g^{self}(X^{(l-1)}) \quad (6)$$

$$X^{(l)} = \sigma\left(\sum_{k=1}^{c} X^{ET_k^{(l)}} + X^{self^{(l)}}\right) \quad (7)$$

方程（5）通过同时利用边权重和边类型来计算边类型特定的节点编码。首先，在传播过程中使用边权重，部分借鉴了

文献中提出的原始图神经网络[15]。与[15], 边类型特定矩阵 由度矩阵 归 A^{ET_k} 化，其中对角元素。这是因为直接使用边类型特定矩阵会改变节点编码的尺度。计算的理论依据见[15]。其次，传播过程中使用了边类型，部分借鉴了文献中提出的原始关系图神经网络的思想[19]。与[19], 每种边类型都配有一个边类型特定的变换函数，并选择线性变换函数及可训练权重矩阵 。节点编码经函数变换后，通过边类型特定的传播过程传递至邻接节点。图中添加了虚拟自边，以考虑传播过程中的自环，公式（6）则计算自边的节点编码。

$$W^{ET_k}$$

公式（7）计算 l^{th} 层中每个节点的编码，方法是累积该节点所有边类型特定编码，其中 是逐元素激活函数，例如 。 l^{th} 层中累积的编码用于在（l+1）th 层进行传播。最终 l^{th} 层中的编码将用作下一小节中节点的特征。

R-HGNN结合了原始图神经网络和原始关系图神经网络的优势，从而能够利用带权多类型边图中的边类型和边权重。在某种程度上，原始图神经网络和原始关系图神经网络可以分别视为不考虑边类型和不考虑边权重时 R-HGNN 的特殊情况。

将 R-HGNN 应用于句子-实体图。上一小节构建的句子-实体图包含三种加权边，即句子-实体边、句子-句子边和实体-实体边。该图包含 M 个句子节点和 N 个实体节点。所提出的 R-HGNN 可用于计算节点编码。输入形式化如下：

- 句子-实体边的邻接矩阵，其中 A^{ET_k} 和 A^{ET_k} 是 如第3.1节所述。
- 句子-实体边的邻接矩阵，其中 A^{ET_k} 的值设置为实体在句子中的出现次数（如果 i 和 j 是不同类型的节点，则设置为 0），如第3.1节所述。
- 实体-实体边的邻接矩阵，其中 A^{ET_k} 的值设为实体 i 和实体 j 在维基百科网页中共现的次数，前提是 i 和 j 都是

实体节点，否则设为 0，如第3.1节所述。

- ，初始节点嵌入矩阵，其中 是初始句子节点编码矩阵， 是初始实体节点编码矩阵，计算方法如第3.1节所述。 是 和 的拼接。

输出是 R-HGNN 的 L^{th} 层的节点编码，即句子节点编码和实体节点编码。以图 2 中的图为例。该图包含五个句子和四个实体。相邻矩阵 A^{SS}、A^{SE} 和 A^{EE} 可相应地计算，节点嵌入矩阵 $X^{(0)}$ 也可按照第3.1节所述进行初始化。R-HGNN 可应用于该图，以计算句子节点编码和实体节点嵌入。在下文小节中，将在 R-HGNN 的基础上定义一个多任务选择器，并使用这些编码作为特征。

3.3 多任务选择器

多任务选择器需要完成两项任务：从句子-实体图中选择显著句子节点和显著实体节点。这两项任务可以相互促进，因为显著句子通常包含显著实体。如图1(a)所示，首句是一个显著句子，其中包含显著实体泰米尔猛虎组织。显著句子和实体的选择会相互影响。所选句子构成抽取式摘要，而所选实体可用作下一小节介绍的实体聚焦生成器的查询。以R-HGNN最后一层的节点编码作为特征，多任务选择器的目标定义如下：

通过句子选择和实体选择共同进行监督，公式（8）通过对句子节点编码应用经MLP变换后的softmax函数，计算句子被选中的概率。公式（9）通过对实体节点编码应用经MLP变换后的softmax函数，计算实体被选中的概率。

$$p(\hat{y}_i^S = 1) \sim soft\max(MLP(S^{(L)})) \quad (8)$$

$$p(\hat{y}_j^E = 1) \sim soft\max(MLP(E^{(L)})) \quad (9)$$

采用交叉熵计算损失 $\hat{y}_i^S = 1$ 表示 i^{th} 句是摘要句子，而 表示该句不是摘要句子，因此

真实值中i^{th} 句子 被选中的概率为。 实体标签表示j^{th} 实体是摘要实体，而 表示该实体不是摘要实体，因此真实值中j^{th} 实体被选中的概率为。公式（10）和（11）分别计算句子损失和实体损失。

$$loss^{S} = CrossEntropy(y^{S}, \hat{y}^{S}) \quad (10)$$

$$loss^{E} = CrossEntropy(y^{E}, \hat{y}^{E}) \quad (11)$$

利用实体-实体边监督实体嵌入 实体嵌入由词级嵌入和实体级嵌入组成，其中实体级嵌入通过 RDF2Vec 初始化，以反映 YAGO2 中实体的全局相关性。由于嵌入在所提出的模型中是可训练的，如果实体嵌入的训练没有得到直接监督，实体之间的相关性信息就会在实体嵌入中丢失或减弱。此外，反映所构建的句子-实体图结构的局部实体相关性也应在实体嵌入中得到增强。可以合理地认为，保留实体嵌入中的实体相关性信息能够提升句子选择器和实体选择器的性能。

$$r_{i,j}^{EE} = \frac{A_{i,j}^{EE}}{\sum_{i,j} A_{i,j}^{EE}} \quad (12)$$

$$\hat{r}_{i,j}^{EE} \sim soft\max(E^{E} \times E^{E^{T}}) \quad (13)$$

$$loss^{EE} = CrossEntropy(r^{EE}, \hat{r}^{EE}) \quad (14)$$

由于实体-实体边反映了维基百科中的实体相关性信息，因此将实体-实体边作为实体相关性的真实值，用于监督实体嵌入的训练。首先，根据公式（12）对文档中任意两个实体之间的实体-实体边进行归一化，其中 是实体-实体边的邻接矩阵，矩阵中的元素 是实体 i 和实体 j 在维基百科网页中的共现次数，计算方法见第3.1节。仅考虑文档中的实体，原因有二：1）在同一文档中共现的实体具有更高的相关性，这会体现在实体嵌入中；2）使用较小的实体集进行监督，而非使用庞大的完整实体集，可以节省有限的计算资源并加快训练。其次，实体-

实体相关性通过 softmax 激活函数对公式（13）中实体级实体矩阵 与进行归一化，采用点积计算，其中 是输入文档中实体的实体级实体嵌入矩阵，按第3.1节所述方法计算。最后，公式（14）将真实值与预测的实体相关性之间的交叉熵作为损失。

多任务选择器的损失 多任务选择器的最终损失在公式（15）中定义，通过线性组合句子选择损失、实体选择损失和实体间相关性损失得到。超参数 和 分别根据经验设置为 0.42 和 0.33。

$$loss^{selector} = loss^{S} + \lambda^{E} * loss^{E} + \lambda^{EE} loss^{EE} \quad (15)$$

推理 在推理阶段，句子和实体分别按照 和进行排序，然后选择排名最高的句子 和实体。

3.4 实体聚焦生成器

实体聚焦生成器输入由多任务选择器选出的显著句子和显著实体，并生成抽象式摘要。实体聚焦生成器通过[49] 使用实体作为查询，扩展了当前最先进的指针生成网络。

句子按照其在原始文档中的出现顺序排列，并连接成一篇完整文本。然后，使用双向循环神经网络对文本进行编码。令 和 分别表示连接文本中第 i^{th} 个词的正向和反向隐藏状态。 表示连接文本的表征，其中 m 为文本的词数。令 表示第 i^{th} 个词的编码。

对于实体编码，仅采用词级实体嵌入（记作 ，用于表示第 j^{th} 个实体，e_j如第3.1节所述）来表示实体。然后，对实体编码进行平均池化，以获得显著实体集的编码，即 。

与原始指针生成网络一样，解码器按照如下所示的词汇表分布和词语的指针分布生成词语。

$$p(w_t) = p_{gen} p_{vocab}(w_t) + (1 - p_{gen}) \sum_{i:w_i = w_t} a_{t,i} \quad (16)$$

显著实体用于计算注意力 和生成概率 p_{gen}。设 h_t为t^{th} 解码步骤中的隐藏状态。 p_{gen} 计算如下：

$$a_t = soft\max(\alpha_t) \quad (17)$$

$$\alpha_{t,i} = v\tanh(W^{aD}h_t + W^{aT}h_i^T + W^{aE}h^E + b^{attn}) \quad (18)$$

$$p_{gen} = sigmoid(W^{pD}h_t + W^{pT}h_i^T + W^{pE}h^E + x_t + b^{attn}) \quad (19)$$

在这些公式中，实体编码作为参数加入，用于计算注意力和生成概率。通过这种方式，实体信息被纳入摘要生成过程。与指针生成网络一样，覆盖损失 cov_loss$_t$ 也被加入。

损失由预测词的负对数似然和覆盖损失组成，即。读者可参阅[49] 了解更多详情。

协方差 对数(( )) 协方差_ 生成器 t t t 损失 p w 损失

= − + λ

3.5 连接选择器与生成器

选择器负责选取显著句子和显著实体，而生成器则对它们进行压缩和改写。此前，两者分别训练，未采用任何形式的参数共享。为连接这两个网络，我们采用基于策略梯度的自我批评学习算法 [52] 来将它们连接起来。

按照马尔可夫决策过程的表述，在每个时间步，选择器从输入文档中采样句子和实体，生成器则利用这些句子和实体生成生成式摘要。我们将该摘要与真实摘要进行比较评估，并以 ROUGE-1 [53] 作为奖励。

在训练过程中， [52]，选择器从 和 第3.3节中介绍的分布中采样句子和实体。随后，生成器生成一份生成式摘要，并返回 Rouge-1 奖励，记为R。令$Sample^S$ 为输入文档中采样句子的索引集，令$Sample^E$为采样实体的索引集。对于每个， 设为 1，而对于，。对于每个， 设为 1， 而对于，。

S i 样本 iy'^S (′ =1 = ∑ ∈) $_S$S

S i i S k k样本y p y y

S i 样本 $_{ip}$ v (′^{S} =1 =0)

E j 示例 iy'^E (′ =1 = ∑ ∈) $_E$E j

E j E k kSample y p y y

E j 示例 $_{ip}$ v (′^{E} =1 =0)

与以往工作一样 [3], , 生成器的参数保持冻结。只有选择器通过强化学习进行训练。以下公式用于计算强化学习的损失。

$$loss^{RL} = R*(CrossEntropy(p(y'^{S}), p(\hat{y}^{S})) + \lambda^{E} CrossEntropy(p(y'^{E}), p(\hat{y}^{E}))) \quad (20)$$

如果选择器能够准确选择显著句子和实体，实体聚焦生成器就更有可能生成高质量的抽象式摘要，从而获得鼓励。反之，导致选择效果不佳的动作则会受到抑制。

公式（21）使用强化学习损失重新计算选择器的损失，其中超参数 经实验设定为 0.6。λ^{RL}

$$loss^{selector\_with\_RL} = loss^{selector} + \lambda^{RL} loss^{RL} \quad (21)$$

4 实验

4.1 语料库与预处理

使用两个常见的摘要数据集来评估所提出的模型：CNN/DailyMail（CNN/DM）数据集 [54] 和 NYT50 数据集 [55]。

对于 CNN/DM，标准数据集按 286649/13359/11490 个样例划分为训练集、验证集和测试集。我们遵循 [54] 中的预处理步骤来获取纯文本。为获取实体和提及，使用原始数据集提供的标注。对于每个样例中的每个实体，数据集都会提供实体 ID 以及所有提及的起始位置和结束位置。

NYT50 是《纽约时报》标注语料库的一个子集，由 [55] 预处理，用于文档摘要生成。该数据集包含 110540 篇带有摘要的文章，并划分为训练集 100834 篇和测试集 9706 篇。按照 [55], 中的预处理步骤，训练集的最后 4000 个样本用作验证集，测试样本则筛选至 3452 个。为获取实体和提及，使用 NYT50 提供的包含实体提及标注和共指标注的独立标注文件。标注文件中的共指 ID 用于对提及进行聚类，每个提及簇代表一个实体。标注文件还提供了实体类型。实验中使用地点类型、人物类型、组织类型和事件类型的实体。

表1 两个数据集的统计信息，其中 Doc.Sent、Sum.Sent、Doc.Ent、Sum.Ent、Sent.Men、Ent.YAGO 和 SE.Density 分别表示文档的平均句子数、摘要的平均句子数、文档的平均实体数、摘要的平均实体数、句子中的平均实体提及数、成功链接到 YAGO2 的实体平均数，以及句子-实体边的密度

	训练集	Dev	测试集	文档句子数	摘要句子数	文档实体数	摘要实体数	句子中的实体提及数	成功链接到YAGO2的实体数	SE.Density
CNN/DailyMail	286649	13359	11490	27.93	3.75	22.59	3.64	2.14	16.81	0.94
NYT50	100834	4000	3452	40.81	2.94	21.21	2.59	0.88	8.75	0.57

表1展示了两个数据集的统计信息。观察结果如下：

• 尽管 CNN/DM 文档中的句子数量少于 NYT50，但 CNN/DM 的实体数量多于 NYT50，这表明 CNN/DM 的实体密度高于 NYT50。
• CNN/DM 中的句子平均包含 2.14 个实体提及，而 NYT50 中的句子平均仅包含 0.88 个提及，这表明 CNN/DM 中的句子更有可能包含相同的实体。

• CNN/DM 中有 74% 的实体可以链接到 YAGO2，几乎是 NYT50 的两倍。
• 最重要的是，CNN/DM 中的句子-实体边比 NYT50 中密集得多。*SE*。密度定义如下。假设从某个文档构建的句子-实体图包含 *SE*。数量 条句子-实体边、*M* 个句子和 *N* 个实体，则该文档的 *SE*。密度 计算如下 。此处不计另外两种类型的边。*SE*。密度>=1 表示句子-实体二部图是连通图，而*SE*。密度<1 表示句子-实体二部图不是连通图。SE。密度越大，二部图的连通性越强。

$$\text{SE.Density} = \frac{\text{边数}}{\text{句子数量} + \text{实体数量}}$$

4.2 实现与参数设置

该模型使用 Python 中的 TensorFlow 实现。代码将在 GitHub 上发布，链接为 https://github.com/jingqiangchen/kbsumm。由于计算资源有限，词汇表限制为 40K 个词语，并使用 128 维 word2vec 嵌入进行初始化 [57]。CNN/DM 的实体词汇表限制为 500K 个实体，因为数据集中有超过 1000K 个实体，词汇表之外的实体将替换为特殊的未知实体 UNK。NYT50 的实体词汇表设为 146894，即 NYT50 中识别出的全部实体。实体词汇表使用 128 维 RDF2Vec 嵌入进行初始化 [23]。采用一个 GRU 单元层作为 RNN 单元。R-HGNN 的层数设为 2。图节点嵌入的维度设为 512。双向循环神经网络编码器的隐藏状态维度设为 256。循环神经网络解码器的隐藏状态维度设为 512。Adam 的参数设置为

[58]。批量大小设为 15。在 50K~80K 个训练步数内即可收敛。在 GTX-1080 TI GPU 显卡上训练 30k ~ 40k 个步数约需一天。

对于多任务选择器，排名前 k 的句子被选作抽取式摘要。对于 CNN/DM，k 设为 4；对于 NYT50，k 设为 3，因为如表1所示，CNN/DM 摘要中的平均句子数为 3.75，NYT50 摘要中的平均句子数为 2.94。同样，根据表1所示两个数据集摘要中的平均实体数，CNN/DM 中排名前 4 的实体和 NYT50 中排名前 3 的实体被选作显著实体。输入文档被截断至最多 100 个句子。输入文档的实体集被截断至最多 100 个实体。对于实体聚焦生成器，输入文本被截断至最多 150 个词语。生成器的解码步数限制为 100 步。

4.3 与现有方法比较

在 CNN/DM 和 NYT50 上，将十余种基线方法与所提出的方法进行比较。以下介绍所提出方法的两个版本以及三种表现强劲的现有基线方法。分析各数据集的评估结果时，还将介绍其他基线方法。

- RHGNNSumExt 通过从所提出的模型中移除实体聚焦生成器，构成了本研究提出的抽取式摘要方法。排名前 k 的句子将被提取为抽取式摘要，其中 k 取决于数据集。

- RHGNNSumAbs 这是所提出的完整模型：首先使用 RHGNNSumExt 提取排名前 k 的句子，然后通过实体聚焦生成器将这些句子改写为生成式摘要，并使用强化学习连接器连接选择器和生成器。
- HGNNSum 这是 [2] 中提出的抽取式摘要方法。它构建了一个
  输入文档的二部句子-词图；当句子包含相应词语时，便建立句子-词边。HGNNSum 将异构图神经网络应用于该图，并通过句子选择器直接选出显著句子作为摘要。

- SENECA 这是 [3] 中提出的由实体驱动的生成式摘要方法，旨在
  生成连贯摘要。该方法首先利用实体选择显著句子，然后应用一个基于强化学习的生成式摘要生成模块，以连贯性、简洁性和清晰性作为奖励，对句子进行压缩并生成最终摘要。

- ASGARD 这是 [27] 提出的一种生成式摘要方法。该方法从输入文档中构建实体-实体图，其中节点是从文档中提取的实体，边是谓词。实体-实体图通过图神经网络进行编码，文档则通过长短期记忆网络进行编码。随后，通过关注图编码和文档编码生成生成式摘要。构建的实体-实体图未与

外部知识图谱相连，且该方法未考虑句子-实体关系。研究比较了两个不使用强化学习的 ASGARD 版本。ASGARD-DOC 将输入文档作为整体进行编码，而 ASGARD-SEG 则将输入文档划分为一组段落，分别进行编码，再通过长短期记忆网络将它们合并。以下分别给出在 CNN/DM 和 NYT50 上的评估与分析。

CNN/DM 上的结果表 2 展示了 CNN/DM 数据集上的评估结果。第一部分是 LEAD-3 基线方法，该方法仅从文档中选择开头的三个句子；以及 ORACLE 上界，即按照第 3 节所述方法获得的真实抽取式摘要。第二部分是近年来发表的四种抽取式摘要方法。第三部分包括四种最先进的生成式摘要方法，包括指针生成网络、句子改写方法、基于强化学习的方法和自底向上方法。最后一部分包括两个强基线方法 SENECA 和 HGNNSum，以及本研究提出的方法。

表 2 使用完整摘要长度下的 ROUGE F1 对 CNN/DM 进行比较的结果

方法	Rouge-1	Rouge-2	Rouge-L
LEAD-3 [49]	40.34	17.70	36.57
ORACLE [59]	52.59	31.24	48.87
JECS [60]	41.70	18.50	37.90
长短期记忆网络+PN [42]	41.85	18.93	38.13
HER（无策略） [60]	41.70	18.30	37.10
HER（有策略） [61]	42.30	18.90	37.60
指针生成网络 [49]	39.53	17.28	36.38
SENTREWRITE [45]	40.88	17.80	38.54
DEEPREINFORCE [62]	41.16	15.75	39.08
自底向上 [47]	41.22	18.68	38.34
SENECA [3]	41.52	18.36	38.09
ASGARD-DOC [27]	40.38	18.40	37.51
ASGARD-SEG [27]	40.09	18.30	37.30
HGNNSum	42.31	**19.51**	38.74
RHGNNSumExt	**42.39**	19.45	**38.85**
RHGNNSumAbs	41.63	18.45	38.00

粗体数值表示最佳结果

根据表2，RHGNNSumExt 在所有抽取式和抽象式方法中取得了最高的 Rouge-1 和 Rouge-L 分数。具体而言，根据 Rouge-1 和 Rouge-L 分数，RHGNNSumExt 的表现优于 HGNNSum。这两个模型的主要区别有三点：1）RHGNNSumExt 构建了一个带有加权多类型边的句子-实体图，而 HGNNSumm 构建了一个仅含一种边类型的句子-词二部图；2）RHGNNSumExt 在传播过程中同时利用边权重和边类型来计算节点编码；3）RHGNNSumExt 通过将实体链接到

知识库，将来自 YAGO2 的外部知识注入图中，并训练实体嵌入以拟合实体-实体边。如表1所示，CNN/DM 中每个句子平均包含 2.14 个实体提及，句子-实体边的密度为 0.94，因此 CNN/DM 数据集的句子-实体图中有足够的句子-实体边，可供 R-HGNN 进行消息传播。然而，RHGNNSumExt 相较于 HGNNSum 的提升并不十分显著。这主要是因为实体链接是一项困难的任务，而且如第3.1节所述，可能出现链接缺失或错误链接。如表1所示，CNN/DM 的 22.59 个实体中，只有 16.81 个可以链接到 YAGO2，而这 16.81 个已链接实体中也可能存在错误链接。在一个高质量、由人工标注实体提及的标准小规模短新闻数据集上，最先进的实体链接方法 AIDA-light 的准确率约为 80% [21]。至于 CNN/DM，其中的新闻文章要长得多，提及标注的质量也低得多。

所提出的抽象式方法 RHGNNSumAbs 在所有抽象式方法中取得了最高的 Rouge-1 分数。具体而言，RHGNNSumAbs 的 Rouge-1 和 Rouge-2 分数均高于 SENECA。SENECA 是一种两步式实体驱动方法，先选择显著句子，再使用实体作为查询生成连贯的抽象式摘要。RHGNNSumAbs 和 SENECA 使用不同的内容选择方法。RHGNNSumAbs 利用基于图的模型来建模句子关系，并结合知识图谱进行内容选择。而 SENECA 则采用单层单向长短期记忆网络，以循环方式抽取显著句子。RHGNNSumAbs 的表现也优于 ASGARD-DOC 和 ASGARD-SEG。ASGARD 在生成式摘要生成之前不使用抽取器。抽取句子的质量极大地决定了生成式摘要的质量。

NYT50上的结果 表3展示了NYT50数据集上的评估结果。与 [55] 和 [2], 中使用的有限长度 ROUGE召回率相同，抽取的句子会截断至与人工撰写的摘要相同的长度，并使用召回率分数而非F1分数。表3的前两行是 [55] 报告的基线方法，接下来的两行分别是 [2]报告的LEAD-3基线方法和抽取式摘要的ORACLE上界。第二部分和第三部分分别报告了其他非BERT研究以及本研究所提出模型的性能。由于SENECA未在NYT50上报告，因此表3未显示SENECA的分数。

表3 NYT50数据集上各方法的比较结果（摘要长度为100个词时的ROUGE召回率）；PG模型的结果取自[59]，而“-”表示原论文未报告该结果

方法	Rouge-1	Rouge-2	Rouge-L
首句 [55]	28.60	17.30	-
前k个词 [55]	35.70	21.60	-
LEAD-3	38.99	18.74	35.35
ORACLE	60.54	40.75	57.22
COMPRESS [55]	42.20	24.90	-
SUMO [63]	42.30	22.70	38.60
指针生成网络 [49]	43.71	26.40	-
DRM [62]	42.94	26.02	-
HGNNSum	46.89	26.26	42.58
RHGNNSumExt	45.80	25.89	40.26
RHGNNSumAbs	44.30	24.33	37.03

所提出的模型RHGNNSumExt在NYT50数据集上的表现优于大多数基线方法。RHGNNSumExt的表现不如HGNNSum，这是因为如表1所示，NYT50中的句子-实体边和实体提及都远少于CNN/DM，导致NYT50中没有足够的句子-实体边供R-HGNN进行消息传播。具体而言，NYT50中的句子平均包含0.88个实体提及，SE.Density值为0.57，分别远低于CNN/DM中的2.18和0.94。此外，NYT50文档中成功链接到YAGO2的实体平均数量为8.75，也远低于CNN/DM的16.81；由于只有当两个实体在维基百科网页上共同出现时才会建立实体-实体边，因此NYT50中的实体-实体边也较少。下一小节将详细讨论SE.Density。

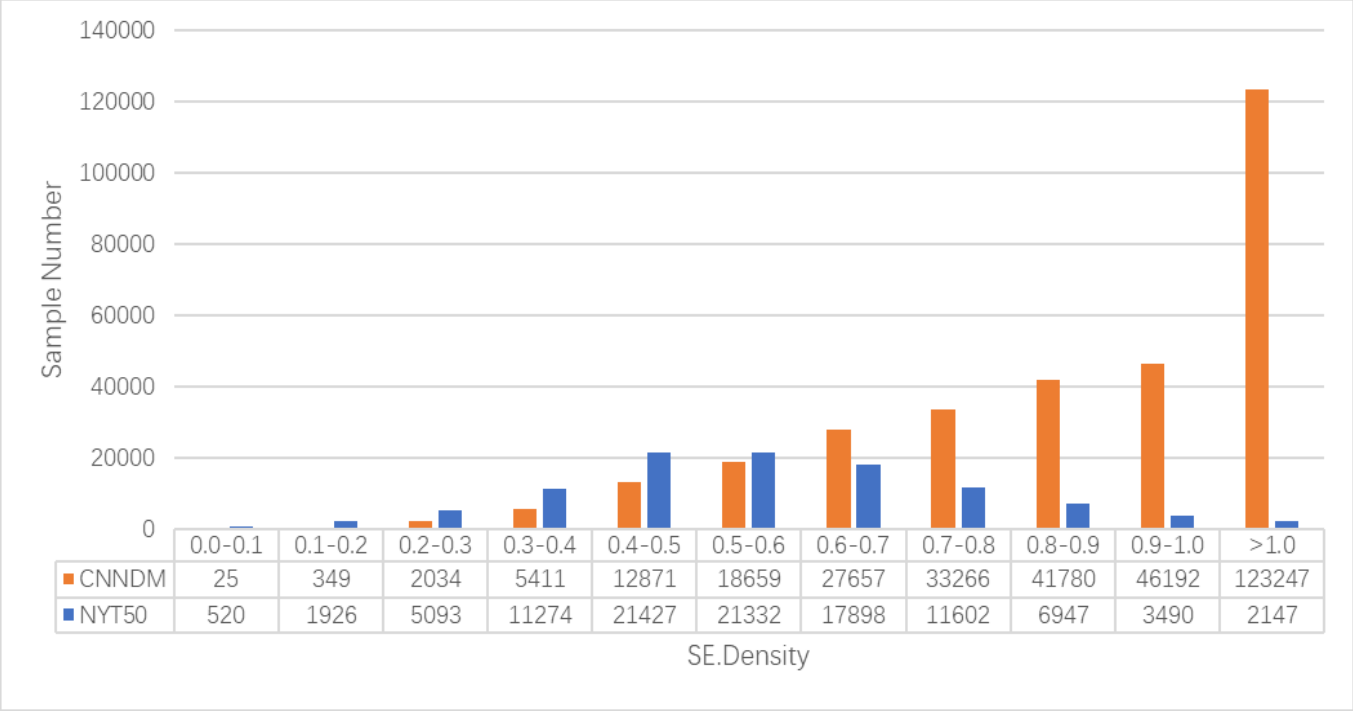
尽管如此，所提出的生成式方法 RHGNNSumAbs 在表3的 Rouge-1 得分方面优于所有抽象式基线和大多数抽取式基线。具体而言，RHGNNSumAbs 的 Rouge-1 得分高于指针生成网络（PG）。RHGNNSumAbs 是一种适用于 NYT50 数据集且具有竞争力的生成式摘要方法。

4.4 句子-实体边密度的讨论

如前一小节所示，所提出的方法在 SE.Density 值较高的 CNN/DM 上表现更好，而在 SE.Density 值较低的 NYT50 上表现较差，这表明数据集的 SE.Density 会影响所提出方法的性能。为了考察所提出的方法在 SE.Density 值不同的数据集上的表现，以及哪些 SE.Density 值适合使用该方法，我们在两个数据集的不同 SE.Density 值子集上进行了实验。

图 5 展示了 CNN/DM 和 NYT50 在不同 SE.Density 值下的样本分布。如图所示，这两个数据集的样本分布差异较大。将图 4 中 SE.Density 小于 0.7 的样本数量相加可知，NYT50 的样本中共有 79470 个符合条件，占 103656 个样本。相比之下，CNN/DM 中 SE.Density 大于 0.7 的样本共有 244485 个，占 311491 个样本。CNN/DM 中 SE.Density 介于 0.0 和 0.1 之间的样本有 25 个，而 NYT50 中有 520 个。CNN/DM 中 SE.Density 大于 1.0 的样本有 123247 个，而 NYT50 中有 2147 个。由此可见，CNN/DM 的平均 SE.Destity 为 0.94，NYT50 的平均 SE.Density 为 0.57。

对于 CNN/DM，通过分别选取 SE.Density 值小于 0.5、0.6、07 和 0.8 的样本，构建了四个子数据集用于比较。对于 NYT50，通过分别选取 SE.Density 值大于 0.4、0.5、06 和 0.7 的样本，构建了四个子数据集用于比较。对于每个子数据集，训练集、开发集和测试集的划分遵循原始数据集。



	0.0-0.1	0.1-0.2	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0	>1.0
CNNDM	25	349	2034	5411	12871	18659	27657	33266	41780	46192	123247
NYT50	520	1926	5093	11274	21427	21332	17898	11602	6947	3490	2147

不同 SE.Density 值下 CNN/DM 和 NYT50 的样本分布

所提出的方法 RHGNNSumExt 和基线方法 HGNNSum 在子数据集上进行训练和评估。对于 HGNNSum，使用 [2] 中提供的默认设置，并采用表现最佳的评估模型进行测试。结果如表 4和表5所示。

不同 SE.Density 值下 CNN/DM 子数据集的评估结果

SE.Density	方法	Rouge-1	Rouge-2	Rouge-L
<0.5	RHGNNSumExt	40.28	18.34	36.72
	HGNNSum	40.36	18.59	36.84
<0.6	RHGNNSumExt	40.82	18.95	37.42
	HGNNSum	40.76	19.19	37.38
<0.7	RHGNNSumExt	41.55	19.75	38.22
	HGNNSum	41.44	19.61	38.12
<0.8	RHGNNSumExt	41.72	19.78	38.37
	HGNNSum	41.46	19.72	38.15

不同 SE.Density 值下 NYT50 子数据集的评估结果

SE.Density	方法	Rouge-1	Rouge-2	Rouge-L
>=0.4	RHGNNSumExt	45.22	25.19	41.37
	HGNNSum	46.03	25.82	41.99
>=0.5	RHGNNSumExt	45.62	25.57	41.81
	HGNNSum	46.37	26.04	42.29
>=0.6	RHGNNSumExt	46.72	26.33	42.81
	HGNNSum	47.23	26.72	43.10
>=0.7	RHGNNSumExt	47.56	27.16	43.71
	HGNNSum	47.53	27.15	43.69

如表4所示，对于 CNN/DM，在 SE.Density<0.5 和 SE.实体度<0.6 的子数据集上，所提出的方法 RHGNNSumExt 的表现不如 HGNNSum；而在 SE.Density<0.7 和 SE.实体度<0.8 的子数据集上，RHGNNSumExt 的表现优于 HGNNSum。对于 CNN/DM，随着子数据集的 SE.Density 增加，RHGNNSum 的性能有所提升。

如表5所示，对于 NYT50，在 SE.Density>=0.7 的子数据集上，RHGNNSumExt 的表现优于 HGNNSum；而在 SE.Density>=0.4、SE.实体度>=0.5 和 SE.实体度>=0.6 的子数据集上，RHGNNSumExt 的表现则不如 HGNNSum。需要注意的是，整个 NYT50 数据集的平均 SE.Density 为 0.57，且 RHGNNSumExt 在整个 NYT50 数据集上的表现不如 HGNNSum。对于 NYT50，随着子数据集的 SE.Density 增大，RHGNNSum 的表现也会有所提升。

4.5 消融实验

为更好地了解所提出的模型中不同模块的有效性，我们进行了消融研究。表6和表7分别展示了在 CNN/DM 和 NYT50 上的结果。

实体级实体嵌入的影响。如第3.1节所述，实体级实体嵌入和词级实体嵌入拼接为实体嵌入。实体级

实体嵌入通过两种方式部分表示来自知识库 YAGO2 的外部知识：1）使用 RDF2Vec 基于 YAGO2 初始化实体级实体嵌入；2）利用基于 YAGO2 构建的实体-实体边进行监督，通过 RHGNNSumExt 训练这些嵌入。

为考察添加实体级实体嵌入的影响，我们将嵌入从模型中移除，结果如表6和表7的第二行所示。RHGNNSumExt的得分高于未使用实体嵌入的模型，表明添加实体级实体嵌入能够提升所提出的摘要模型的性能。

表6 CNN/DM上的消融研究

方法	Rouge-1	Rouge-2	Rouge-L
RHGNNSumExt	42.39	19.45	38.85
不使用实体级实体嵌入	42.28	19.35	38.69
不使用实体嵌入监督	42.26	19.32	38.66
不使用边权重	42.33	19.39	38.76
不使用边类型	42.25	19.24	38.62
使用均值聚合	42.30	19.36	38.73
不使用实体-实体边和句子-句子边	42.22	19.32	38.67
RHGNNSumAbs	41.63	18.45	38.00
不使用强化学习连接器	41.55	18.36	37.91

表7 NYT50 上的消融研究

方法	Rouge-1	Rouge-2	Rouge-L
RHGNNSumExt	45.80	25.89	40.26
不使用实体级实体嵌入	45.64	25.66	40.12
不使用实体嵌入监督	45.61	25.55	40.02
不使用边权重	45.69	25.79	40.22
不使用边类型	45.62	25.65	40.11
使用均值聚合	45.76	25.68	40.11
不含实体-实体边和句子-句子边	45.54	25.57	39.99
RHGNNSumAbs	44.30	24.33	37.03
不含强化学习连接器	44.24	24.28	36.94

为了考察使用实体-实体边监督嵌入的效果，我们从模型中移除了这一子目标，结果见表6和表7的第三行。不使用实体-实体边监督的模型，其性能不如使用实体-实体边监督的模型。实体-实体边监督旨在保留实体嵌入中的实体-实体相关性，从而提升所提出的模型。

R-HGNN 中边权重和边类型的作用。句子-实体图中有三种权重不同的边。R-HGNN 在

传播过程中同时利用边权重和边类型来计算节点编码，结合了传统图神经网络和传统关系图神经网络的优势。

为考察边权重的影响，从图中移除边权重，并使用传统的 R-GNN 替代 R-HGNN。如表6和表7的第四行所示，不使用边权重的 R-GNN 表现不如使用边权重的 R-HGNN，这表明 R-HGNN 中的边权重能够提升性能。

为考察边类型的影响，从图中移除边类型，并使用传统图神经网络替代 R-HGNN。如表6和表7的第五行所示，不考虑边类型的图神经网络表现不如考虑边类型的 R-HGNN，这表明 R-HGNN 中的边类型能够提升性能。

聚合算法的影响。为考察聚合算法对所提出的模型的影响，在传播过程中忽略边权重，使用 HinSAGE [41] 的均值聚合算法替代 R-HGNN 的聚合算法。如表 6 和表 7 的第六行所示，使用 Agg.Mean 的 RHGNNSumExt 表现不如 RHGNNSumExt，这表明能够利用边权重的原始图神经网络的聚合算法比均值聚合算法更为适用。

实体-实体边和句子-句子边的影响。本研究构建的句子-实体图与 [2] 中构建的句子-词图的一个区别是：除句子-实体边外，句子-实体图还添加了实体-实体边和句子-句子边，而句子-词图仅包含句子-词边。表6和表7的第七行展示了所提出的模型在不使用实体-实体边和句子-句子边时的性能。不使用这两类边的模型表现不如使用这两类边的模型，甚至不如不使用实体嵌入的模型。实体-实体边和句子-句子边对于句子-实体图是必不可少的。

强化学习连接器的作用。表6和表7的最后两行分别展示了使用和不使用强化学习连接器的抽象式模型的 ROUGE 分数。RHGNNSumAbs 的表现优于不使用强化学习连接器的 RHGNNSumAbs。需要注意的是，不使用强化学习连接器的 RHGNNSumAbs 会直接将生成器应用于多任务选择器 RHGNNSumExt 选出的句子。选择器的性能会极大地影响生成器的表现。强化学习连接器通过微调选择器，使其更适配生成器，从而改善生成式摘要。

5 结论

本文提出了一种实体引导的文本摘要框架，通过连接知识图谱和图神经网络，利用文本之外的知识以及文本中的跨句关系，生成忠实的摘要。所提出的摘要框架的核心组成部分包括利用实体连接图神经网络与知识图谱的方法，以及用于摘要生成的关系异构图神经网络。通过构建句子-实体图，并基于知识图谱初始化和训练实体嵌入，实现图神经网络与知识图谱的连接。具体而言，知识图谱中的外部知识通过以下方式用于图神经网络：基于维基百科网页构建实体-实体边、初始化 YAGO2 中的实体节点编码，以及利用实体-实体边监督实体节点编码。关系异构图神经网络通过结合传统图神经网络与传统关系图神经网络的优势，利用传播过程中的边权重和边类型，计算带有加权多类型边的句子-实体图的节点编码。在 CNN/DM 上进行的实验表明，所提出的抽取式摘要方法优于所有已报告的不使用预训练语言模型的基线方法。在 NYT50 上进行的实验表明，所提出的方法优于大多数已报告的基线方法。针对 CNN/DM 和 NYT50 子数据集开展的实验表明，所构建的句子-实体图中句子-实体边的密度会显著影响所提出模型的性能；密度越高，模型性能越好。消融研究表明，所提出的方法是有效的，并且 R-HGNN 在利用边权重和边类型进行摘要生成方面优于传统图神经网络和传统关系图神经网络。研究结果为其他自然语言处理任务（如多模态摘要、问答和新闻图像描述）利用文档中的跨句关系及知识图谱中的外部知识，迈出了有希望的一步。

致谢 本研究由国家自然科学基金（编号：61806101）资助。

数据和材料的可用性 数据集和代码可在以下网址获取https://github.com/jingqiangchen/kbsumm。

参考文献

1. Zhuge H（2016）信息物理社会中的多维摘要。Morgan Kaufmann。

2. Wang D、Liu P、Zheng Y 等（2020）。用于抽取式文档摘要的异质图神经网络。载于：计算语言学协会第58届年会论文集，第6209–6219页。3. Sharma E、Huang L、Hu Z 等（2019）。生成式摘要的实体驱动框架。载于：2019年自然语言处理实证方法会议和第9届自然语言处理国际联合会议（EMNLP-IJCNLP）论文集，第3271–3282页。4. Cheng J、Lapata M（2016）。通过抽取句子和词语进行神经网络摘要。载于：计算语言学协会第54届年会论文集（第1卷：长篇论文），第484–494页。5. Nallapati R、Zhai F、Zhou B（2017）。Summarunner：一种基于循环神经网络的序列模型，用于文档抽取式摘要。载于：AAAI人工智能会议论文集，第3075–3081页。6. Liu Y、Lapata M（2019）。用于多文档摘要的层次化Transformer。载于：计算语言学协会第57届年会论文集，第5070–5081页。7. Li H、Zhu J、Zhang J 等。确保摘要的正确性：将蕴含知识融入生成式句子摘要。第27届国际计算语言学会议论文集。2018：1430–1441。8. Feng X、Feng X、Qin B。通过异构图网络将常识知识融入抽象式对话摘要。全国中文计算语言学大会。施普林格，Cham，2021：127–142。9. Li W、Xu J、He Y 等（2019）。利用图到序列模型为中文文章生成连贯评论。载于：计算语言学协会第57届年会论文集，第4843–4852页。10. Gunel B、Zhu C、Zeng M 等（2020）。注意事实：知识增强的连贯生成式文本摘要。arXiv预印本 arXiv:2006.15435。11. Zhang Z、Han X、Liu Z 等（2019）。ERNIE：利用信息实体增强语言表示。载于：计算语言学协会第57届年会论文集，第1441–1451页。

12. Liu W, Zhou P, Zhao Z, 等（2020）K-BERT：利用知识图谱实现语言表征。载于：AAAI人工智能会议论文集，第2901-2908页。13. Erkan G, Radev DR（2004）LexRank：基于图的词汇中心性作为文本摘要中的显著性。《人工智能研究杂志》22：457-479。14. Mihalcea R, Tarau P（2004）TextRank：为文本梳理脉络。载于：2004年自然语言处理实证方法会议论文集，第404-411页。15. Kipf TN, Welling M（2016）基于图卷积网络的半监督分类。arXiv预印本 arXiv:1609.02907。16. Zhao L, Xu W, Guo J. 利用图结构和主题词改进生成式对话摘要。国际计算语言学会议第28届论文集。2020：437-449。17. Yasunaga M, Zhang R, Meelu K, 等（2017）基于图的神经多文档摘要。载于：第21届计算自然语言学习会议（CoNLL）论文集，第452-462页。18. Xu J, Gan Z, Cheng Y, 等（2020）话语感知神经抽取式文本摘要。载于：计算语言学协会第58届年会论文集，第5021–5031页。19. Schlichtkrull M, Kipf TN, Bloem P, 等（2018）利用图卷积网络对关系数据建模。载于：欧洲语义网会议，施普林格，Cham，第593-607页。20. Hoffart J, Yosef MA, Bordino I, 等（2011）文本中命名实体的鲁棒消歧。载于：2011年自然语言处理实证方法会议论文集，第782-792页。21. Nguyen DB, Hoffart J, Theobald M, 等（2014）AIDA-light：高吞吐量命名实体消歧。2014年网络关联数据研讨会，第1-10页。22. Cao Y, Hou L, Li J, 等（2018）神经集体实体链接。载于：第27届国际计算语言学会议论文集，第675-686页。23. Ristoski P, Rosati J, Di Noia T, 等（2019）RDF2Vec：RDF图嵌入及其应用。《语义网》10(4)：721-752。

24. Nenkova A（2008）面向多文档摘要的实体驱动改写。载于：第三届自然语言处理国际联合会议论文集，第118-125页。25. Inan E（2021）Somun：融合预训练语言模型的实体中心摘要。《神经计算与应用》33：5301–5311。 https://doi.org/10.1007/s00521-020-05319-226. Parveen D, Ramsl HM, Strube M（2015）基于图的抽取式摘要的主题连贯性。载于：2015年自然语言处理实证方法会议论文集，第1949-1954页。~~27. Huang L, Wu L, Wang L. 利用语义~~驱动的完形填空奖励增强知识图谱生成式摘要。计算语言学协会第58届年会论文集。2020：5094-5107。28. Zhuge H（2020）人机自然共生中的信息物理社会智能。施普林格。29. Bordes A, Usunier N, Garcia-Durán A, 等（2013）用于多关系数据建模的平移嵌入。载于：第26届神经信息处理国际会议论文集，第2787-2795页。30. Vaswani A, Shazeer N, Parmar N, 等（2017）注意力就是一切。载于：第31届神经信息处理系统国际会议论文集，第6000-6010页。31. Devlin J, Chang MW, Lee K, 等（2019）BERT：用于语言理解的深度双向变换器预训练。载于：2019年北美计算语言学协会分会人类语言技术会议论文集：第1卷（长篇与短篇论文），第4171-4186页。32. Veličković P, Cucurull G, Casanova A, 等（2017）图注意力网络，arXiv预印本 arXiv:1710.10903。33. Gilmer J, Schoenholz SS, Riley PF, 等（2017）用于量子化学的神经消息传递。载于：国际机器学习会议，PMLR，第1263-1272页。34. Fernandes P, Allamanis M, Brockschmidt M（2018）结构化神经摘要。arXiv预印本 arXiv:1811.01824。

35. Zhang C, Song D, Huang C, 等 (2019) 异质图神经网络。载于：第25届ACM SIGKDD知识发现与数据挖掘国际会议论文集，第793-803页。36. Shi J, Ji H, Shi C, 等 (2020) 用于推荐的异质图神经网络。arXiv预印本 arXiv:2009.00799。37. Wang S, Chen Z, Li D, 等 (2019) 注意力异质图神经网络：在程序再识别中的应用。载于：2019年SIAM数据挖掘国际会议论文集。工业与应用数学学会，第693-701页。38. Hamilton WL, Ying R, Leskovec J (2017) 大型图上的归纳式表示学习。载于：第31届神经信息处理系统国际会议论文集，第1025-1035页。39. Wang, X, Ji H, Shi C, Wang B, Ye Y, Cui P, Yu PS (2019) 异质图注意力网络。载于：万维网大会，第2022-2032页。40. Kang S (2021) 利用图神经网络进行缺失数据下的产品故障预测。《神经计算与应用》, 33:7225–7234. https://doi.org/10.1007/s00521-020-05486-241. CSIRO的Data61 (2018) Stellargraph机器学习库。https://github.com/stellargraph/stellargraph42. Xiao W, Carenini G (2019) 结合全局和局部上下文的长文档抽取式摘要。载于：2019年自然语言处理实证方法会议和第9届自然语言处理国际联合会议（EMNLP-IJCNLP）论文集，第3002-3012页。43. Zhong M, Liu P, Wang D, 等 (2019) 探索有效的神经抽取式摘要：哪些方法有效，接下来是什么。载于：计算语言学协会第57届年会论文集，第1049-1058页。44. Li S, Xu J (2021) 一种采用异步且信息增强解码的两步式抽象摘要模型。《神经计算与应用》, 33:1159–1170。https://doi.org/10.1007/s00521-020-05005-345. Chen YC, Bansal M (2018) 通过强化学习选择的句子重写实现快速抽象式摘要。载于：计算语言学协会第56届年会论文集（第1卷：长篇论文），第675-686页。

46. Li C, Xu W, Li S, 等 (2018) 基于关键信息引导网络的抽象式文本摘要生成引导。载于：2018年北美计算语言学协会分会：人类语言技术会议论文集，第2卷（短篇论文），第55-60页。47. Gehrmann S, Deng Y, Rush AM (2018) 自底向上抽象式摘要。载于：2018年自然语言处理中的经验方法会议论文集，第4098-4109页。48. Bahdanau D, Cho K, Bengio Y (2014) 通过联合学习对齐与翻译实现神经机器翻译。arXiv预印本 arXiv:1409.0473。49. See A, Liu PJ, Manning CD (2017) 直奔主题：使用指针生成网络进行摘要。载于：计算语言学协会第55届年会论文集（第1卷：长篇论文），第1073-1083页。50. Hoffart J, Suchanek FM, Berberich K, 等 (2013) YAGO2：来自维基百科、在空间和时间维度上增强的知识库。《人工智能》194:28-61。https://doi.org/10.1016/j.artint.2012.06.00151. Manning CD, Surdeanu M, Bauer J, 等 (2014) Stanford CoreNLP自然语言处理工具包。载于：计算语言学协会第52届年会系统演示论文集，第55-60页。52. Rennie SJ, Marcheret E, Mroueh Y, 等 (2017) 用于图像描述生成的自批判序列训练。载于：IEEE计算机视觉与模式识别会议论文集，第7008-7024页。53. Lin CY (2004) ROUGE：用于摘要自动评估的工具包。《文本摘要的拓展》，第74-81页。54. Hermann KM, Kočiský T, Grefenstette E, 等 (2015) 教机器阅读和理解。载于：第28届神经信息处理系统国际会议论文集，第1卷，第1693-1701页。55. Durrett G, Berg-Kirkpatrick T, Klein D (2016) 基于学习的单文档摘要：压缩与照应性约束。载于：计算语言学协会第54届年会论文集（第1卷：长篇论文），第1998-2008页。56. Sandhaus E (2008) 《纽约时报》标注语料库，语言数据联盟。费城。

57. Mikolov T、Sutskever I、Chen K 等（2013）词语和短语的分布式表示及其组合性。arXiv 预印本 arXiv:1310.4546。58. Kingma DP、Ba J（2014）Adam：一种随机优化方法。arXiv 预印本 arXiv:1412.6980。59. Liu Y、Lapata M（2019）使用预训练编码器进行文本摘要。载于：2019年自然语言处理实证方法会议和第9届自然语言处理国际联合会议（EMNLP-IJCNLP）论文集，第3721–3731页。60. Xu J、Durrett G（2019）结合句法压缩的神经抽取式文本摘要。载于：2019年自然语言处理实证方法会议和第9届自然语言处理国际联合会议（EMNLP-IJCNLP）论文集，第3283–3294页。61. Luo L、Ao X、Song Y 等（2019）像 HER 一样阅读：受人类阅读启发的抽取式摘要。载于：2019年自然语言处理实证方法会议和第9届自然语言处理国际联合会议论文集，第3024–3034页。62. Paulus R、Xiong C、Socher R（2017）用于生成式摘要的深度强化模型。arXiv 预印本 arXiv:1705.04304。63. Liu Y、Titov I、Lapata M（2019）将单文档摘要视为树诱导。载于：2019年北美计算语言学协会分会人类语言技术会议论文集，第1卷，第1745–1755页。