文章编号：1003-0077（2017）00-0000-00

基于篇章结构图网络的话题分割

徐邵洋 [1] 蒋峰 [1] 李培峰 [1]
(1.苏州大学 计算机科学与技术学院，江苏 苏州 215006)

摘要：话题分割是自然语言处理领域的经典任务之一，它的目标是将输入的篇章分割成语义连续的段落。先前的工作使用基于词频、隐式、序列以及 transformer 的方法来编码句子，忽略了对篇章整体语义信息的建模。因此，该文采用篇章结构图网络编码句子，获得具有篇章全局信息的句子表示用于话题分割。具体的，该模型首先为每一个篇章单独建图，其中包含篇章的所有句子和单词节点以及它们之间的邻接信息。然后，模型使用 GGNN 对图进行迭代，得到包含篇章全局信息的句子表示。它们被进一步送入 Bi-LSTM 层以预测话题分割点。实验结果表明，与其他基准系统相比，该模型能够获得更适合话题分割任务的句子表示，在多个流行的数据集上取得了最好的性能。

关键词：话题分割；句子编码；图神经网络

中图分类号：TP391 **文献标识码：**A

Topic Segmentation via Discourse Structure Graph Network

XU Shaoyang[1], JIANG Feng[1], LI Peifeng[1]
(1.School of Computer Science and Technology, Soochow University, Suzhou, Jiangsu 215006, China)

Abstract ： Topic segmentation is one of the classic tasks in the field of natural language processing. Its goal is to segment the input discourse into paragraphs with continuous semantics. Previous works used word frequency-based, latent-based, sequential-based, and transformer-based methods to encoder sentences, ignoring modeling global semantic information of the discourse. Therefore, this paper used Discourse Structure Graph Network encoding sentences to obtain the sentence representation with the global information of the discourse for topic segmentation. In detail, the model firstly constructs a separate graph for each discourse, which contains all sentence and word nodes of it as well as the adjacency information between them. The model then uses GGNN to iterate the graph that gets the sentence representation with the global information of the discourse. They are further fed to the Bi-LSTM layer to predict the segmentation points. The experimental results demonstrate that the model gets a more suitable sentence representation than other baselines for topic segmentation and achieves the best performance on various popular datasets.

Key words： topic segmentation; sentence encoding; graph neural networks

0 引言

话题分割是自然语言处理领域的经典任务之一，其目标是将输入的篇章分割成语义连续的段落。经过分割的篇章可以方便读者浏览感兴趣的内容[1]，话题分割在文本摘要、信息抽取等下游任务中应用广泛[2]。如图 1 所示，考虑一个包含

收稿收稿日期：2017-03-16；**定稿日期**：2017-04-26

基金项目：国家自然基金面上项目 面向话题的事件关系抽取与网络构建研究（61772354）；国家自然基金重点项目 面向领域大数据的事件知识图谱构建研究(61836007)

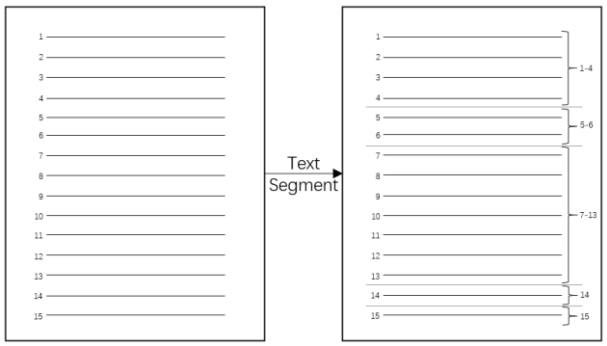

图 1 话题分割示意图

15 个句子的篇章，话题分割任务需要预测出篇章中潜在的分割点，把篇章分成 1-4，5-6，7-13，14，15 这 5 个语意连贯的段落。

现有的话题分割方法主要分为无监督的和有监督的这两种。话题的变化带来词汇的变化，这一变化在话题间的边界上表现最为明显，无监督的方法基于词汇的变化特征。基于这一特征，分类模型寻找合适的相似度度量，去衡量相邻片段间的差异性，确定差异性大的位置为分割点[3,4]；聚类算法则把相似性大的相邻片段聚为一段[5,6]；生成式的方法寻找最有可能生成被观测篇章的话题序列，进而确定分割点[7]。随着大型标注数据的出现，预测篇章内的句子是否是分割点的有监督模型被提出[2,8-13]。它们为每个句子预测一个二值标签，表明其是否是分割点。在此基础上，Barrow 等人[8]进一步为不同的段落预测不同的话题标签，Glavaš 等人[9]进一步为不同的段落进行连续性打分。总的来说，话题分割方法从识别词汇变化特征的无监督方法过渡到了判断句子是否是分割点的有监督方法。

本文将话题分割建模为判断每一个句子是否是分割点的有监督任务。在此情境下，本文认为，语义信息建模和计算复杂度是该任务的两个关键点。一方面，话题分割任务以篇章为单位，其颗粒度较粗，既要考虑句子内部的信息，又要考虑篇章内部的信息交互，因此其对语义信息的建模要求较高；另一方面，与传统句子分类任务不同，话题分割任务中的一个篇章单位包含不同数量的句子，所以该任务对计算的复杂度也有较高要求。然而，现有的方法主要存在以下两个缺陷：语义信息建模能力不足以及计算复杂度较高。早期的研究使用基于词频和滑动窗口的方法[3]，然而这种方法得到的特征向量相对稀疏，且忽略了单词之间的位置信息。基于奇异值分解的隐式方法[4]在一定程度上解决了特征向量的稀疏性问题，但它依然没有考虑单词之间的位置信息。序列方法[2,8]通过 Bi-LSTM+pooling 得到稠密的特征向量，同时考虑了单词之间的序列关系，序列方法还可以被用于上层网络，进一步捕捉篇章内的序列关系。但是，一方面，句子内部存在自然的语法结构，因此单词之间的关系并不一定是序列的[14]；另一方面，篇章内部的上下文关系也不一定是序列的。最近的研究中使用的树形方法[15]和基于 transformer[16]的方法[9,12]能够同时考虑单词之间以及篇章内部的非序列关系。遗憾的是，这两种方法都具有较高的计算复杂度，特别是在以篇章为单位的话题分割任务中，使用它们需要很大的计算资源，而且还需要做出对长句子进行截断、把长篇章截断成多份的牺牲。总的来说，先前的工作没有找到一种能够同时考虑单词之间以及篇章内部的非序列关系，从而得到篇章的全局语义信息，并具有较低计算复杂度的话题分割方法。

本文提出基于篇章结构图网络的话题分割模型（DSG-SEG），它能够同时解决全局语义信息建模和计算复杂度的问题。具体的，受到 Yao 等人[17]和 Zhang 等人[18]工作的启发，DSG-SEG 模型首先取出一个篇章内的所有单词和句子节点，把每一个篇章单独构建成图。在初始化图的特征矩阵和邻接矩阵之后，该模型把建好的图送入 GGNN 网络[19]来编码句子。最后，模型把得到的句子表示序列进一步送入 Bi-LSTM 网络，进行分割点的预测。

为了得到篇章的全局语义信息，本文的模型利用了单词节点、单词与句子节点之间的邻接信息，而随着 GGNN 网络的迭代，句子节点之间也产生了间接的信息交互，模型最终得到的句子表示能够同时考虑单词之间以及篇章内部的非序列关系。另外，实验结果显示本文的模型具有很低的计算复杂度，不需要做出任何对长句子进行截断、把长篇章截断成多份的牺牲。相较于其它句子编码方式，本文的模型能够同时取得更好的结果指标和时间性能。本文的主要贡献总结如下：

(1)本文提出了一个基于篇章结构图网络的话题分割模型（DSG-SEG）。据我们所知，这是

图神经网络首次被用于话题分割任务。

(2)本文提出的模型能够同时考虑单词之间以及篇章内部的非序列关系，具有全局的语义信息建模能力。

(3)本文提出的模型能够在多个数据集上同时取得最好的结果指标和时间性能。

1 相关工作

此部分归纳了话题分割任务中已有的，以及在其它工作中较为典型的几种句子编码方法。一般的，之前的研究主要使用基于词频[3]、隐式[4]、序列[2,8]和 transformer 的方法[9,12]来编码句子。除此以外，Shi 等人[15]比较了多种树形句子编码方法，用于单句子分类、句子关系分类和句子生成等下游任务；Yao 等人[17]和 Zhang 等人[18]提出了基于图神经网络的句子编码方法，用于文本分类任务。下面将详细介绍各个方法。

1.1 词频&隐式方法

TextTiling[3]使用词频方法表示窗口(句子)。方法定义了一个大小为 k 的窗口，表示每个窗口包含 k 个句子。假设词表大小为 m，每个窗口则被表示成一个 m 维的词频向量 $\boldsymbol{x}$。对于第 i 个可能的分割点，方法对该间隔点左右两个窗口的词频向量 $\boldsymbol{x}_l^{(i)}$ 和 $\boldsymbol{x}_r^{(i)}$ 计算余弦距离，得到该间隔点的相似度分数 $score_i$。方法使用曲线平滑等方式，把所有相似度分数转化成一条深度曲线，最后通过与预先设定的深度阈值作比较，来得到所有的分割点。

LSA[4]把篇章中的句子表示为 m 维的词频向量，一个长度为 n 的篇章被表示为矩阵 $\boldsymbol{X}\epsilon\mathbb{R}^{n\times m}$。方法利用奇异值分解，提取 k 个最大的奇异值来近似描述原来的矩阵 $\boldsymbol{X}$，所以，原来 m 维的词频向量被压缩成了 k 维的隐式向量。方法进一步的分割过程与 TextTiling 类似。

TextTiling 使用词频方法来得到窗口表示，利用词汇的变化特征来寻找分割点。但是，它获得的特征向量相对稀疏，且忽略了单词之间的真实依赖。LSA 通过奇异值分解得到了较为稠密的特征向量，但它依然没有解决单词之间的真实依赖。

1.2 序列方法

TextSeg[2]首先把句子表示成单词序列，并使用 300 维的 word2vec 词嵌入模型对单词进行初始化。然后把单词序列输入到 Bi-LSTM 中，取出每一个时间步的隐藏层向量，使用 max-pooling 得到句子向量表示。该句子表示序列被送入上层 Bi-LSTM 以捕捉它们之间的序列关系，方法进一步使用全连接层来给每一个句子预测一个二值标签，表示其是否是分割点。

序列方法使用 Bi-LSTM+pooling 的方式得到稠密的特征向量，能够考虑单词之间的序列关系，上层的 Bi-LSTM 能够进一步捕捉篇章内部的序列关系。然而，单词之间不仅仅是简单的序列关系，句子内部通常有自然的语法结构[14]，而且，篇章内部的上下文间也存在复杂的关系。

1.3 基于 transformer 的&树形方法

Hier.BERT[12]使用了 768 维的 $\text{BERT}_{\text{Base}}$，把[CLS]的向量表示作为句子的表示。然后，该句子表示序列被进一步送入上层 transformer 模型，模型通过 cross-attention 把不同的句子联系起来。方法最终使用全连接层来给每一个句子预测一个二值标签，表示其是否是分割点。

Shi 等人[15]首先把句子表示成带有树形结构的单词序列，并使用 300 维的 GloVe 词嵌入模型[20]对单词进行初始化。然后把单词序列输入到 Tree-LSTM 模型，不同的树形结构意味着不同的计算过程。作者尝试了直接使用最后一个隐藏节点的表示以及使用 pooling 结合所有隐藏节点这两种方式，来得到句子向量表示，句子向量表示序列被进一步送入上层模型进行下游任务。

基于 transformer 的方法和树形方法都能同时考虑单词之间以及篇章内部的非序列关系，但是这两种方法的计算复杂度较高，在话题分割任务中使用它们时需要很大的计算资源，而且需要做出截断长句子、将长篇章截断成多份的牺牲。

1.4 基于图神经网络的方法

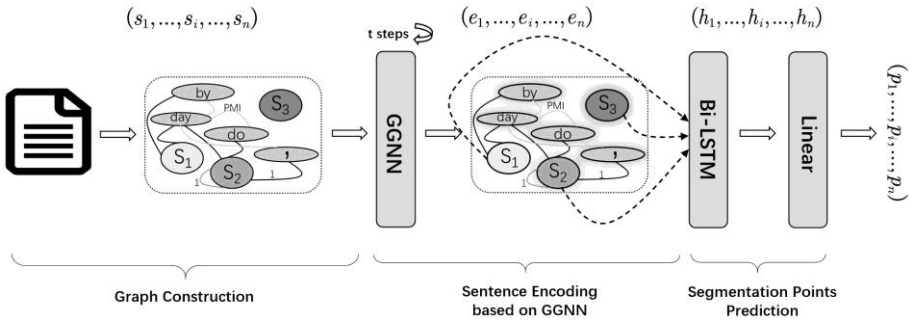

图 2 DSG-SEG 模型框架

TextGCN[17]和 TextING[18]都是图神经网络在文本分类任务上的应用，这两个模型在图的构建部分和图网络的选择部分有所不同。

TextGCN 将整个语料构建成图，图中同时包含单词和文章(句子)节点，节点都被初始化为 one-hot 向量。方法使用共现信息(PMI 指标)来衡量单词节点之间的边的权重，使用 TF-IDF 指标来计算单词和文章节点之间的边的权重。在得到图的特征矩阵 $\boldsymbol{X}$ 以及邻接矩阵 $\boldsymbol{A}$ 之后，方法把建好的图送入 GCN 网络[21]进行迭代，然后取出图中的 document embedding 作为文章的向量表示，并将其送入全连接层用于文本分类。

TextING 给每个文章(句子)单独建图，图中仅包含了该文章中的所有单词节点，节点使用 300 维的 GloVe 词嵌入模型进行初始化。单词节点之间存在边，方法赋予每一个在滑动窗口内共同出现过的单词对一条权重为 1 的连边。在得到图的特征矩阵 $\boldsymbol{X}$ 以及邻接矩阵 $\boldsymbol{A}$ 之后，方法把建好的图送入 GGNN 网络[19]进行迭代，然后通过 readout function 融合所有单词节点来得到文章的向量表示，并将其送入全连接层用于文本分类。

TextGCN 的优点在于把句子节点加入到了图中，当多层 GCN 网络被堆叠、更远的邻接信息被整合时，句子节点之间也产生了间接的信息交互，也就是说，方法能够得到整个语料的全局信息。但是，该方法将整个语料构建成一张图，需要很大的内存[22]。TextING 的优点在于为每个文章单独建图，减小了内存的消耗。但是该方法没有把句子节点加入图中，也就无法得到语料的全局信息。

2 模型

为了得到篇章的全局语义信息，并解决现有句子编码方法在话题分割任务中的计算复杂度问题，受到 TextSeg[2]、TextGCN[17]和 TextING[18]等工作的启发，本文提出了一个基于篇章结构图网络的话题分割模型（DSG-SEG），模型的总体架构如图 2 所示。接下来的三个部分将依次介绍该模型的架构：图的构建、句子编码、分割点预测。

2.1 图的构建

模型的输入是一个篇章，视其为句子序列 $(s_1,...,s_i,...,s_n)$，模型将该篇章构建成一个单独的图，该图由一个特征矩阵和一个邻接矩阵组成。接下来，本文将介绍图的构建过程，主要包括三个部分：节点的表示、邻接边的连接、图的整合。

节点的表示 TextGCN 为整个语料构建了一个以语料中所有单词和句子为节点的图，与之不同的是，DSG-SEG 把每个篇章构建成一个单独的图，该图以篇章中所有单词和句子为图中的节点，也就是说，每一个图存在两种节点：(1)单词节点和(2)句子节点。

(1)单词节点 模型将单词节点初始化为预训练的词嵌入向量，因为词嵌入方法不仅缓解了冷启动问题，也带来了更准确的语义信息[23]。

(2)句子节点 模型用 max-pooling 的方式来得到句子节点的初始化向量。特别的，TextING 为每个句子构建单独的图，图中只包含单词节点而不包含句子节点，为了与之比较，本文会在消融实验部分探究图中是否包含句子节点对于模型表现的影响。

由此，模型得到了这个图的特征矩阵 $\boldsymbol{X}\epsilon\mathbb{R}^{n^+\times m}$，其中，$n^+$ 表示单词和句子节点的总数，m 表示节点的特征维度。特别的，$\boldsymbol{X}$ 的前 n 行存放 n 个句子节点，第 n+1 行开始存放单词节点：

$$\boldsymbol{X}_i=\begin{cases}\max\left(w_1^{(i)},...,w_{l_i}^{(i)}\right) & 1\leqslant i\leqslant n\\ word_{i-n} & n+1\leqslant i\leqslant n^+\end{cases} \quad (1)$$

其中，$\left(w_1^{(i)},...,w_{l_i}^{(i)}\right)$表示第 i 个句子的单词序列，l_i表示其长度。

邻接边的连接 一个图存在三种边：(1)单词节点之间的边、(2)单词和句子节点之间的边以及(3)自环边。

(1)单词节点之间的边 参照 TextGCN 的工作，模型使用单词的共现信息来衡量单词之间的边的权重。具体来说，模型使用一个固定大小的滑动窗口，在一个篇章的所有句子上滑动，来计算单词之间的 PMI 指标。对于给定的单词对<*i*, *j*>，PMI 指标的计算公式如下：

$$PMI(i,j)=\log\frac{p(i,j)}{p(i)p(j)} \quad (2)$$

$$p(i,j)=\frac{\#W(i,j)}{\#W} \quad (3)$$

$$p(i)=\frac{\#W(i)}{\#W} \quad (4)$$

其中，*#W* 表示所有的滑动窗口数，*#W(i)*表示出现过单词 *i* 的窗口数，*#W(i, j)*表示同时出现过单词 *i*、单词 *j* 的窗口数，*p(i)*、*p(j)*分别表示所有滑动窗口中，单词 *i* 和单词 *j* 各自出现的频率，*p(i, j)*表示所有滑动窗口中，单词 *i* 和单词 *j* 共同出现的频率。

由于负的 PMI 指标代表了很低的语义关联度，模型只保留 PMI 指标为正的单词节点之间的边。

特别的，TextING 简单地将所有邻接单词节点之间的连边权重设置为 1，为了与之比较，本文会在消融实验部分探究是否使用 PMI 指标对于模型表现的影响。

(2)单词和句子节点之间的边 对于一个句子节点，它与它所包含的每一个单词节点之间都存在一条权重为 1 的连边。模型没有使用 TF-IDF 指标作为权重，因为实验结果表明，使用 TF-IDF 指标并没有为模型带来性能上的提升。

(3)自环边 和 TextGCN 一样，模型给图中所有的节点都设置一条权重为 1 的自环边，从而使得每个节点不仅能够关注邻接节点的信息，也能够保留自身已经学到的信息[23]。

由此，模型得到了这个图的邻接矩阵 $\boldsymbol{A}\epsilon\mathbb{R}^{n^+\times n^+}$：

$$\boldsymbol{A}_{ij}=\begin{cases}PMI(i,j) & i,j\ are\ words, PMI(i,j)>0\\ 1 & i\ is\ sentence, j\ is\ word\\ 1 & i=j\\ 0 & other\end{cases} \quad (5)$$

图的整合 如上所述，对于一个篇章，模型首先得到了图的特征矩阵 $\boldsymbol{X}\epsilon\mathbb{R}^{n^+\times m}$ 和图的邻接矩阵 $\boldsymbol{A}\epsilon\mathbb{R}^{n^+\times n^+}$。接着，模型计算图的度矩阵 $\boldsymbol{D}$，度矩阵的计算公式如下：

$$\boldsymbol{D}_{ii}=\sum_{j=1}^{n^+}\boldsymbol{A}_{ij} \quad (6)$$

模型进一步使用度矩阵 $\boldsymbol{D}$ 和邻接矩阵 $\boldsymbol{A}$ 计算得到归一化的对称邻接矩阵 $\widetilde{\boldsymbol{A}}$：

$$\widetilde{\boldsymbol{A}}=\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}} \quad (7)$$

2.2 句子编码

经过图的构建过程，模型把输入的篇章构建成图，得到了图的特征矩阵 $\boldsymbol{X}$ 和归一化的对称邻接矩阵 $\widetilde{\boldsymbol{A}}$。模型把建好的图送入 GGNN 网络用以学习句子的向量表示。需要特别说明的是，在网络迭代一轮之后，句子节点聚合了与它邻接的单词节点的信息，单词节点则聚合了与它语义关联度高的其它邻接单词节点的信息。网络迭代两次及以上之后，原本没有直接边相连的句子节点，通过共有的邻接单词节点，也间接产生了信息交换。网络的迭代公式如下：

$$a^t=\widetilde{A}h^{t-1}W_a \quad (8)$$

$$z^t=\sigma(W_z a^t+U_z h^{t-1}+b_z) \quad (9)$$

$$r^t=\sigma(W_r a^t+U_r h^{t-1}+b_r) \quad (10)$$

$$\tilde{h}^t=\tanh\left(W_h a^t+U_h(r^t\odot h^{t-1})+b_h\right) \quad (11)$$

$$h^t=\tilde{h}^t\odot z^t+h^{t-1}\odot(1-z^t) \quad (12)$$

其中 $h^0=\boldsymbol{X}$，σ 是 sigmoid 函数，所有的 *W*、*U* 和 *b* 都是可训练的参数，*z* 和 *r* 分别代表更新门和重置门，以决定邻接信息对于当前节点的贡献程度[18]。

如上所述，模型在特征矩阵 $\boldsymbol{X}$ 的前 *n* 行保存了该篇章中所有的句子节点。经过 *t* 次网络的迭代，模型取出 h^t 的前 *n* 行作为句子向量表示序列

表 1 数据集部分信息

DATASET	WIKI-727K	WIKI-SECTION	CHOI	CLINICAL	ELEMENTS	CITES	WIKI-50	MANIFESTO
documents	727,746	21,376	920	227	118	100	50	5
sent/seg	13.6±20.3	7.2	7.4±2.96	28.0	3.33±3.05	5.15±4.57	13.6	8.99±10.8
seg/doc	3.48±2.23	7.9	9.98±0.12	5.0	6.82±2.57	12.2±2.79	3.5	127±42.9
real world	√	√	×	√	√	√	√	√

$(e_1,\ldots,e_i,\ldots,e_n)$：

$$e_i = h_i^t \quad 1 \leqslant i \leqslant n \tag{13}$$

2.3 分割点预测

经过图的构建和句子编码过程，模型得到了输入篇章的句子表示序列$(e_1,\ldots,e_i,\ldots,e_n)$。该句子表示序列被送入双层的 Bi-LSTM 模型，从而得到句子的隐藏层向量序列$(h_1,\ldots,h_i,\ldots,h_n)$：

$$(h_1,\ldots,h_i,\ldots,h_n) = \mathrm{BiLSTM}(e_1,\ldots,e_i,\ldots,e_n) \tag{14}$$

最后，模型使用一层全连接层和 softmax 层来将每一个句子的隐藏层向量映射到一个 0 到 1 的概率分布p_i。与 TextSeg 一样，对于一个包含 n 个句子的篇章，模型最小化前 n-1 个句子的交叉熵之和来训练参数，损失函数公式如下：

$$L(\theta) = \sum_{i=1}^{n-1}[-y_i \log p_i - (1-y_i)\log(1-p_i)] \tag{15}$$

其中，θ表示模型的可训练参数，y_i表示第 i 个句子的真实标签。

3 实验

为了充分验证本文提出的模型在话题分割任务中的有效性，实验在一个较大的数据集上训练模型，而在其它 8 个数据集上测试模型。这些数据集的部分信息如表 1 所示。

3.1 数据集

实验选择 WIKI-727K(Koshorek 等人，2018)作为模型的训练语料，由于硬件资源的限制，实验从中抽取了 1 万个篇章(WIKI-10K)，并按照 8/1/1 的比例划分数据集。也就是说，实验把 8000 个从 WIKI-727K 中抽取的篇章作为训练集，将训练好的模型在以下 8 个测试集上进行评估：

WIKI-10K 是从 WIKI-727K 中抽取的 1 万个篇章。WIKI-727K 包含了超过 72 万个来自英文维基百科的篇章，按照 8/1/1 的比例被划分成训练集、验证集和测试集。实验按照同样的比例对 WIKI-10K 进行数据集的划分。在数据预处理部分，和作者一样，列表、代码片段和其它特殊元素被过滤掉了。实验选取 WIKI-10K 中的测试集作为 8 个测试集的 1 个。

WIKI-SECTION(Arnold 等人，2019)包含了英语和德语两种语料，实验使用其中的英语语料 WIKI-SECTION(EN)。它包含了 21376 个来自维基百科的篇章，和作者一样，实验按照 7/1/2 的比例划分训练集(14472)、验证集(2073)和测试集(4142)，并选取其中的测试集作为 8 个测试集的 1 个。

CHOI(Choi，2000)包含了 920 个从布朗语料中人工拼接而成的篇章，因此，该数据集并不能真实反映存在的话题转移，但为了方便比较，依然考虑使用该数据集。

CLINICAL BOOKS(Malioutov 和 Barzilay，2006)是作者从一本医学教科书中抽取的 227 个篇章。

ELEMENTS(Chen 等人，2009)是作者从维基百科中抽取的 118 个篇章。

CITIES(Chen 等人，2009)是作者从维基百科中抽取的 100 个篇章。

WIKI-50(Koshorek 等人，2018)是作者从 WIKI-727K 中随机抽取的 50 个篇章。

MANIFESTO(Glavaš 等人，2016)是作者手

表 2 所有模型的实验结果(P_k%)

MODEL	DATASET								
	WIKI-10K	WIKI-SECTION	CHOI	CLINICAL	ELEMENTS	CITES	WIKI-50	MANIFESTO	MEAN
random	49.04	47.87	49.54	47.66	51.04	48.96	49.40	**51.32**	49.35
latent-sr	32.37	36.16	47.33	32.37	51.34	41.97	29.64	54.13	40.66
freq-sr	31.38	34.15	47.22	32.69	51.04	40.39	29.10	54.15	40.02
BERT+Bi-LSTM[12]	33.05	39.00	47.73	**32.41**	51.85	41.67	30.00	54.17	41.24
tree-sr-left	31.21	32.67	47.06	35.07	51.11	40.18	27.39	54.06	39.84
TextSeg[2]	28.17	31.12	45.10	34.68	48.92	37.43	23.01	54.16	37.82
BERT+Bi-LSTM$_{nft}$	27.01	29.40	43.50	35.04	**47.28**	38.54	22.71	54.13	37.20
DSG-SEG	**26.09**	**28.29**	**38.64**	33.80	47.49	**32.57**	**22.27**	54.10	**35.41**

工分割的 5 篇政治宣言。

实验直接使用以上 6 个数据集的全部，作为剩下的 6 个测试集。

3.2 评估指标

实验使用 P_k 指标(Beeferman 等人，1999)来评估模型的性能。P_k 指标的计算公式如下：

$$P_k(ref,hyp)=\sum_{i=0}^{n-k}\delta_{ref}(i,i+k)\neq\delta_{hyp}(i,i+k) \quad (16)$$

其中，δ 是一个标识符，当第 i 个位置和第 $i+k$ 个位置的句子属于同一分割时，$\delta=1$，否则 $\delta=0$。k 代表窗口的大小，一般取 *ref* 长度的一半。P_k 指标衡量了真实情况和预测情况的差异程度，所以，越小的 P_k 值表示越好的性能表现。

3.3 基准系统

为了证明图神经网络作为句子编码的一种方法，在话题分割任务中的有效性，实验将 DSG-SEG 与其它五种句子编码方法(词频、隐式、序列、树形、基于 transformer 的方法)作比较。需要特别说明的是，实验使用五种句子编码方法替换 DSG-SEG 模型的 Graph Construction 和 Sentence Encoding 部分，在得到各自的句子向量表示之后，所有基准模型都进一步使用与 DSG-SEG 的 Segmentation Points Prediction 部分一致的网络，来进行分割点的预测。下面将详细介绍各个方法及其参数设置。

词频方法(freq-sr)取 10000 个高频词构成词典，10000 个词典内的单词和 1 个词典外的词(<*UNK*>)被表示成 10002 维 one-hot 向量(含<*PAD*>)，每一个句子被表示成句内所有单词向量的加和，即一个 10002 维的句子向量。

隐式方法(latent-sr)首先把一个含有 n 个句子的篇章表示成 $n\times m$ 维的词频矩阵，然后利用奇异值分解，提取 k=300 个最大的奇异值来近似描述原来的词频矩阵。对于每一个句子而言，原来 m 维的词频向量被压缩成了 300 维的句子向量。

序列方法(TextSeg[2])首先把每一个句子表示为单词序列，并使用预训练词向量对单词进行初始化。然后将单词序列送入隐藏层维度为 256 维的双层 Bi-LSTM，最后使用 max-pooling 得到 512 维的句子向量。

树形方法(tree-sr)首先把每一个句子表示为单词序列，并使用预训练词向量对单词进行初始化。然后将单词序列送入 Tree-LSTM，最后使用 attn-pooling 得到 300 维的句子向量表示。实验考虑 Binary-balanced tree、Left-branching tree 和 Right-branching tree 这三种平凡的树形结构，由

表 3 消融实验结果(P_k%)

MODEL	DATASET								
	WIKI-10K	WIKI-SECTION	CHOI	CLINICAL	ELEMENTS	CITES	WIKI-50	MANIFESTO	MEAN
-SENTNODE	+0.90	+1.69	+4.65	-0.75	+2.15	+5.72	+0.82	+0.05	+1.90
- PMI	-0.04	+0.56	+2.47	-1.44	+0.58	+1.94	+0.68	+0.04	+0.60
DSG-SEG	26.09	28.29	38.64	33.80	47.49	32.57	22.27	54.10	35.41

于 Left-branching tree 的实验结果最好，所以仅使用它作为基准模型之一，记作 tree-sr-left。

基于 transformer 的方法(BERT+Bi-LSTM[12]) 首先把每一个句子表示为单词序列，接着将其送入 BERT 的 tokenizer 得到子词序列，并在句首加入[cls]字符。然后将子词序列送入 $BERT_{Base}$，取出 768 维的[cls]字符向量表示，作为该句子的向量表示。在不微调和微调 $BERT_{Base}$ 这两种情况下，实验得到了 BERT+Bi-$LSTM_{nft}$ 和 BERT+Bi-LSTM 这两种基准模型。

3. 4 实验参数

词嵌入 词嵌入部分，除了 freq-sr、latent-sr、BERT+Bi-$LSTM_{nft}$ 和 BERT+Bi-LSTM 这四个模型，所有其它模型的单词初始化均使用 300 维的 Google News word2vec 预训练模型。

DSG-SEG 模型参数 图的构建部分，滑动窗口大小 *window_size*=3；基于 GGNN 网络的句子编码部分，模型中的 *W*、*U* 和 *b* 等可训练参数的维度都被设置为 300 维，模型的迭代次数 *t*=2,*dropout_rate*=0.3，激活函数使用 Tanh；分割点预测部分，实验设置 Bi-LSTM 的隐藏层维度 *hidden_dim*=256。

批处理大小 & 训练 & 预测 在训练 BERT+Bi-LSTM 模型时，由于显存的限制，*batch size* 被设置为 3，训练其它模型时，*batch size* 均被设置为 8；训练阶段，实验使用 Adam[28]优化器，学习率设置为 0.001，当模型在验证集上的表现超过 5 轮没有提升后，训练结束；与 TextSeg 一样，模型设置一个阈值，对于一个句子，当它的预测概率超过该阈值时，其预测标签为 1，否则为 0。所有模型在验证集上优化该阈值，在测试集上使用最佳阈值进行预测。

实验设备 所有实验都在 Tesla M40 24GB 上进行。

3. 5 实验结果

为了保证结果的可靠性，实验使用 5 组不同的 *seed* 来得到所有模型在所有数据集上的平均 P_k 结果。

表 2 展示了所有模型在 WIKI-10K 的训练集上训练，然后在其它 8 种测试集上的评估结果。实验结果表明：(1)DSG-SEG 模型在 WIKI-10K、WIKI-SECTION、CHOI、CITES 和 WIKI-50 这 5 个测试集上同时取得了最好的结果，分别比基准模型中的最好结果提高了 3.4%、3.8%、11.2%、13.0%和 1.9%。在 CLINICAL 和 ELEMENTS 测试集上，DSG-SEG 与基准模型中的最好结果分别相差 4.4%和 0.4%。平均来看，DSG-SEG 比 BERT+Bi-$LSTM_{nft}$ 提升了 4.8%，比 TextSeg 提升了 6.3%；(2)latent-sr 和 freq-sr 在大部分测试集上取得了较差的结果，可能的原因是：它们的句子编码部分是无监督的；(3)有趣的是，BERT+Bi-LSTM 远远比不过 BERT+Bi-$LSTM_{nft}$，除此以外，tree-sr-left 也取得了较差的结果，出现上述情况的原因可能是：由于显存的限制，在训练这两个模型时，实验对句子长度 *sent_length*>40 的句子进行了截断，把篇章长度 *document_length*>60 的篇章截断成了多个子样本，这样的操作可能对结果产生了较大的影响；(4)相比之下，在所有的基准模型中，TextSeg 和 BERT+Bi-$LSTM_{nft}$ 取得了较好的结果，这表明了序列方法和基于 transformer 的方法在话题分割任务中的有效性；(5)所有模型在 MANIFESTO 这一

表 4 样例分割结果

MODEL	SEGMENTATION
golden	0 1 2 3 4 | 5 6 7 8 9 10 11 12 | 13 14 | 15 16 17 18 19 | 20 21 22 23 24
BERT+Bi-LSTM$_{nft}$	0 1 2 3 4 5 6 7 8 9 10 | 11 12 13 14 | 15 16 17 18 19 | 20 21 22 23 24
TextSeg[2]	0 1 2 3 4 5 6 7 8 | 9 10 11 12 13 14 15 16 17 18 19 | 20 21 22 23 24
DSG-SEG	0 1 2 3 4 | 5 6 7 8 | 9 10 11 12 13 14 | 15 16 17 18 19 | 20 21 22 23 24

测集上都取得了比 RANDOM BASELINE 还要差的结果，可能的原因是：MANIFESTO 中的篇章篇幅过长，分割点过于密集，因此，有监督的模型无法在该数据集上得到有效的迁移结果。

4 实验分析

4.1 消融实验分析

为了探究 DSG-SEG 在图的构建过程中：(1)把句子节点加入图和(2)单词节点之间的连边权重使用 PMI 指标，这两个细节对于模型性能的贡献度，下面将进行两组消融实验：(1)图中不包含句子节点，仅保留单词节点，单词节点之间的连边权重依然使用 PMI 指标(记作-SENTNODE)和(2)图中同时包含单词和句子节点，但所有邻接的单词节点之间的连边权重均设置为 1，而不使用 PMI 指标(记作-PMI)。值得注意的是，在(1)的情况下，由于图中不包含句子节点，句子之间也就不可能通过共有的邻接单词节点产生间接的信息交换。

消融实验结果如表 3 所示，可以看到：(1)把句子节点排除在图之外的-SENTNODE 消融模型，在除 CLINICAL 以外的其它 7 个数据集上，P_k 值都出现了不同程度的上升，平均来看，P_k 值上升了 1.90%。这一结果说明了：在图的构建部分，DSG-SEG 将句子节点加入图的有效性，模型允许句子节点之间通过共有的邻接单词节点产生间接的信息交换，能够同时考虑单词之间和篇章内部的非序列关系来得到具有全局语义信息的句子向量表示；(2)没有使用 PMI 指标的-PMI 消融模型，在除 WIKI-10K、CLINICAL 以外的其它 6 个数据集上，性能都有所下降，平均来看，P_k 值上升了 0.6%，这说明使用 PMI 指标能够为模型带来更加准确的语义信息。

4.2 样例分析

实验取出 WIKI-10K 测试集中的一个篇章(将其命名为《Bardwell Park》，其全文见附录中的图 3)来做样例分析。该篇章介绍了一个名叫巴德韦尔的公园，篇章总共分为 5 个段落：0-4，5-12，13-14，15-19，20-24，标注的话题标签分别是：历史，巴德韦尔公园与沃利溪山谷，商业区，交通，人口统计数据。表 4 从上到下分别是：该篇章正确的段落分割以及 BERT+Bi-LSTM$_{nft}$、TextSeg 和 DSG-SEG 这三个模型的分割预测结果。可以看到：(1)只有 DSG-SEG 预测出了第 4 句话这一分割点。事实上，0-4 段，与 5-12 段的前 4 句话都是在讨论巴德韦尔公园，差别在于：0-4 段介绍了它的历史，而 5-8 段介绍的是公民对它的环境保护运动。也就是说，三个模型中，只有 DSG-SEG 识别出了这一细微差别，并做出了正确的分割；(2)TextSeg 和 DSG-SEG 都错误地把第 8 个句子预测成了分割点，可能的原因如前所述：5-8 段和 9-12 段本质上都是在讨论环境保护问题，但前后段落的描述对象从巴德韦尔公园转移到了沃利溪山谷，这一点从 5-12 段所被标注的话题标签上也可以明显地看出；(3)三个模型都没有预测出 13-14 段，这可能说明：无论是序列方法、基于 transformer 的方法，还是本文提出的 DSG-SEG 模型，它们在预测较短的段落时，都不能准确地捕捉到话题转移的语义信息；(4)TextSeg 没能预测出 15-19 段，可能的原因是：序列方法把 13-14 段错误的语义信息传递了下来，进而导致了连续的分割错误；(5)总的来说，DSG-SEG 取得了更好的分割结果，因此，它具有更强的全局语义信息建模能力。

4.3 时间性能&参数量分析

实验统计了 TextSeg、BERT+Bi-LSTM$_{nft}$、BERT+Bi-LSTM 以及 DSG-SEG 这四个模型在 WIKI-10K 上训练一轮所需要的时间和各自的可训练参数量。如表 5 所示，可以看到：(1)由于双层 Bi-LSTM 的存在，TextSeg 的参数量较多，训

练结果较慢；(2)虽然 BERT+Bi-LSTM$_{nft}$ 并没有对 BERT$_{Base}$ 进行微调，参数量比 TextSeg 少得多，但由于 transformer 模型二次方的计算复杂度，它在话题分割任务中的时间性能依然很差；(3)BERT+Bi-LSTM 则具有最多的参数量和最差的时间性能；(4)相比之下，DSG-SEG 具有最少的参数量，运行速度也最快，其训练速度分别是 TextSeg 的 1.6 倍，BERT+Bi-LSTM$_{nft}$ 的 4.6 倍，BERT+Bi-LSTM 的 9.9 倍。

表 5 时间性能&参数量统计

MODEL	TIME(s)/EPOCH	PARAMS
TextSeg[2]	559	5874690
BERT+Bi-LSTM$_{nft}$	1639	3679247
BERT+Bi-LSTM[12]	3522	111989519
DSG-SEG	**356**	**3352870**

5 总结与展望

本文首次在话题分割任务中使用图神经网络，提出了一个基于篇章结构图网络的话题分割模型（DSG-SEG）。首先，模型把每一个篇章单独构建成图，图中包含了单词、句子节点，以及单词节点、单词和句子节点之间的邻接关系；接着，模型把建好的图作为输入，使用 GGNN 网络对其进行迭代，句子节点之间通过共有的邻接单词节点产生了间接的信息交互；最终，模型得到了具有全局语义信息的句子向量表示，并将其送入 Bi-LSTM 网络进行分割点的预测。实验结果表明，本文提出的 DSG-SEG 模型能够在多个数据集上同时取得最好的结果指标和时间性能。在接下来的工作中，我们会探索更适合于话题分割的句子向量表示方法；另外，我们将尝试把话题分割建模成分割点的预测和其它任务同时进行的联合任务；我们也将尝试在中文语料集上进行实验。

参考文献

[1] Tur G, De Mori R. Spoken language understanding: Systems for extracting semantic information from speech[M]. New Jersey: John Wiley & Sons, 2011.

[2] Koshorek O, Cohen A, Mor N, et al. Text segmentation as a supervised learning task[C]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018: 469-473.

[3] Hearst M A. Text Tiling: Segmenting text into multi-paragraph subtopic passages[J]. Computational linguistics, 1997, 23(1): 33-64.

[4] Dennis S, Landauer T, Kintsch W, et al. Introduction to latent semantic analysis[C]//Proceedings of the 25th Annual Meeting of the Cognitive Science Society, 2003: 25.

[5] Choi F Y Y. Advances in domain independent linear text segmentation[C]//Proceedings of the 6th Applied Natural Language Processing Conference(ANLP), 2000: 26-33.

[6] Glavaš G, Nanni F, Ponzetto S P. Unsupervised text segmentation using semantic relatedness graphs[C]//Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics, 2016.

[7] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. the Journal of machine Learning research, 2003, 3: 993-1022.

[8] Barrow J, Jain R, Morariu V, et al. A joint model for document segmentation and segment labeling[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020: 313-322.

[9] Glavaš G, Somasundaran S. Two-Level Transformer and Auxiliary Coherence Modeling for Improved Text Segmentation[C]//Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, 2020: 2306-2315.

[10] Badjatiya P, Kurisinkel L J, Gupta M, et al. Attention-based neural text segmentation[C]//Proceedings of the 2018 European Conference on Information Retrieval, 2018: 180-193.

[11] Li J, Sun A, Joty S R. SegBot: A Generic Neural Text Segmentation Model with Pointer Network[C]//Proceedings of the 2018 International Joint Conference on Artificial Intelligence(IJCAI), 2018: 4166-4172.

[12] Lukasik M, Dadachev B, Simoes G, et al. Text segmentation by cross segment attention[C]//Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing(EMNLP), 2020: 4707-4716.

[13] Xing L, Hackinen B, Carenini G, et al. Improving Context Modeling in Neural Topic Segmentation[C]//Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing(AACL/IJCNLP), 2020: 626-636.

[14] Tai K S, Socher R, Manning C D. Improved semantic representations from tree-structured long short-term memory networks[C]//Proceedings of the 53rd Annual Meeting of the Association for Computational Linguis, 2015: 1556-1566.

[15] Shi H, Zhou H, Chen J, et al. On tree-based neural sentence modeling[C]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018: 4631-4641.

[16] Vaswani A, Shazeern N, Parmar N, et al. Attention is all you need[C]//Proceedings of the 31st International Conferenceon Neural Information Processing Systems, 2017: 5998-6008.

[17] Yao L, Mao C, Luo Y. Graph convolutional networks for

text classification[C]//Proceedings of the 33rd AAAI Conference on Artificial Intelligence, 2019: 7370-7377.
[18] Zhang Y, Yu X, Cui Z, et al. Every document owns its structure: Inductive text classification via graph neural networks[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics(ACL), 2020: 334-339.
[19] Li Y, Tarlow D, Brockschmidt M, et al. Gated graph sequence neural networks[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics(ACL), 2020: 273-283.
[20] Pennington J, Socher R, Manning C D. Glove: Global vectors for word representation[C]//Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014: 1532-1543.
[21] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[C]//Proceedings of the 5th International Conference on Learning Representations(ICLR), 2017.
[22] Huang L, Ma D, Li S, et al. Text level graph neural network for text classification[C]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing(EMNLP-IJCNLP), 2019: 3442-3448.
[23] Sun Z, Jiang F, Li P, et al. Macro Discourse Relation Recognition via Discourse Argument Pair Graph[C]//Proceedings of the 2020 CCF International Conference on Natural Language Processing and Chinese Computing, 2020: 108-119.
[24] Arnold S, Schneider R, Cudré-Mauroux P, et al. Sector: A neural model for coherent topic segmentation and classification[J]. Transactions of the Association for Computational Linguistics, 2019, 7: 169-184.
[25] Malioutov I, Barzilay R. Minimum cut model for spoken lecture segmentation[C]//Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics, 2006: 25–32.
[26] Chen H, Branavan S R K, Barzilay R, et al. Global models of document structure using latent permutations[C]//Proceedings of the North American Chapter of the Association of Computational Linguistics, 2009: 371-379.
[27] Beeferman D, Berger A, Lafferty J. Statistical models for text segmentation[J]. Machine learning, 1999, 34(1): 177-210.
[28] Kingma D P, Ba J. Adam: A method for stochastic optimization[C]//Proceedings of 3rd International Conference on Learning Representation, 2014.

6 附录

《Bardwell Park》的全文如图 3 所示。

Bardwell Park was named after free settler Thomas Hill Bardwell who owned land in the area.
His grant was originally heavily timbered and bounded by Wolli Creek, Dowling Street and Wollongong Road.
In 1881, the land was auctioned and were subdivided.
The railway station opened on 21 September 1931 which opened up the area for home sites.
The school opened in September 1943 and the post office opened in May 1946.
Bardwell Park borders an important piece of remnant bushland, the Wolli Creek Valley, beside Wolli Creek.
There have been active citizens' movements lobbying for its preservation in the face of demands for urban expansion.
The most public of these prevented the building of the M5 South Western Motorway through the valley, resulting in the road being built as a tunnel under the valley.
Nevertheless, community concern remains over what is seen as the release of particle pollution from exhaust emissions into the atmosphere in the Bardwell Valley.
The Wolli Creek Valley contains the only bushland of any size left in inner south-west Sydney.
It is also the only large undeveloped natural space that remains in a heavily developed residential and industrial region.
The park offers public transport access family picnic areas, parkland, birdwatching, bushwalking, extensive views of sandstone escarpments, heathland and woodland forest.
A 60 hectare regional park is under development.
Bardwell Park is a leafy, predominantly residential area but features a small shopping centre around Hartill-Law 14 Avenue and Slade Road, beside the Bardwell Park railway station.
The Bardwell Park - Earlwood RSL is also located beside the railway station and includes a new gym and the club has undergone a renovation in October 2011.
Bardwell Park railway station is on the Airport, Inner West & South Line of the Sydney Trains network.
Bardwell Park is also serviced by 2 bus services, including State Transit Authority route 473 from Campsie to Rockdale and route 491 from Five Dock to Hurstville.
The M5 South Western Motorway runs beneath parts of Bardwell Park in a 4 km tunnel.
The nearest entrances to travel south west towards Beverly Hills and Liverpool are located at Arncliffe and Bexley North.
The nearest entrances to travel north east towards Botany and the city are located at Kingsgrove and Arncliffe.
According to the 2011 census of Population, there were 2,266 residents in Bardwell Park.
62 9% of residents were born in Australia.
The most common other countries of birth were China 6.5%, Greece 5.6% and England 2.1%.
50.0% of residents spoke only English at home.
Other languages spoken at home included Greek 19.3%, Mandarin 5.6% and Arabic 4.0%.

图 3 《Bardwell Park》