基于图神经网络的情感分析调研

吴林志

June 28, 2020

传统的机器学习算法所处理的几乎都是结构规整的欧式数据，例如文字的序列结构、图片的平面结构，而在现实世界中更多的数据表示并不是序列或者平面等简单的排列，而是表现为更为复杂的图结构，如社交网络、电商网络、文献引用网络等不规则的非欧式数据。我们可以把这类数据创造性地转换成由节点及连接节点的边构成的图结构。图神经网络(Graph Neural Networks, GNNs)是采用端到端的方式解决图相关任务的深度学习模型，由于其较好的性能和可解释性，最近已成为一种广泛应用的图分析方法。

在社交网络场景下，我们往往需要抽取观点持有者对评价对象的性(粗粒度或细粒度)情感倾向，而这其中很大一部分是源于对文本情感的极性分析。考虑到GNNs对处理图结构等非序列化数据存在绝对优势，本文结合几篇代表性文章，对GNNs在情感分析/文本分类任务中的应用进行分析和总结。

1 GNN概述

首先对有关GNNs的典型模型、benchmark图数据集及GNNs在NLP领域的应用做一个简单总结。

1.1 模型

GNNs的概念最初由Scarselli et al. (2009) [1]提出，其扩展了现有的神经网络，用于处理图领域的数据。图中的每个节点由其自身特征和其他相关节点来定义。以图结构和节点的特征信息作为模型输入，GNNs的输出一般可分为节点级别(与节点回归和分类任务相关)、边级别(与边分类和链接预测任务相关)和图级别(与图分类任务相关)。下面总结了几类典型的GNN模型[1]：

[1]更多相关文献请参见:
https://github.com/nnzhan/Awesome-Graph-Neural-Networks

- **图循环神经网络(Graph Recurrent neural Networks, GRNN)** 对GNNs的研究最早是从GRNN开始的[1][2][3]，GRNN旨在通过循环神经结构学习节点表示，为了学习目标节点的表示，通过迭代地与其邻居节点交换信息，直到达到稳定的状态。该过程在计算上有很高代价，最近已经做出更多努力来克服这些挑战[3]。

- **图卷积神经网络(Graph Convolutional neural Networks, GCN)** 不同于传统CNN网络基于网格数据的卷积操作，GCN在图结构数据上执行卷积，主要的思想是通过整合节点自身的特征和邻居节点的特征来生成当前节点的新表示。主流的方法有两类：基于谱(spectral)的方法和基于空间(spatial)的方法，其中基于谱的方法需要学习的参数取决于(全)图的结构，不能应用于具有不同结构的图(无法采用同一套参数来学习)，同时也难以并行或扩展到大图上；而基于空间的方法通常关注节点的空间关系，采用中心节点表示和其邻居表示的聚合来获得该节点的新表示(不是整个图中执行卷积)。基于谱的方法由Bruna et al. (2013) [4]提出，此后基于谱的GCN的改进和扩展不断涌现[5][6][7][8][9][10]，其中比较具有代表性的为Kipf and Welling (2017) [7] 提出的GCN，而基于空间的GCNs也是异军突起[11][12][13][14][15][16]，其中代表性的为Hamilton et al. (2017) [14]提出的GraphSAGE和Velikovi et al. (2018) [15]提出的GAT。

- **图自编码器(Graph AutoEncoders, GAE)** GAE是无监督的学习框架，可将节点或图编码成潜在的向量空间，并从编码的信息中重构图数据。常用于学习网络嵌入(Network Embedding)[17][18][19]和图的生成(Graph Generation)[20][21][22][23][24]。

- **图时空网络(Graph Spatial-Temporal neural Networks, GST)** GST旨在从时-空图中学习隐藏模式，能捕捉图的动态性(图结构或输入的变化)，这在各种现实应用中变得越来越重要，如交通速度预测和人类行为识别，其关键思想是同时考虑空间依赖性和时间依赖性[25][26][27]。

1.2 数据集

表1总结了常用的(图结构)benchmark数据集

1.3 NLP应用

1.3.1 相关工作

表5展示了GNNs模型应用于具体NLP任务中的统计结果[2]

[2]更多应用请参见: https://github.com/IndexFziQ/GNN4NLP-Papers

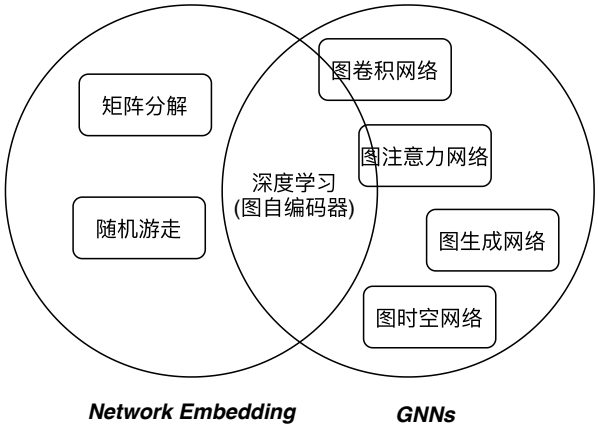

Figure 1: 网络嵌入和图神经网络的区别

Table 1: Summary of benchmark datasets

数据集	**Cora**	**Citeseer**	**Pubmed**
类型	Citation	Citation	Citationn
节点数	2708	3327	19717
边数	5429	4732	44338
类别数	7	6	3
特征数	1433	3703	500
任务	Transductive	Transductive	Transductive
引用	[28]		
数据集	**NELL**	**Reddit**	**PPI**
类型	Knowledge Graph	Social	Biology
节点数	65755	232965	56944(含24个图)
边数	266144	11606919	818716
类别数	210	41	121(多标签)
特征数	5414	602	50
任务*	Transductive	Inductive	Inductive
引用	[29]	[14]	

*Inductive任务是指训练过程中，测试数据不可见(测试集只用于评估模型)
Transductive任务是指训练过程中，测试数据可见(测试集为训练提供无标签数据)

Table 2: Applications of GNNS in NLP

任务	模型	引用
文本分类	GCN	[14][7][11][5][6]
	GAT	[15]
	DGCNN	[30]
	Text GCN	[31]
语义角色标注	Syntactic GCN	[32]
依存句法分析	Syntactic GNN	[33]
机器翻译	GGNN	[34]
	Syntactic GCN	[35][36]
关系抽取	GCN	[37]
事件抽取	Syntactic GCN	[38][39]
AMR-to-Text Generation	Graph LSTM	[40]
	GGNN	[34]
关系推理	Interaction Network	[41]

1.3.2 开源实现

表3汇总了几类典型模型开源实现的链接和两大开发深度学习图模型的开源框架

Table 3: A Summary of Open-source Implementations

模型	Github
GraphSAGE[14]	https://github.com/williamleif/graphsage-simple
GCN[7]	https://github.com/tkipf/pygcn
GAT[15]	https://github.com/Diego999/pyGAT
DGCNN[16]	https://github.com/muhanzhang/pytorch_DGCNN
GIN[42]	https://github.com/weihua916/powerful-gnns
DGI[43]	https://github.com/PetarV-/DGI
Cluster-GCN[44]	https://github.com/benedekrozemberczki/ClusterGCN
ST-GCN[27]	https://github.com/yysijie/st-gcn
GraphRNN[22]	https://github.com/snap-stanford/GraphRNN
DiffPool[45]	https://github.com/RexYing/diffpool
PyTorch Geometric[46]	https://github.com/rusty1s/pytorch_geometric
Deep Graph Library[47]	https://github.com/dmlc/dgl

2 文本级(Text-Level)文本分类

Text GCN [31]是一种典型的固定语料库级图结构(Corpus-Level)，也就是为整语料库构建一个异构图，包括(待分类)文档节点和单词节点,边的权重是

固定的(单词节点间的边权重为两个单词的PMI，文档-单词节点间的边权重是TF-IDF)，固定权重限制了边的表达能力，同时为了获取一个全局表示不得不使用一个非常大的连接窗口。因而，构建的图非常大，而且边非常多，模型有很高的内存消耗。最关键的是，Text GCN这种类型的模型，无法为新文本进行分类(在线测试)，因为图的结构和参数依赖于语料库，训练结束后就不能再修改了(除非将新文本加入到语料库中，更新图的结构，重新训练，但是一般不会这么做)。

Huang et al. (2019) [48]等人提出为每个输入文本(text-level)都单独构建一个图，文本中的单词作为节点，而不是给整个语料库(corpus-level)构建一个大图(每个文本和单词作为节点)。在每个文本中，使用一个非常小的滑动窗口，文本中的每个单词只与其左右的若干个词有边相连(包括自己，自连接)，而不是所有单词节点全连接。相同单词节点的表示以及相同单词对之间边的权重全局(语料库中的所有文本)共享，通过文本级别图的消息传播机制进行更新。这样就可以消除单个输入文本和整个语料库的依赖负担，支持在线测试(新文本测试)，而且上下文窗口更小，边数更少，内存消耗更小。

2.1 方法

2.1.1 构建文本图

设一个包含l个单词的文本为$T = \{r_1, \cdots, r_i, \cdots, r_l\}$，$r_i$表示文本中第$i$个单词的向量表示，初始化为一个全局共享的词嵌入矩阵(d维)，每个单词(节点)的初始表示从该嵌入矩阵中查询，嵌入矩阵作为模型参数在训练过程中更新。

为每个输入文本构建一个图，把文本中的单词看作是节点，每个单词与其左右相邻的p个单词有边相连(包括自身)。输入文本的图表示为：

$$\begin{aligned} N &= \{\mathbf{r_i} | i \in [1, l]\}, \\ E &= \{e_{ij} | i \in [1, l]; j[i-p, i+p]\}, \end{aligned} \tag{1}$$

其中，N和E是文本图的节点集和边集，每个单词节点的表示以及单词节点间边的权重分别来自两个全局共享矩阵(训练过程中更新)。此外，对于训练集中出现次数少于$k(k=2)$次的边(词对)均匀地映射到一个"公共边"，使得参数得到充分学习。该方法相比于Text GCN在节点和边的角度缩减了图的规模，内存消耗更小，而且可以为新文本进行分类，每个文本对应的图在内容上是独立的。

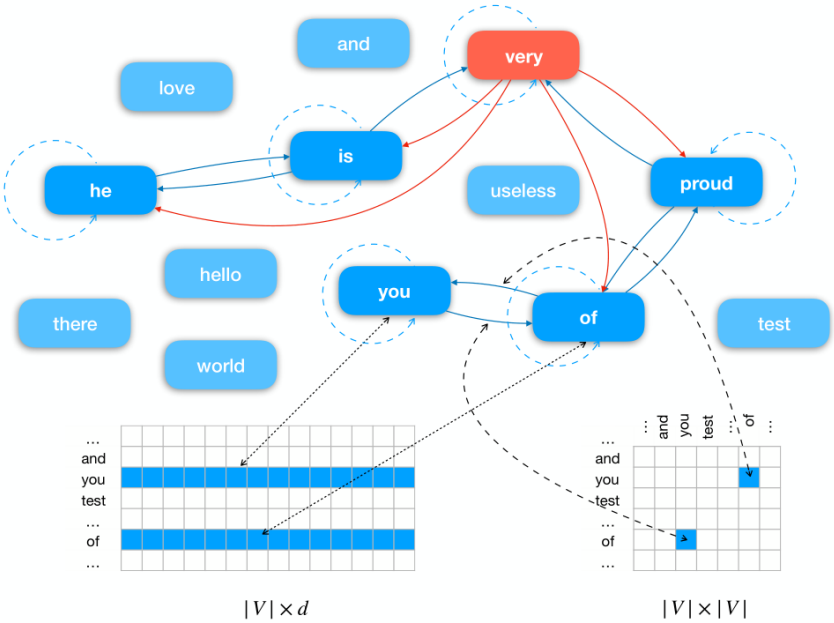

Figure 2: 单文本"he is very proud of you."的图结构。对于节点"very"，p设为2，其他节点p设为1(实际场景下，p值是一致的)；文本图中所有参数来自于全局共享的矩阵。

2.1.2 消息传递机制(MPM)

对于文本图中每个单词节点，MPM首先基于节点之前的表示以及从该节点的邻接节点收集的信息来更新该节点的表示，定义如下：

$$\begin{aligned} \mathbf{M_n} &= \max_{a \in \mathcal{N}_n^p} e_{an} \mathbf{r_a}, \\ \mathbf{r_n'} &= (1 - \eta_n)\mathbf{M_n} + \eta_n \mathbf{r_n} \end{aligned} \tag{2}$$

$\mathbf{M_n} \in \mathbb{R}^d$是节点$n$从它的邻接节点$a$获取的信息，即每个邻接节点$a$的表示乘以它们之间的权重$e_{an} \in \mathbb{R}^1$，再按维度取max pooling，得到一个新的$d$维表示。$\mathbf{r_n}$是节点$n$之前的表示，可训练参数$\eta_n \in \mathbb{R}^1$表示$\mathbf{r_n}$中多少信息被保留，$\mathbf{r_n'}$表示节点$n$更新后的表示。

MPM机制通过邻接节点的表示生成当前节点的表示，意味着该表示从上下文获取信息。即使对于多义词，也能在特定上下文中通过其邻接节点的加权信息表示精准确定它的含义。此外，text-level图的参数来自全局共享矩阵，意味着该模型也可以像其他基于图的模型(如Text GCN)一样，其表示也携带着全局信息。

2.1.3 分类

对于一个文本图中的所有节点的表示通过下式，得到对该文本的预测：

$$y_i = softmax(ReLU(\mathbf{W} \sum_{n \in N_i} \mathbf{r}_\mathbf{n}^{'} + \mathbf{b})) \tag{3}$$

其中，$\mathbf{W} \in \mathbb{R}^{d \times c}$是把向量表示映射到输出空间的矩阵，$N_i$是文本$i$的节点集，$\mathbf{b} \in \mathbb{R}^c$为偏置项。

训练的目标是最小化模型预测和真实标签g_i(one-hot向量表示)的交叉熵损失：

$$\boldsymbol{loss} = -g_i \log y_i \tag{4}$$

2.2 实验

在相同的数据集上(R8, R52和Ohsumed)，作者将提出的方法与CNN、LSTM、fastText、Graph-CNN和Text GCN等基准模型(使用默认参数设置)进行比较。实验表明：基于图形的模型的结果要优于CNN，LSTM和fastTest等传统模型。这可能是由于图结构的特性：图结构允许存在不同数量的邻居节点，这使单词节点可以通过不同的搭配学习更多的准确表示。此外，单词之间的关系可以记录在边的权重中并全局共享，这些对于传统模型都是不可能的。同时作者的模型比Graph-CNN和Text GCN性能更好，达到SOTA性能，并在内存消耗方面具有显著优势。

同时，实验发现随着邻接词数量p变大，精度会提高，并且当p变大时，将达到最佳性能(p=3)。然后，精度随着p的增加而降低了波动性。这表明，当仅与最近的邻域连接时，节点无法理解上下文中跨越多个单词的依赖关系，而与较远的邻域连接(p更大)时，图忽略了局部特征。此外，边越少，内存消耗越小。与固定边的权重相比，可训练边可以更好地模拟单词之间的关系。

3 方面级(Aspect-Level)情感分析

方面级情感分析是更细粒度的任务，给定一个句子和句子中出现的某个Aspect(词或短语)，目标是分析出这个句子在给定Aspect上的情感倾向[3]。例如，"great food but the service was dreadful"，其中"food"的情感极性为正面，"service"的情感极性为负面。之前大多数神经网络模型将一个句子视为一个词序列，并通过各种方法(如注意力[49]或门机制[50])将aspect信息嵌入到句子表示中。这些方法很大程度上忽略了句子的句法结构，句法

[3]注：target-dependent情感分析与aspect-level情感分类类似，不同之处在于，target是明确出现在句子中的词，而aspect是更宽泛的一个概念，不一定出现在句子中。

信息在某种程度上有助于识别与aspect目标直接相关的情感特征。例如，句子"The food, though served with bad service, is actually great"，当aspect 词与其情感短语分开时，很难在序列中找到相关的情感词。而在依存图中，情感词"great"离aspect词"foot"更近。使用依存关系还有助于解决单词序列中的潜在歧义问题，例如句子"Good food bad service"，"good"和"bad"可以替换着使用。采用基于注意力的方法很难区分"good"和"bad"中哪个和"food"与"service"相联系，而有了句法知识可以很容易认识到"good (bad)"是"food (service)"的形容词修饰语。

Huang and Carley (2019) [51]提出了一种新的目标相关(target-dependent)图注意力网络(TD-GAT)用于方面级别的情感分类，并显式利用了词之间的依存关系。不同于先前的方法，作者的方法将一个句子表示成依存图而不是词序列。在依存图中，aspect目标和相关的词将被直接联系在一起。作者利用多层GAT网络将情感特征从重要的句法上邻近词传播到aspect词，同时进一步整合LSTM来显式捕捉(跨层)aspect相关的信息。类似地，Zhang et al. (2019) [52]提出在句子的依存树上构建图卷积网络(GCN)，以利用句法信息和单词依赖。在此基础上，提出一种特定aspect的情感分类框架，实验证明了利用句法信息和远程单词依存关系的重要性。本报告对后者不展开叙述和讨论。

3.1 方法

3.1.1 文本表示

给定长度为n的句子$s = [w_1, w_2, \cdots, w_i, \cdots, w_n]$和一个aspect目标词$w_i$，首先将每个词$w_i$映射到低维的词嵌入向量$x_i \in \mathbb{R}^d$。作者使用现成的斯坦福依存解析器[53]将句子转换成依存图，每个节点代表一个单词，并与一个嵌入向量相关联，作为其局部特征向量。两个单词之间的无方向边意味着这两个单词在语法上相关。如图3所示，对于目标词"delivery"，我们可以将特征从它的2-hop邻居传播到1-hop邻居，再到它自己。

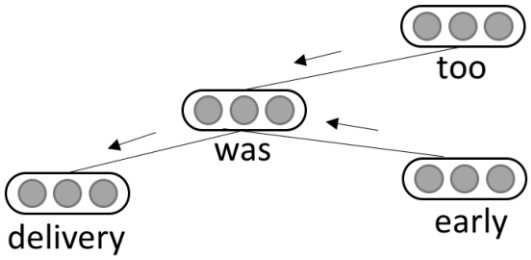

Figure 3: "delivery was early too"的依存图。特征可以从邻居节点传播到aspect节点"delivery"。

当aspect目标不止一个词时，首先将整个目标词序列用一个特殊符

号”__target__”替换，然后修改后的句子经过依存解析器进行解析。aspect的表示是其组成词的词向量的平均值。

3.1.2 GAT

文中GAT网络将特征从aspect的句法上下文传播到aspect节点。给定一个有N个节点的依存图，每个节点关联一个局部词向量x，在GAT层通过聚合邻近节点的隐层状态来计算节点表示。使用一个L层的GAT网络，L跳(hops)的特征可以传播到aspect目标节点。特征传播过程可记作:

$$H_{l+1} = GAT(H_l, A; \theta_l) \tag{5}$$

其中，$H_l \in \mathbb{R}^{N \times D}$是第$l$层所有节点的堆叠(stacked)状态，$A \in \mathbb{R}^{N \times N}$是图的邻接矩阵，$\theta_l$是GAT第$l$层参数集。

3.1.3 TD-GAT

为了明确地在这样的GAT网络中利用目标信息，作者进一步使用LSTM对跨层的aspect目标进行依赖性建模，这也有助于克服图中的噪声信息。目标独立(target-dependent)的GAT网络前向处理过程可记作:

$$\begin{aligned} H_{l+1}, C_{l+1} &= LSTM(GAT(H_l, A; \theta_l), (H_l, C_l)) \\ H_0, C_0 &= LSTM(XW_p + [b_p]_N, (0, 0)) \end{aligned} \tag{6}$$

其中，C_l是第l层LSTM的堆叠(stacked) 细胞(cell)状态，LSTM的初始隐层状态和细胞状态设为0。$W_p \in \mathbb{R}^{d \times D}$矩阵将堆叠的嵌入向量$X$转化成隐层状态的维度，$[b_p]_N$表示堆叠偏置向量$b_p$ N次，并形成维度为$\mathbb{R}^{N \times D}$的偏置矩阵。

3.1.4 分类

在L层的TD-GAT网络中，我们获得了aspect目标节点的最终表示。我们仅从所有节点表示H_L中获取aspect目标节点的对应隐藏状态h_L^t(第L层的目标节点t)，然后经过一个线性层进行分类。aspect目标的最终预测情感极性是概率最高的标签，并通过最小化带L2正则化的交叉熵损失来训练模型。

3.2 实验

作者采用选自SemEval 2014任务四的两个通用的评论数据集:laptop和restaurant(每个数据项都是一个句子和一个aspect词组成的对)来检验模型的有效性。对于图中的每个节点作者采用300维的Glove向量，同时还使用了预训练的英

文BERT表示(”[CLS]”+句子+”[SEP]”+aspect+”[SEP]”)来进一步提升模型性能。

为了验证提出方法的有效性，作者与系列基线方法做对比：

Feature-based SVM 基于n-gram特征和字典特征的SVM[54]。

TD-LSTM 使用两个LSTM分别对aspect词的前后上下文进行建模(本文采用GAT来建模aspect词的句法上下文)，LSTM网络的末隐层状态拼接在一起来预测情感极性[55]。

AT-LSTM 首先通过LSTM对句子进行建模，然后将LSTM中的隐藏状态与aspect嵌入向量相结合，以生成注意力向量，最终的句子表示是隐藏状态的加权和[49]。

MemNet 对词嵌入多次施加attention，最后attention的输出喂给softmax层进行预测[56]。

IAN 使用两个LSTM网络分别对句子和aspect进行建模。使用句子中的隐藏状态为target生成注意力向量，反之亦然。基于这两个注意力向量，它输出句子表示和target表示来进行分类[57]。

PG-CNN 基于CNN模型，aspect特征用作门来控制句子的特征提取[50]。

AOA-LSTM 引入基于attention-over-attention网络以一种联合方式来建模aspect和句子，显示捕捉aspect和上下文句子间的交互[58]。

BERT-AVG 使用BERT句子表示(不微调)的均值来训练一个线性分类器。

BERT-CLS 直接使用”[CLS]”表示作为分类特征，以对用于paired句子分类的BERT模型进行微调。

实验结果(表4)表明：相比于直接的竞争对手TD-LSTM，作者的模型表现更好，直接证明了整合句法信息的必要性。采用BERT表示进一步提升模型性能，微调的BERT-CLS相比于BERT-AVG性能更好，但是这种微调很不稳定。尽管原始的BERT模型已经有出色的表现，但作者提出的模型整合BERT-AVG和BERT-CLS后性能会进一步提升，表明模型能更好地利用这些语义表示。

4 用图嵌入增强BERT

使用注意力机制(例如BERT)的模型已表明具有捕获句子或文档上下文信息的能力。但是，它们捕获有关语言词汇的全局信息的能力很有限，而这正是GCN的优势。例如：”Although it’s a bit smug and repetitive, this

	Laptop	Restaurant
Feature+SVM	70.5	80.2
TD-LSTM	68.1	75.6
AT-LSTM	68.9	76.2
MemNet	72.4	80.3
IAN	72.1	78.6
PG-CNN	69.1	78.9
AOA-LSTM	72.6	79.7
TD-GAT-GloVe (3)	73.7	81.1
TD-GAT-GloVe (4)	**74.0**	80.6
TD-GAT-GloVe (5)	73.4	**81.2**
BERT-AVG	76.5	78.7
BERT-CLS	77.1	81.2
TD-GAT-BERT (3)	79.3	82.9
TD-GAT-BERT (4)	79.8	**83.0**
TD-GAT-BERT (5)	**80.1**	82.8

Figure 4: 在laptop和restaurant数据集上比较不同方法的结果。括号中的数字代表模型层数。

documentary engages your brain in a way few current films do”，消极和积极的观点都出现在这个句子中，但是”a way few current films do”是积极的态度，用隐性的方式表达电影有种非常强的创新特点。如果没有在电影评论的上下文中将此表达更明确地与“创新”的含义联系起来，则分类器可能会忽略这种强烈的观点，因此该句子可能被错误地归类为否定。在这个示例中，将此表达关联到句子中其他tokens的自注意机制可能起不到作用(无法明确获取到有关语言(词汇)的知识)，而GCN能够学习词和概念之间的全局上下文(词汇)信息，可以将表达”a way few current films do”关联到”innovation”。

Lu et al. (2020) [59]等人提出了VGCN-BERT模型，该模型将BERT与Vocabulary GCN(VGCN)相结合。局部信息和全局信息通过BERT的不同层进行交互，从而使它们相互影响并共同构建最终的表示用于分类。

4.1 方法

4.1.1 Vocabulary Graph

作者采用文档中的词共现来构建词汇图，具体采用归一化的PMI，即NPMI，

词i和词j的NPMI值计算如下：

$$
\begin{aligned}
NPMI(i,j) &= -\frac{1}{\log p(i,j)} \log \frac{p(i,j)}{p(i)p(j)}, \\
p(i,j) &= \frac{\#W(i,j)}{\#W}, \\
p(i) &= \frac{\#W(i)}{\#W}
\end{aligned}
\tag{7}
$$

$\#W$为滑动窗口的总数量，$\#W(i)$为包含单词i的滑动窗口数量 (在一个语料库中)，$\#W(i,j)$为同时包含单词i和单词j的滑动窗口的数量。为了获得长距离依赖，作者将窗口大小设成这个句子长度；NPMI的取值范围在[-1, 1]之间，NPMI值为正表明词之间有较高语义相关性，为负表明有较小或没有语义关系；文中作者采用的方法是，当两个词的NPMI值大于某个阈值(0到0.3之间)时，在二者之间建立一条边。

4.1.2 Vocabulary GCN (VGCN)

通用GCN[7]是多层(通常是2层)神经网络，直接在图上进行卷积。给定一个图G=(V, E) ($|V| = n$)，对于一个单层的GCN，节点的新表示计算方法如下：

$$
\begin{aligned}
H &= \tilde{A}XW, \\
\tilde{A} &= D^{-\frac{1}{2}}AD^{-\frac{1}{2}}, \\
D_{ii} &= \sum\nolimits_j A_{ij}
\end{aligned}
\tag{8}
$$

其中，$X \in \mathbb{R}^{n \times m}$为n个节点且特征维度为m的输入矩阵，$W \in \mathbb{R}^{m \times h}$为权重矩阵，$\tilde{A}$为归一化的对称邻接矩阵[4]。

作者的目的是对相关的词而非语料中的文档进行卷积来做分类，因此作者提出的GCN图的构建是基于**词表**而不是文档。于是，对于一个单文档，设该文档是由词表中的词构成的一个行向量$\mathbf{x}$，则单层卷积定义如下：

$$
\begin{gathered}
\mathbf{h} = (\tilde{A}\mathbf{x}^T)^T W = \mathbf{x}\tilde{A}W, \;\; \tilde{A}^T = \tilde{A} \\
\mathbf{x} \in 1 \times |V|, \tilde{A} \in |V| \times |V|, W \in |V| \times h
\end{gathered}
\tag{9}
$$

其中，$\mathbf{x}\tilde{A}$提取与输入句子x相关的词表图的部分。对于m个文档组成的mini-batch，单层图卷积变成：

$$
H = X\tilde{A}W \tag{10}
$$

[4]邻接矩阵A的归一化操作是为了避免在使用深度神经网络模型时出现数值不稳定以及梯度爆炸或消失

相应的2层VGCN为：

$$\boldsymbol{VGCN} = ReLU(X_{mv}\tilde{A}_{vv}W_{vh})W_{hc} \tag{11}$$

其中，m为batch大小(文档数量)，v为词表大小，h为隐层大小，c为类别数或句子嵌入维度。X_{mv}是包含文档特征的向量矩阵(bow vector / word embedding of BERT)。

4.1.3 整合VGCN到BERT

作者不仅使用BERT中输入句子的词嵌入，还通过方程11获得的词表图嵌入和词嵌入序列都输入到BERT transformer。这样，不仅保留了句子中单词的顺序，而且利用了VGCN获得的背景信息。通过自注意力得分的计算过程中，局部嵌入和全局嵌入会随着(12层12个头的编码器)一层一层的迭代而被完全整合在一起。相应的VGCN可以表示成：

$$\boldsymbol{G_{embedding}} = ReLU(X_{mev}\tilde{A}_{vv}W_{vh})W_{hg} \tag{12}$$

其中，g为图嵌入大小，其维度和每个词嵌入相同；m为batch大小；e为词嵌入维度；v为词表大小。

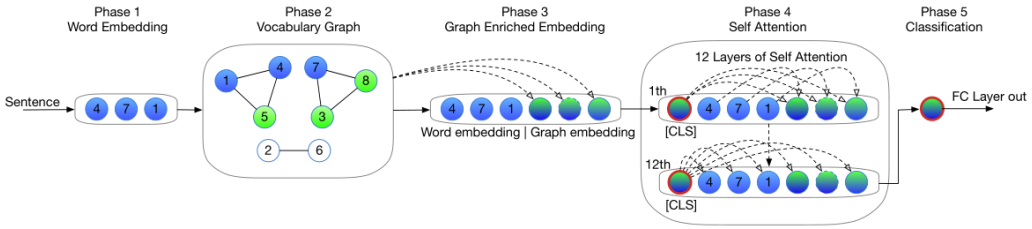

Figure 5: VGCN-BERT模型结构：将输入句子的嵌入(阶段1)与词表图(阶段2)结合起来以生成图嵌入，将其与句子输入(阶段3)拼接起来。需要注意的是词表图中仅提取和嵌入与输入相关的部分。在阶段4中，拼接的表示用于几个自注意力层，从而使得词嵌入和图嵌入之间的进行交互。末层的最终嵌入喂给全连接层(阶段5)用作分类。

4.2 实验

作者在5个数据集(SST-2 / MR / CoLA / ArangoHate / FountaHate)上评估VGCN-BERT并与基线模型做比较，基线模型有：MLP、BiLSTM、Text GCN、VGCN(只利用词表图的全局信息)、BERT和Vanilla-VGCN-BERT(分别获取BERT和GCN的表示后进行拼接)。

除了FountaHate数据集采用均方差损失函数外(为了利用标注员的voting信息)，所有模型作者采用交叉熵损失函数。由于CoLA、ArangoHate和FountaHate等

数据集的标签分布不均衡，作者采用带权损失函数，每个类别的权重计算方式为：

$$W_c = \frac{\#dataset}{\#classes \cdot \#every\_class} \tag{13}$$

其中，$\#dataset$为数据集大小，$\#classes$为类别数，$\#every_class$为每个类别的数量。此外，作者采用加权平均F1和宏F1来衡量分类器的性能：

$$\begin{aligned} \boldsymbol{Weighted\ avg\ F1} &= \sum_{i=1}^{C} F1_{c_i} * W_{c_i}, \\ \boldsymbol{Macro\ F1} &= \frac{1}{C}\sum_{i=1}^{C} F1_{c_i} \end{aligned} \tag{14}$$

实验结果表明：VGCN-BERT性能优于所有baseline模型，特别地，它的性能优于VGCN和BERT，这证明了 二者结合的优势。VGCN-BERT比Vanilla-VGCN-BERT表现更好，说明让局部和全局信息进行交互的好处。为了更好地理解BERT和BERT与VGCN的结合，作者可视化了BERT、VGCN-BERT和Vanilla-VGCN-BERT的自注意力模块中[CLS]标记的注意力分布(如图6)。 对于电影评论”Although it’s a bit smug and repetitive, this documentary

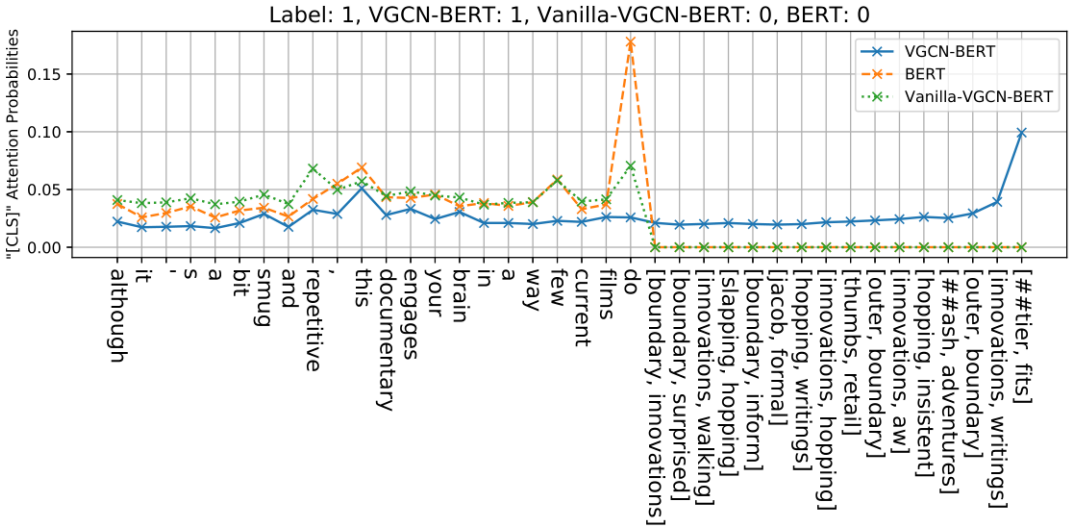

Figure 6: 可视化[CLS]标记(用作句子嵌入表示)对其他token的注意力。第一部分对应于句子的词嵌入，第二部分是图嵌入。由于BERT未使用图嵌入，对图嵌入(第二部分)的注意力比重为0。

engages your brain in a way few current films do”，句子的前半部分明显是负面的，而其余部分则隐含地表达了积极的态度，这使句子难以判断情感极

性。在此示例中，BERT非常关注”do”同时较多关注”this”，而这些词在情感表达上没有多大意义。 BERT的最终分类结果为0 (负面)，而真正的标签为1 (正面)。Vanilla-VGCN-BERT将图嵌入与BERT拼接在一起，而它们之间没有交互。我们可以看到，仍然没有attention图嵌入分布，这表明这种简单的组合不能有效地利用词汇(表)信息。对于VGCN-BERT，我们看到相当多的注意力集中在图嵌入上。通过逐步将句子中的局部信息与图中的全局信息整合在一起，可以生成图嵌入。最后，图嵌入的几个维度暗示着“创新”的含义，对此 赋予了较高的关注，这将导致句子分类为正确的类别(正面)。

5 总结

本文以NLP中的文本分类/情感分析为起点，讨论了GNNs(以GCN和GAT为代表)的具体应用和方法，这将在社交网络领域有极大的应用和发展空间。一般在应用GNNs模型时，形式上没有太大变动，基本上都是结合任务自身特点，套用现成的公式。而关键之处是如何有效建图，如Yao et al. (2019) [31]等人提出在整个语料库(文档)上建图，Huang et al. (2019) [48]等人为每个输入文本单独建图，Zhang et al. (2019) [52]和Huang and Carley (2019) [51]等人提出在句子的依存树上建图，Lu et al. (2020) [59]等人基于词表建图。

总的来说，图节点类型可以是词、句子、篇章(篇章的抽象表示节点)，以及一些抽取实体词并再处理的方法。连边策略方面，包括词与词之间是否共现、词与句子之间词是否出现在句中、句子与句子之间是否包含同样的词；其他还包括句法树，实体图等等。权重一般有(N)PMI、Cosine或者以共现次数为基础的公式计算等。如何选择构图方法还是和下游任务紧密相关的，像文本分类任务大多主要用词汇构图，推理任务用句子级节点就比较多，还有一些不太通用的构图方法高度适配特定任务，不太有泛化的借鉴价值。在特定的NLP任务场景下，如何采用更有效的建图方法仍有待进一步探究。

参考文献

[1] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," IEEE Transactions on Neural Networks, pp. 61–80, 2009.

[2] C. Gallicchio and A. Micheli, "Graph echo state networks," IJCNN, pp. 1–8, 2010.

[3] Y. Li, D. Tarlow, M. Brockschmidt, and R. S. Zemel, "Gated graph sequence neural networks," in ICLR, 2015.

[4] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," in ICLR, 2013.

[5] M. Henaff, J. Bruna, and Y. Lecun, "Deep convolutional networks on graph-structured data." arXiv: Learning, 2015.

[6] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in NLPS, 2016, pp. 3844–3852.

[7] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in ICLR, 2017.

[8] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, "Cayleynets: Graph convolutional neural networks with complex rational spectral filters," IEEE Transactions on Signal Processing, vol. 67, no. 1, pp. 97–109, 2019.

[9] R. Li, S. Wang, F. Zhu, and J. Huang, "Adaptive graph convolutional neural networks," in AAAI, 2018, pp. 3546–3553.

[10] C. Zhuang and M. Qiang, "Dual graph convolutional networks for graph-based semi-supervised classification," in WWW, 2018, pp. 499–508.

[11] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," in NLPS, 2016, pp. 1993–2001.

[12] M. Niepert, M. H. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in ICML, 2016, pp. 2014–2023.

[13] J. Gilmer, S. S. Schoenholz, P. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in ICML, 2017, pp. 1263–1272.

[14] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in NIPS, 2017, pp. 1024–1034.

[15] P. Velikovi, G. Cucurull, A. Casanova, A. Romero, P. Li ò , and Y. Bengio, "Graph attention networks," in ICLR, 2018.

[16] M. Zhang, Z. Cui, M. Neumann, and C. Yixin, "An end-to-end deep learning architecture for graph classification," in AAAI, 2018, pp. 4438–4445.

[17] S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations," in AAAI, 2016, pp. 1145–1152.

[18] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," KDD, pp. 1225–1234, 2016.

[19] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially regularized graph autoencoder for graph embedding," in IJCAI, 2018, pp. 2609–2615.

[20] T. N. Kipf and M. Welling, "Variational graph auto-encoders," NIPS Workshop on Bayesian Deep Learning, 2016.

[21] Y. Li, O. Vinyals, C. Dyer, R. Pascanu, and P. W. Battaglia, "Learning deep generative models of graphs," in ICML, 2018.

[22] J. You, Rex, X. Ren, W. L. Hamilton, and J. Leskovec, "Graphrnn: Generating realistic graphs with deep auto-regressive models," in ICML, 2018.

[23] A. Bojchevski, O. Shchur, D. Zugner, and S. Gunnemann, "Netgan: Generating graphs via random walks." in ICML, 2018.

[24] M. Simonovsky and N. Komodakis, "Graphvae: Towards generation of small graphs using variational autoencoders," ICANN, pp. 412–422, 2018.

[25] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," arXiv: Machine Learning, 2016.

[26] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in IJCAI, 2018, pp. 3634–3640.

[27] S. Yan, Y. Xiong, D. Lin, and X. Tang, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in AAAI, 2018, pp. 7444–7452.

[28] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassirad, "Collective classification in network data," Ai Magazine, vol. 29, no. 3, pp. 93–106, 2008.

[29] Z. Yang, W. W. Cohen, and R. Salakhutdinov, "Revisiting semi-supervised learning with graph embeddings," in ICML, 2016, p. 4048.

[30] H. Peng, J. Li, Y. He, Y. Liu, M. Bao, L. Wang, Y. Song, and Q. Yang, "Large-scale hierarchical text classification with recursively regularized deep graph-cnn," WWW, pp. 1063–1072, 2018.

[31] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," vol. 33, no. 01, pp. 7370–7377, 2019.

[32] D. Marcheggiani and I. Titov, "Encoding sentences with graph convolutional networks for semantic role labeling," in EMNLP, 2017, pp. 1506–1515.

[33] T. Ji, Y. Wu, and M. Lan, "Graph-based dependency parsing with graph neural networks," in ACL, 2019, pp. 2475–2485.

[34] D. Beck, G. Haffari, and T. Cohn, "Graph-to-sequence learning using gated graph neural networks," in ACL, 2018, pp. 273–283.

[35] J. Bastings, I. Titov, W. Aziz, D. Marcheggiani, and K. Simaan, "Graph convolutional encoders for syntax-aware neural machine translation," in EMNLP, 2017, pp. 1957–1967.

[36] D. Marcheggiani, J. Bastings, and I. Titov, "Exploiting semantics in neural machine translation with graph convolutional networks," pp. 486–492, 2018.

[37] Y. Zhang, P. Qi, and C. D. Manning, "Graph convolution over pruned dependency trees improves relation extraction," pp. 2205–2215, 2018.

[38] T. H. Nguyen and R. Grishman, "Graph convolutional networks with argument-aware pooling for event detection," pp. 5900–5907, 2018.

[39] X. Liu, Z. Luo, and H. Huang, "Jointly multiple events extraction via attention-based graph information aggregation," arXiv: Computation and Language, 2018.

[40] L. Song, Y. Zhang, Z. Wang, and D. Gildea, "A graph-to-sequence model for amr-to-text generation," 2018.

[41] P. W. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, and K. Kavukcuoglu, "Interaction networks for learning about objects, relations and physics," in NIPS, 2016, pp. 4509–4517.

[42] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in ICLR, 2019.

[43] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep Graph Infomax," in ICLR, 2019.

[44] W.-L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C.-J. Hsieh, "Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks," KDD, 2019.

[45] R. Ying, J. You, C. Morris, X. Ren, W. L. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in NeurIPS, 2018.

[46] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," in ICLR Workshop on Representation Learning on Graphs and Manifolds, 2019.

[47] M. Wang, L. Yu, D. Zheng, Q. Gan, Y. Gai, Z. Ye, M. Li, J. Zhou, Q. Huang, C. Ma, Z. Huang, Q. Guo, H. Zhang, H. Lin, J. Zhao, J. Li, A. J. Smola, and Z. Zhang, "Deep graph library: Towards efficient and scalable deep learning on graphs," ICLR Workshop on Representation Learning on Graphs and Manifolds, 2019.

[48] L. Huang, D. Ma, S. Li, X. Zhang, and H. Wang, "Text level graph neural network for text classification," in Proceedings of EMNLP-IJCNLP, 2019, pp. 3444–3450.

[49] Y. Wang, M. Huang, and L. Zhao, "Attention-based lstm for aspect-level sentiment classification," in Proceedings of EMNLP, 2016, pp. 606–615.

[50] B. Huang and K. M. Carley, "Parameterized convolutional neural networks for aspect level sentiment classification," in Proceedings of EMNLP, 2018, pp. 1091–1096.

[51] ——, "Syntax-aware aspect level sentiment classification with graph attention networks," in Proceedings of EMNLP, 2019, pp. 5468–5476.

[52] C. Zhang, Q. Li, and D. Song, "Aspect-based sentiment classification with aspect-specific graph convolutional networks," in Proceedings of EMNLP-IJCNLP, 2019, pp. 4568–4578.

[53] D. Chen and C. Manning, "A fast and accurate dependency parser using neural networks," in Proceedings of EMNLP, 2014, pp. 740–750.

[54] S. Kiritchenko, X. Zhu, C. Cherry, and S. Mohammad, "NRC-canada-2014: Detecting aspects and sentiment in customer reviews," in Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014, pp. 437–442.

[55] D. Tang, B. Qin, X. Feng, and T. Liu, "Effective lstms for target-dependent sentiment classification," in Proceedings of COLING, 2016, pp. 3298–3307.

[56] D. Tang, B. Qin, and T. Liu, "Aspect level sentiment classification with deep memory network," 2016.

[57] D. Ma, S. Li, X. Zhang, and H. Wang, "Interactive attention networks for aspect-level sentiment classification," in Proceedings of IJ, 2017, pp. 4068–4074.

[58] B. Huang, Y. Ou, and K. M. Carley, "Aspect level sentiment classification with attention-over-attention neural networks," in Proceedings of SBP-BRiMS, 2018, pp. 197–206.

[59] Z. Lu, P. Du, and J.-Y. Nie, "Vgcn-bert: Augmenting bert with graph embedding for text classification," in Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part I, ser. Lecture Notes in Computer Science, vol. 12035. Springer, 2020, pp. 369–382.