

每日小结

	周一	周二	周三	周四	周五
早	阅读文献	阅读文献	阅读文献	GraphTrans 代码	GraphTrans 代码
中	阅读文献	Response 撰写	阅读文献	阅读文献	GraphTrans 代码
晚	阅读文献		阅读文献	GraphTrans 代码	阅读文献

注：简单表述当前时间段工作，如看文献1，整理数据等

科研详情

文献阅读

文献 1

题目：GRAPH-BERT: Only Attention is Needed for Learning Graph Representations

作者：Jiawei Zhang, Haopeng Zhang, Congying Xia, Li Sun

出处：Nips2020

方法：

GNN 过度依赖图上的连接，此外由于大图非常耗显存，可能无法直接进行操作。该论文提出了一种新的只依赖 attention 机制的图网络 Graph-BERT，Graph-BERT 的输入不是全图，而是一堆采样得到的子图（不带边）。Graph-Bert 将原始图采样为多个子图，并且只利用 attention 机制在子图上进行表征学习，形成基于图的预训练模型。模型结构如下图所示。

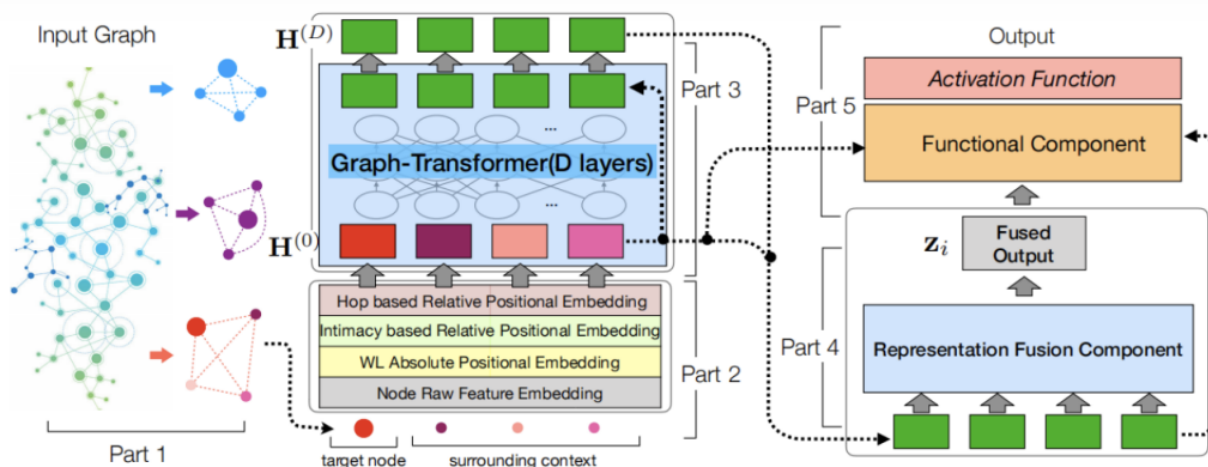


Figure 1: Architecture of the GRAPH-BERT Model. (Part 1: linkless subgraph batching; Part 2: node input vector embeddings; Part 3: graph transformer based encoder; Part 4: representation fusion; Part 5: functional component. Depending on the target application task, the function component will generate different output. In the sampled subgraphs, it covers both the target node and the surrounding context nodes.)

和之前 NLP 中的 BERT 不一样的地方主要是 position encoding，Graph-BERT 使用了三种 PE，分别是 WL absolute PE，intimacy based relative PE 和 Hop based relative PE，这里三个 PE 都是根据 complete graph 计算得到的。

为了便于操作，作者将 subgraph 根据图亲密度矩阵进行排序 $[i, j, \dots, m]$ ，其中 $S(i, j) > S(i, m)$ ，得到序列化的结果。

其中 Weisfeiler-Lehman Absolute Role Embedding 如下：经过 WL 之后，子结构一样的节点就会得到相同的 hash code，如果是 1，WL 有点像 degree centrality（针对无向图而言）。因此，WL PE 可以捕获全局节点角色信息。

Intimacy based Relative Positional Embedding 这个 PE 捕获的是偏 local 的信息，因为输入已经根据图亲密度矩阵进行排序过了，这里只需要简单地设 $P(v_i) = 0$ ，越接近 i 的节点 $P(v_i)$ 会越小。

Hop based Relative Distance Embedding 该 PE 捕获的是介于 global 和 local 之间的信

息：将节点 embedding 和这些 PE 加起来，然后输入到 Transformer 中得到 subgraph 每个节点的表征 $\mathbf{H}^{(l)}$ 。因为 subgraph 中有很多节点，作者希望最后只得到 target node 的表征 \mathbf{z}_i ，因此作者还设计了一个 Fusion function，原文中是把所有节点表征做一下 average。 $\mathbf{H}^{(l)}, \mathbf{z}_i$ 都会被输出，根据下游任务选择所需的进行使用。

启发：

半监督的工作将 Graph-Bert 原始图采样为多个子图，也可以考虑这种方式，因为我的 Graph 也存在比较大的问题

文献2

题目：GraphiT: Encoding Graph Structure in Transformers

作者：Grégoire Mialon, Dexiong Chen, Margot Selosse, Julien Mairal

出处：arXiv 2021

方法：

该工作表明，将结构和位置信息合并到 transformer 中，能够优于现有的经典 GNN。

GraphiT:

(1) 利用基于图上的核函数的相对位置编码来影响 attention scores,

(2) 并编码出 local sub-structures 进行利用。

之前 GT 发现 self-attention 在只关注 neighboring nodes 的时候会取得比较好的效果，但是在关注到所有节点的时候，性能就不行。这篇论文发现 transformer with global communication 同样可以达到不错的效果。因此，GraphiT 通过一些策略将 local graph structure 编码进模型中，(1) 基于正定核的注意力得分加权的相对位置编码策略 (2) 通过利用 graph convolution kernel networks (GCKN) 将 small sub-structure (e.g., paths 或者 subtree patterns) 编码出来作为 transformer 的输入。

实验结果：

Table 1: Average mean classification accuracy/mean absolute error.

Method / Dataset	MUTAG	PROTEINS	PTC	NCI1	ZINC (no edge feat.)
Size	188	1113	344	4110	12k
Classes	2	2	2	2	Reg.
Max. number of nodes	28	620	109	111	37
MF [11]	81.5±11.0	71.9±5.2	57.3±6.9	80.6±2.5	0.387±0.019
GCN [18]	78.9±10.1	75.8±5.5	54.0±6.3	75.9±1.6	0.367±0.011
GAT [37]	80.3±8.5	74.8±4.1	55.0±6.0	76.8±2.1	0.384±0.007
GIN [40]	82.6±6.2	73.1±4.6	55.0±8.7	81.7±1.7	0.387±0.015
GCKN-subtree [7]	87.8±9.4	72.0±3.7	62.1±6.4	79.6±1.8	0.474±0.001
[12]	79.3±11.6	65.8±3.1	58.4±8.2	78.9±1.1	0.359±0.014
[12] + LapPE	83.9±6.5	70.1±3.2	57.7±3.1	80.0±1.9	0.323±0.013
Transformers (T)	82.2±6.3	75.6±4.9	58.1±10.5	70.0±4.5	0.696±0.007
T + LapPE	85.8±5.9	74.6±2.7	55.6±5.0	74.6±1.9	0.507±0.003
T + Adj PE	87.2±9.8	72.4±4.9	59.9±5.9	79.7±2.0	0.243±0.005
T + 2-step RW kernel	85.3±6.9	72.8±4.5	62.0±9.4	78.0±1.5	0.243±0.010
T + 3-step RW kernel	83.3±6.3	76.2±4.4	61.0±6.2	77.6±3.6	0.244±0.011
T + Diffusion kernel	82.7±7.6	74.6±4.2	59.1±7.4	78.9±1.6	0.255±0.010
T + GCKN	84.4±7.8	69.5±3.8	61.5±5.8	78.1±5.1	0.274±0.011
T + GCKN + 2-step RW kernel	90.4±5.8	72.5±4.6	58.4±7.6	81.0±1.8	0.213±0.016
T + GCKN + Adj PE	90.5±7.0	71.1±6.9	57.9±4.2	81.4±2.2	0.211±0.010

启发：

想在 graph transformer 中保留时序的信息，目前还在找文献。

文献3

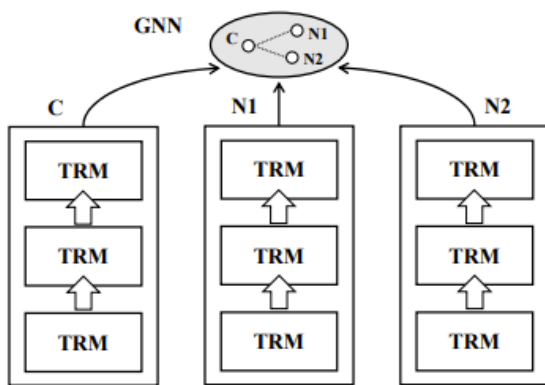
题目：GraphFormers:GNN-nested Transformers forRepresentation Learning on Textual Graph

作者：Junhan Yang, Zheng Liu, Shitao Xiao, Chaozhuo Li, Defu Lian, Sanjay Agrawal, Amit Singh, Guangzhong Sun, Xing Xie

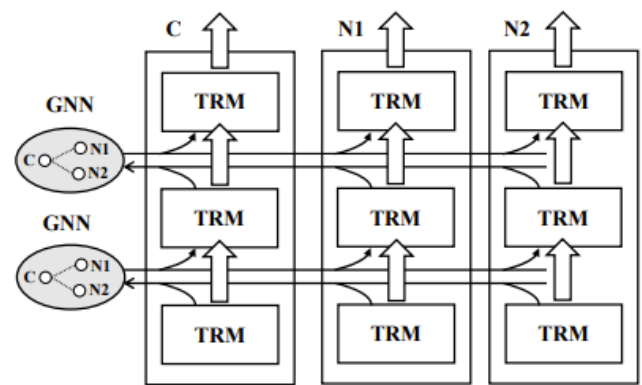
出处：

方法：预训练语言模型（PLM: Pretrained Language Model）与图神经网络（GNN: Graph Neural Network）是生成高质量的文本图表示的基本模块。

目前主流的研究通常借助级联的架构（Cascaded Framework，如图 1.A）来整合 PLM 与 GNN，即先由 PLM（e.g., BERT）生成各个节点的文本表示，再借助 GNN（e.g., GraphSage）来融合中心节点与邻域的语义。作者认为级联架构是一种相对低效的整合模式：在 PLM 的编码阶段，各个节点无法有效参照邻域节点的语义信息，进而对文本表示的质量产生不利的影响。



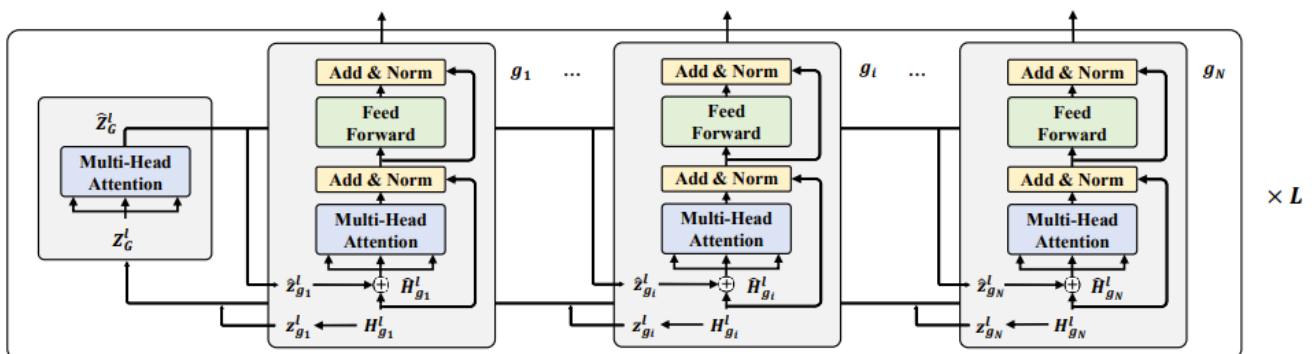
(A) Cascaded Transformers-GNN



(B) GNN-nested Transformers

在本文中，提出了一种 GNN 与 PLM 深度融合的网络架构（GNN-nested Transformers，如图 1.B），并将其命名为 GraphFormers。

GraphFormers 采取了层级化的 PLM-GNN 整合方式（如图）：在每一层中，每个节点先由各自的 Transformer Block 进行独立的语义编码，编码结果汇总为该层的特征向量（默认由 CLS 所关联的 hidden state 来表征）；各节点的特征向量汇集到该层的 GNN 模块进行信息整合；信息整合的结果被编码至对应各个节点的图增广（graph augmented）特征向量中，并分发至各个节点；各节点依照图增广特征向量进行下一层级的编码。



相较于此前的级联架构，GraphFormers 在 PLM 编码阶段便充分参照了邻域信息，从而大大提升了各节点文本表示的质量。

同时，考虑到节点间的信息交互是借由特征向量在极其轻量 GNN 模块中进行，每层整体的运算开销与单纯利用 Transformer Block 进行各节点独立的编码相差无几。

启发：

文本图表示的相关工作也可以借鉴思路：先由 PLM（e.g., BERT）生成各个节点的文本表示（我的工作中是 RS 图像信息），再借助 GNN（e.g., GraphSage）来融合中心节点与邻域的语义

工作进展

1: 阅读文献；

2: NIPS 2021 代码跑通：Graph Transformer (GraphTrans)

想在 graph transformer 中保留时序的信息，目前还在找类似模型。

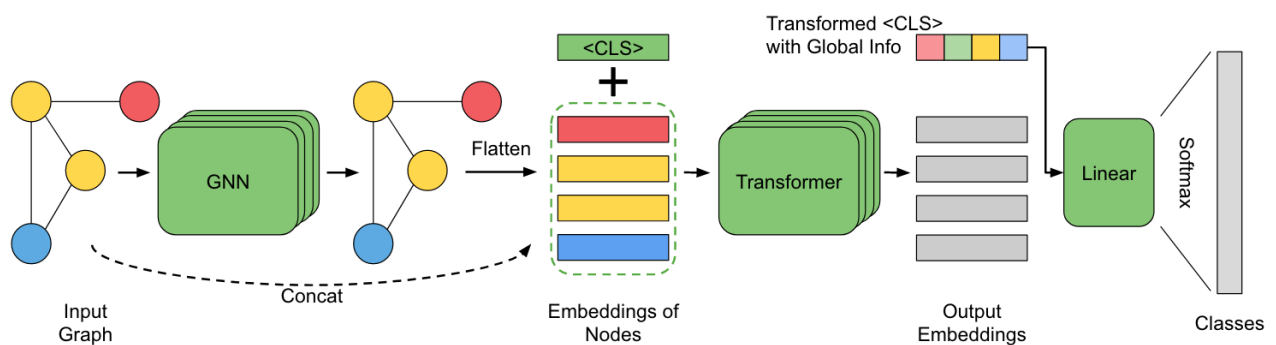


Figure 1: Architecture of GraphTrans. A standard GNN submodule learns local, short-range structure, then a global Transformer submodule learns global, long-range relationships.

3.论文 response 撰写等

4.准备组会，学习大量 Graph Transformer 的模型

下周计划

1. 投递论文

2. Graph transformer 代码尝试修改套用