基于图神经网络的文本分类

讲师：Houye

目录

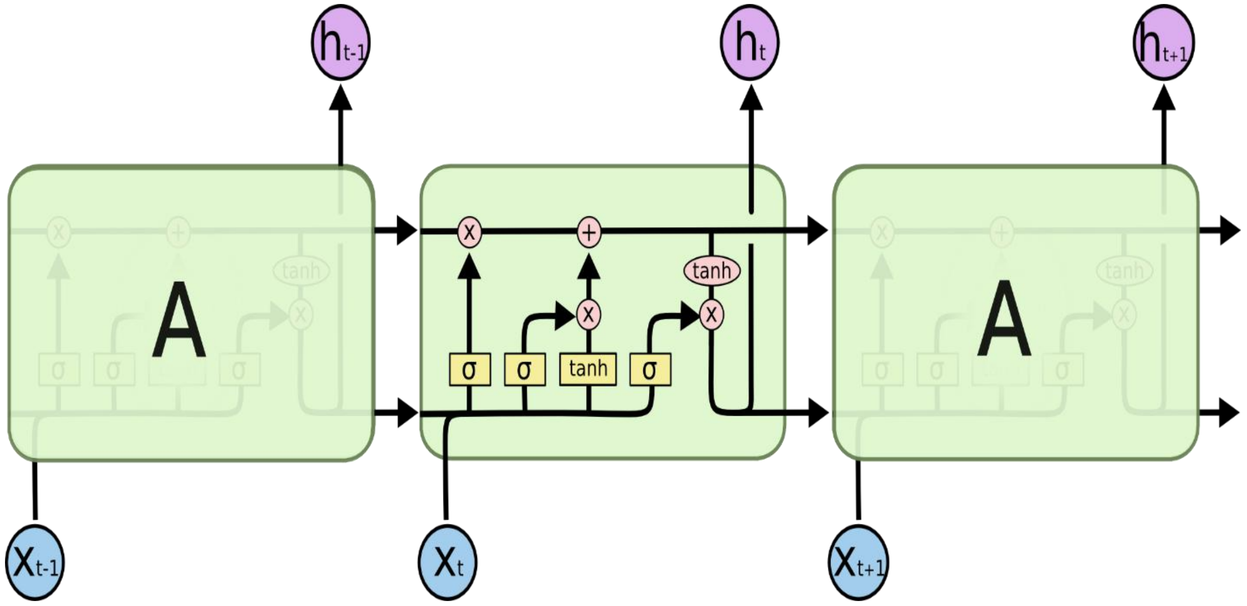
CONTENTS

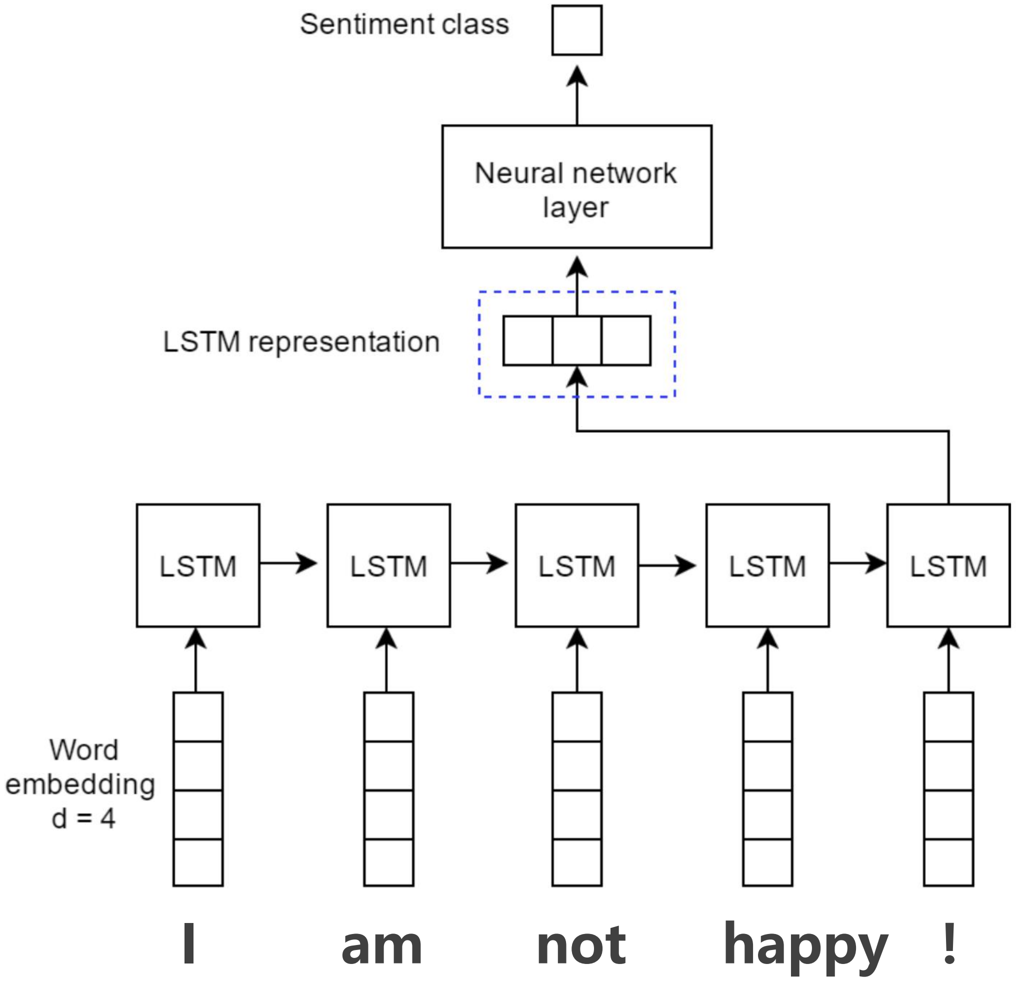
01 传统文本分类(文本序列)

传统文本分类算法

传统文本分类算法

- 以序列形式来描述文本。单词的**顺序**非常重要.
- I am **not** happy -> 不高兴
- 通常以循环神经网络RNN来建模



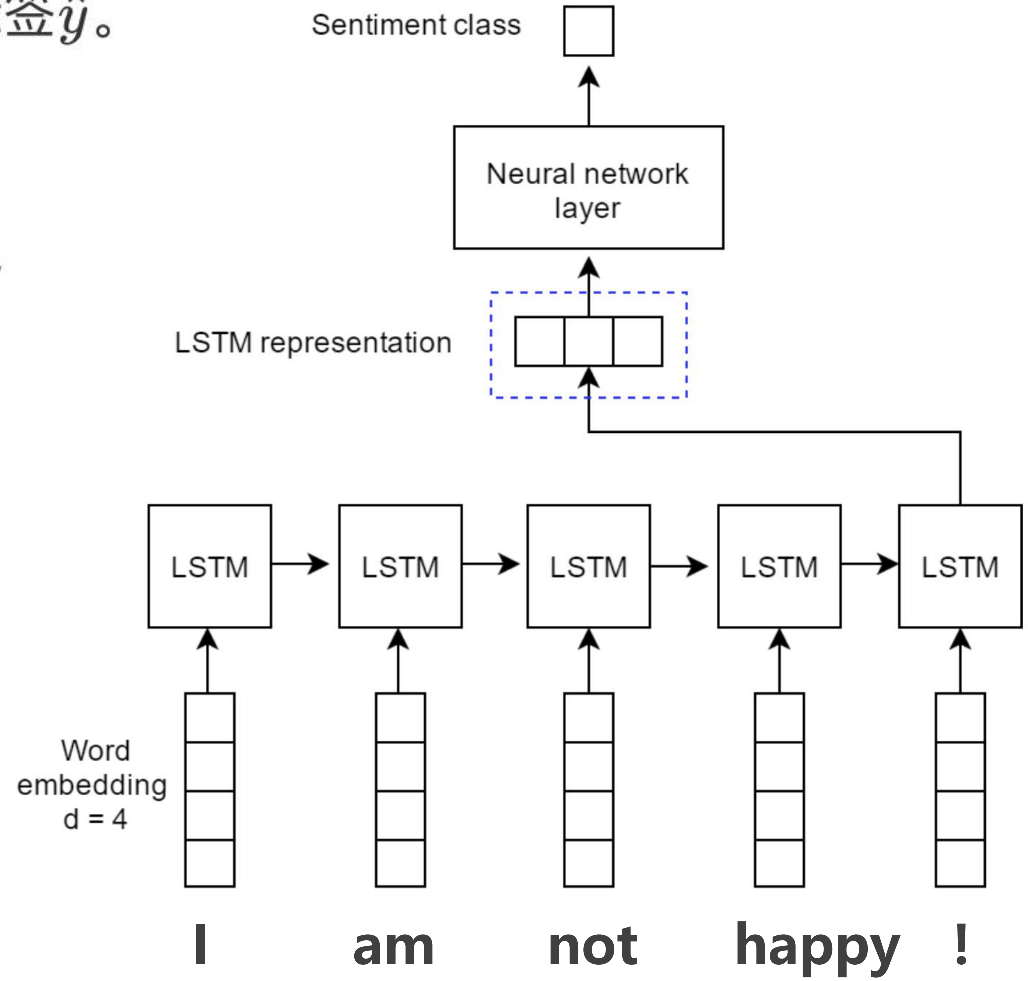


传统文本分类算法

基于LSTM的文本分类：单词序列$w_1, w_2, ..., w_n$ -> 文本标签$\hat{y}$。

- 将各个单词(表示)$E_{[w_1]}, ..., E_{[w_n]}$作为LSTM输入
- 用LSTM按顺序逐个编码单词，得到整个句子的表示Sen
- 用MLP将句子表示Sen映射到标签$\hat{y}$
- 计算预测标签$\hat{y}$与真实标签y的loss，优化模型

$$Sen = \mathrm{LSTM}\left(E_{[w_1]}, ..., E_{[w_n]}\right)$$
$$\hat{y} = \mathrm{softmax}(\mathrm{MLP}(Sen))$$

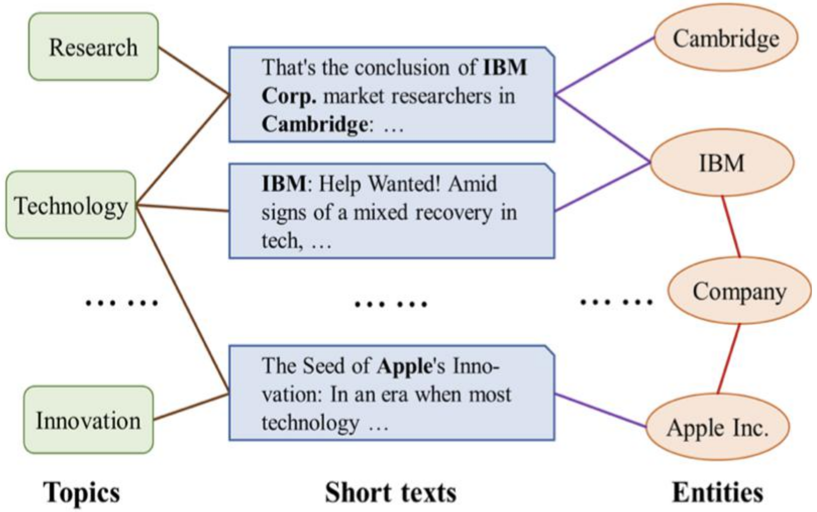


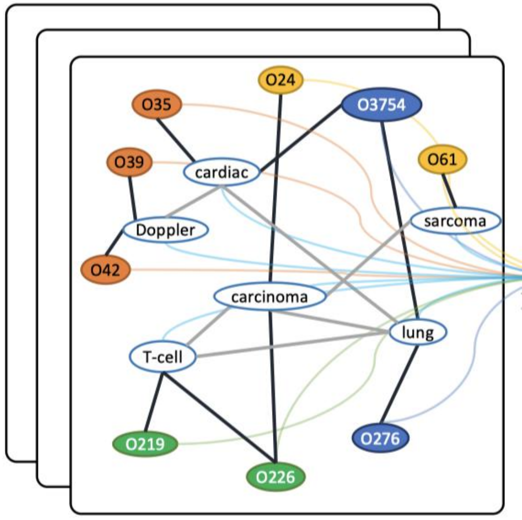
02 图神经网络文本分类算法(文本图)

图神经网络文本分类算法(文本图)

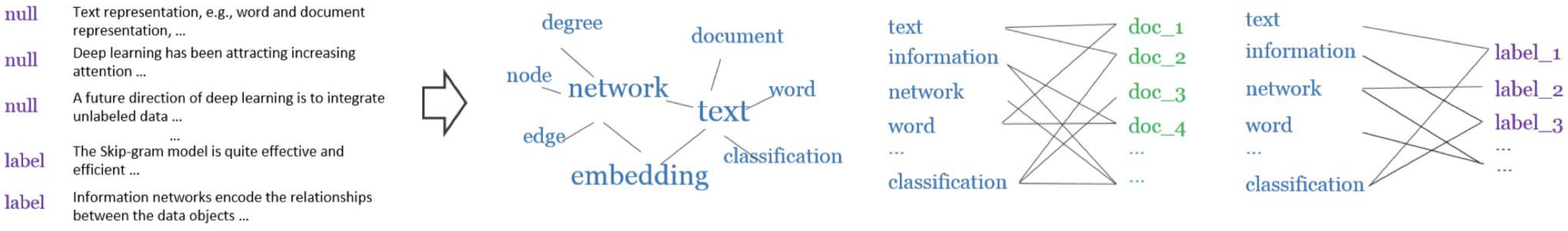
文本数据 -> 图结构的文本数据
文本分类算法 -> 图算法
经典论文：

- 15KDD Predictive **Text** Embedding through Large-scale Heterogeneous Text **Networks**
- 19AAAI **Graph** Convolutional Networks for **Text Classification**





Word Document Graph

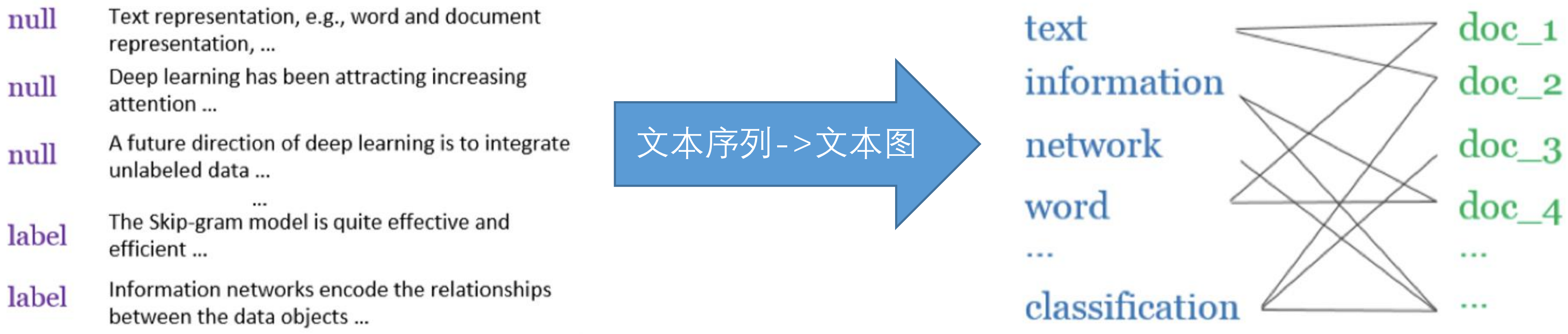


(a) word-word network (b) word-document network (c) word-label network

图神经网络文本分类算法(文本图)

基于图神经网络的文本分类 2大步骤

- 如何将文本数据转换为图数据?
- 如何设计相应的图神经网络?



图神经网络文本分类算法(文本图)

异步社区
人 民 邮 电 出 版 社

如何将文本数据转换为图数据?

方式非常灵活，如：

- 基于共现：共同出现的两个单词i和j，就认为他们之间有一条边$A_{ij}=1$
- 基于相似度：计算两个单词表示的相似度 $sim(i,j)$ 。如果$sim(i,j)>\delta$，则$A_{ij}=1$

序列 -> 图的转换：丢失了顺序信息。序列是有先后的，节点的邻居没有先后的区别。

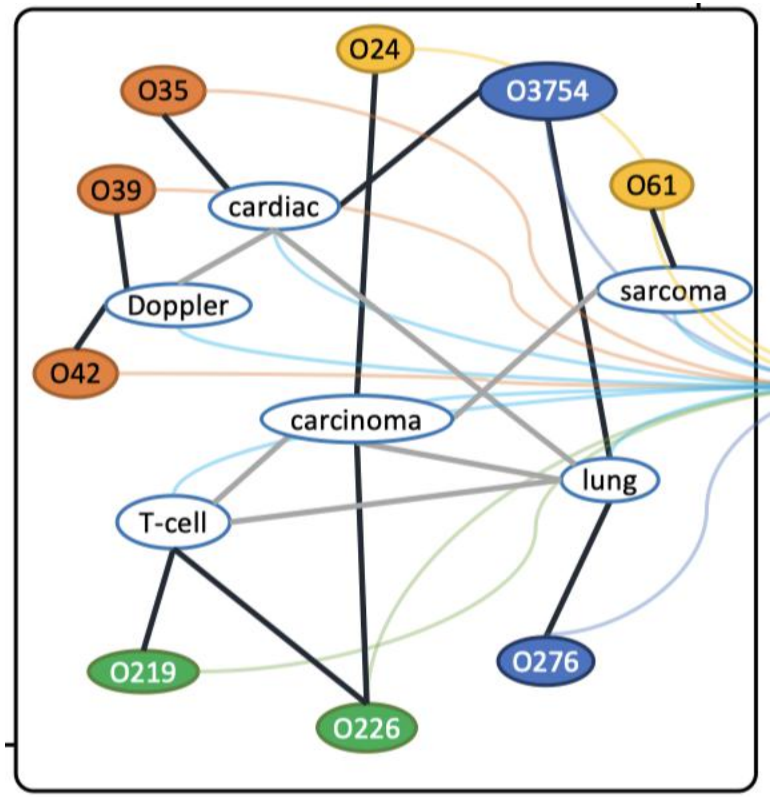
论文19AAAI TextGCN Graph Convolutional Networks for Text Classification是如何构图的?

$$A_{ij}=\begin{cases}\mathrm{PMI}(i,j) & i,j \text{ are words, } \mathrm{PMI}(i,j)>0\\ \text{TF-IDF}_{ij} & i \text{ is document, } j \text{ is word}\\ 1 & i=j\\ 0 & \text{otherwise}\end{cases}$$

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

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

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



Word Document Graph

图神经网络文本分类算法(文本图)

回忆GCN

- 属性信息$X \in \mathbb{R}^{N\times d}$，一些属性(特征)作为节点初始表示
- 结构信息$A \in \mathbb{R}^{N\times N}$，聚合邻居信息来更新节点表示

$$Z = AXW$$

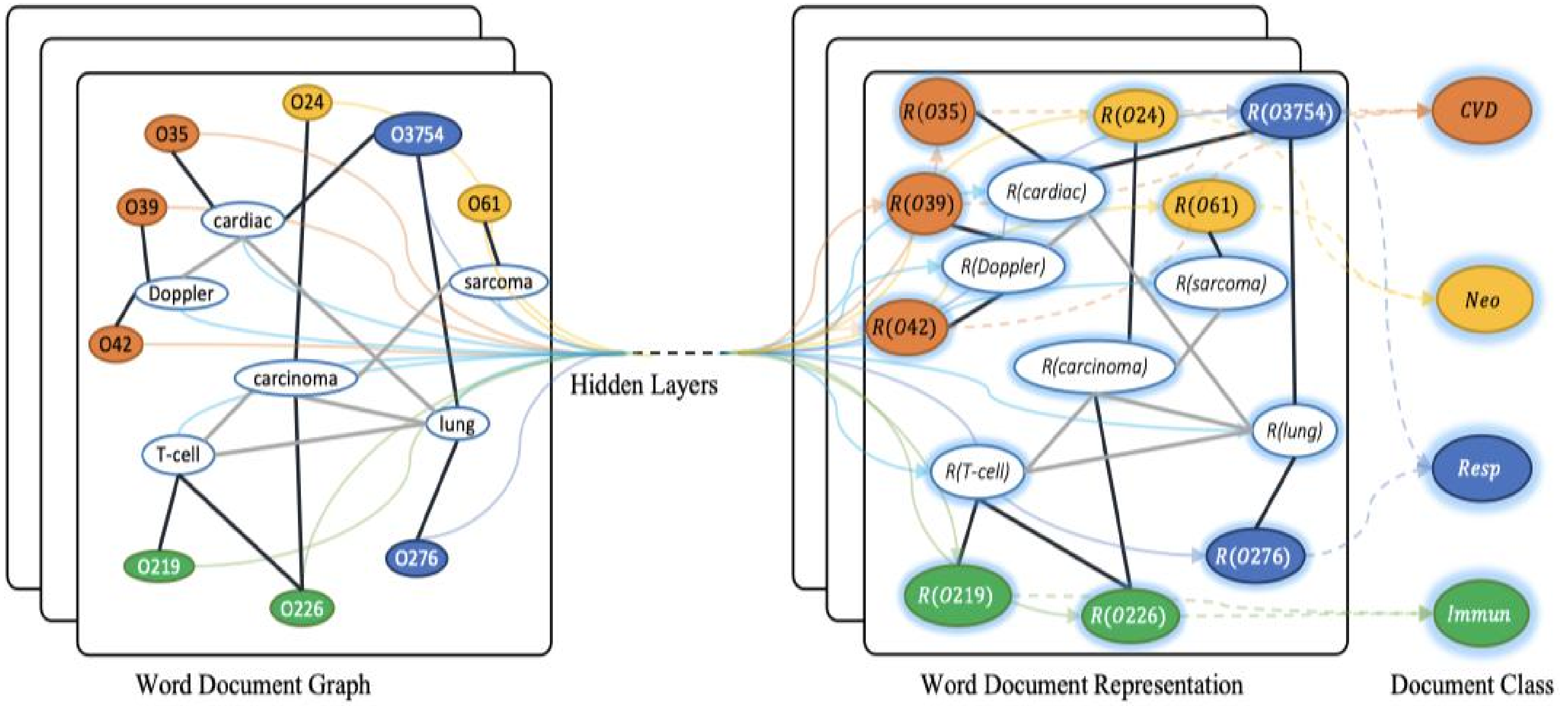
构图之后(得到了A)，如何构建节点的属性(X)呢?

- ID表示。$X = I$ 也是19AAAI TextGCN Graph Convolutional Networks for Text Classification的做法。
- 特征表示。抽取一些节点特征。例如文档的TF-IDF
- 预训练Embedding作为特征。例如，单词节点的特征为其word embedding

图神经网络文本分类算法(文本图)

如何设计相应的图神经网络?

- 同质图神经网络(GCN)
- 更加复杂的图神经网络(异质图神经网络 Heterogeneous Graph Attention Networks for Semi-supervised Short Text Classification)



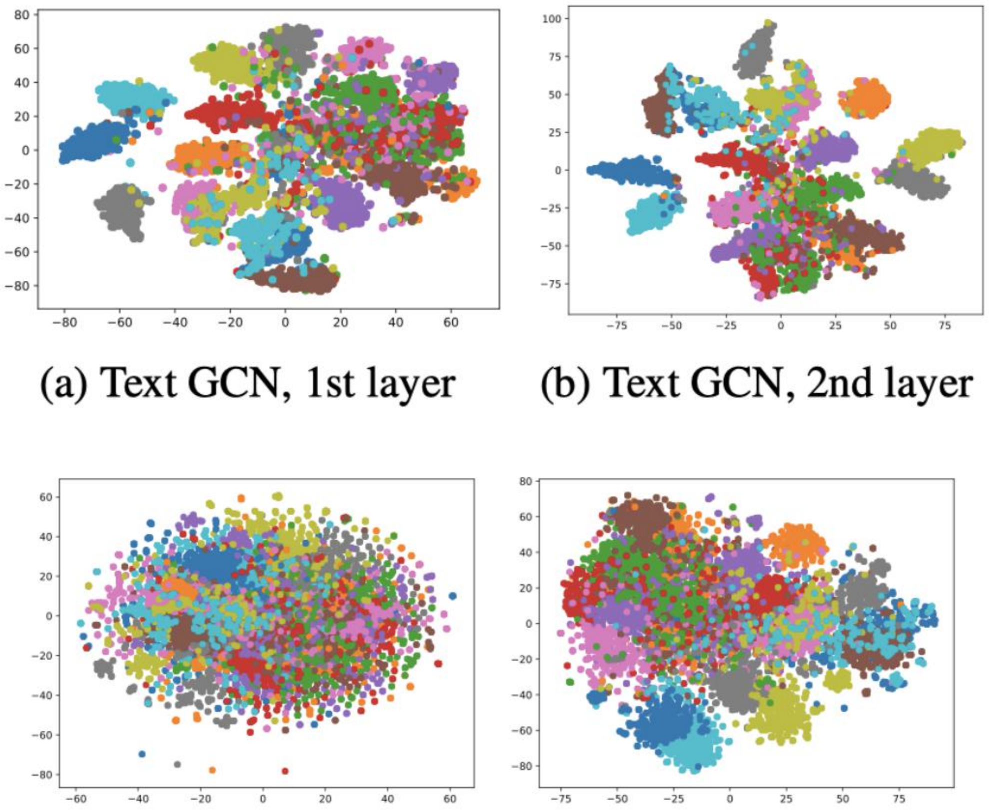
$$Z = \mathrm{softmax}(\tilde{A}\,\mathrm{ReLU}(\tilde{A}XW_0)W_1)$$

$$\mathcal{L} = -\sum_{d\in\mathcal{Y}_D}\sum_{f=1}^{F} Y_{df}\ln Z_{df}$$

图神经网络文本分类算法(文本图)

19AAAI TextGCN Graph Convolutional Networks for Text Classification的效果

Model	20NG	R8	R52	Ohsumed
TF-IDF + LR	0.8319 ± 0.0000	0.9374 ± 0.0000	0.8695 ± 0.0000	0.5466 ± 0.0000
CNN-rand	0.7693 ± 0.0061	0.9402 ± 0.0057	0.8537 ± 0.0047	0.4387 ± 0.0100
CNN-non-static	0.8215 ± 0.0052	0.9571 ± 0.0052	0.8759 ± 0.0048	0.5844 ± 0.0106
LSTM	0.6571 ± 0.0152	0.9368 ± 0.0082	0.8554 ± 0.0113	0.4113 ± 0.0117
LSTM (pretrain)	0.7543 ± 0.0172	0.9609 ± 0.0019	0.9048 ± 0.0086	0.5110 ± 0.0150
Bi-LSTM	0.7318 ± 0.0185	0.9631 ± 0.0033	0.9054 ± 0.0091	0.4927 ± 0.0107
PV-DBOW	0.7436 ± 0.0018	0.8587 ± 0.0010	0.7829 ± 0.0011	0.4665 ± 0.0019
PV-DM	0.5114 ± 0.0022	0.5207 ± 0.0004	0.4492 ± 0.0005	0.2950 ± 0.0007
PTE	0.7674 ± 0.0029	0.9669 ± 0.0013	0.9071 ± 0.0014	0.5358 ± 0.0029
fastText	0.7938 ± 0.0030	0.9613 ± 0.0021	0.9281 ± 0.0009	0.5770 ± 0.0049
fastText (bigrams)	0.7967 ± 0.0029	0.9474 ± 0.0011	0.9099 ± 0.0005	0.5569 ± 0.0039
SWEM	0.8516 ± 0.0029	0.9532 ± 0.0026	0.9294 ± 0.0024	0.6312 ± 0.0055
LEAM	0.8191 ± 0.0024	0.9331 ± 0.0024	0.9184 ± 0.0023	0.5858 ± 0.0079
Graph-CNN-C	0.8142 ± 0.0032	0.9699 ± 0.0012	0.9275 ± 0.0022	0.6386 ± 0.0053
Graph-CNN-S	–	0.9680 ± 0.0020	0.9274 ± 0.0024	0.6282 ± 0.0037
Graph-CNN-F	–	0.9689 ± 0.0006	0.9320 ± 0.0004	0.6304 ± 0.0077
Text GCN	**0.8634 ± 0.0009**	**0.9707 ± 0.0010**	**0.9356 ± 0.0018**	**0.6836 ± 0.0056**



(a) Text GCN, 1st layer (b) Text GCN, 2nd layer

(c) PV-DBOW (d) PTE

Figure 5: The t-SNE visualization of test set document embeddings in 20NG.