Label Incorporated Graph Neural Networks for Text Classification

Yuan Xin[†], Linli Xu[†§], Junliang Guo[†], Jiquan Li[†], Xin Sheng[†], Yuanyuan Zhou[‡]
[†]Anhui Province Key Laboratory of Big Data Analysis and Application,
School of Computer Science and Technology, University of Science and Technology of China
[‡]Information Science Center, University of Science and Technology of China
[§]IFLYTEK Co., Ltd.
[†]lwjxy@mail.ustc.edu.cn, linlixu@ustc.edu.cn, {guojunll,lijiquan,xins}@mail.ustc.edu.cn
[‡]yoyo@ustc.edu.cn

Introduction

Background

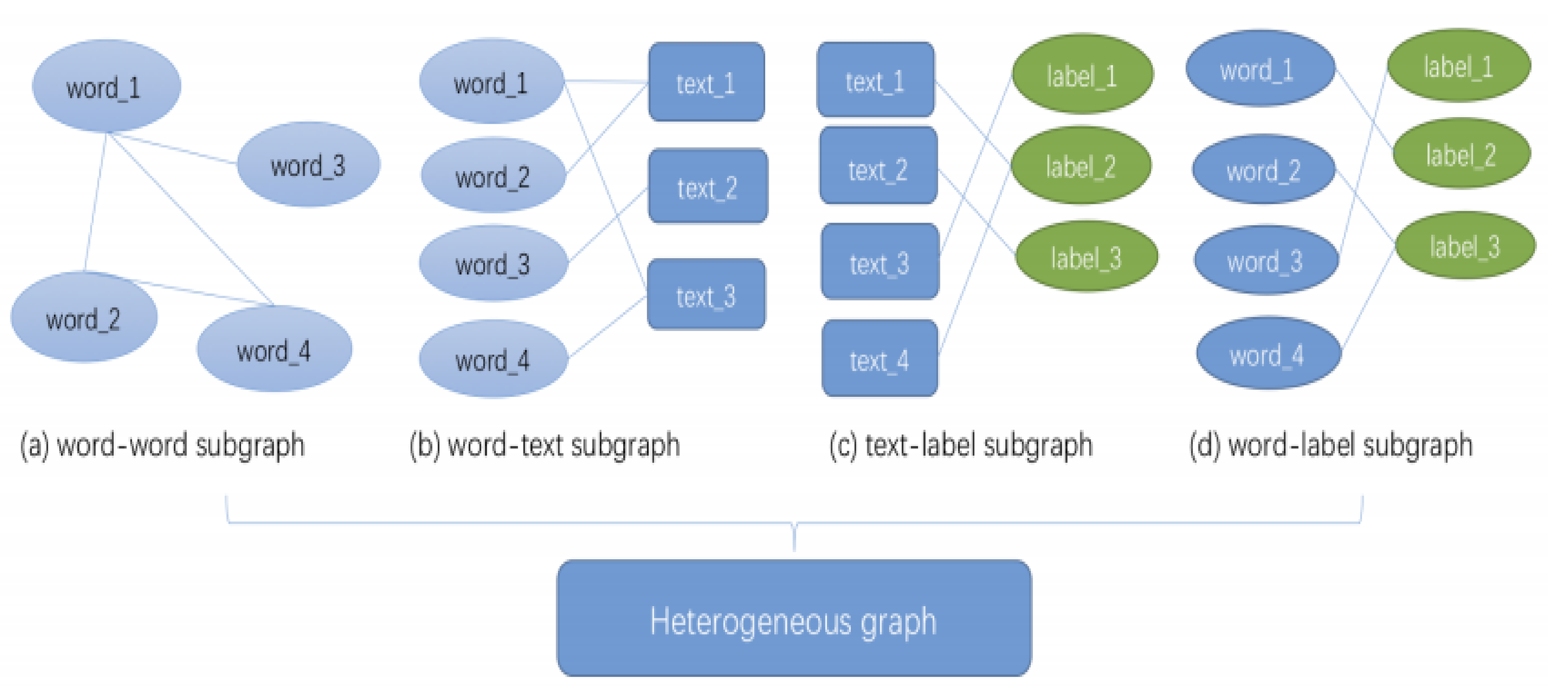
- **What is Text Classification?**
  A task to annotate a given text sequence with one or multiple class labels.
- **Why Estimate Graph Convolutioanl Network ?**
  A Graph Convolution Network (GCN) is a multi-layer neural network that directly operates on graph structures and learns node embeddings based on their neighbors
- **The difference between existing studies and our work.**
  Previous works only consider the text information while building the graph, heterogeneous information such as labels is ignored

Challenges

- **How to incorporate label information?**
  Directly add label nodes when bulding the graph

Label-incorporated GCN

Framework



Preliminaries

We build a label-incorporated graph on the whole corpus and feed the graph into a two-layer GCN model. There are three kinds of nodes which includes text nodes, word nodes and label nodes and we add different kinds of connections on them. After we get node embeddings. We add softmax layer to the learned embeddings to make classification directly.

$$Z = \mathrm{softmax}(\widetilde{A} f(\widetilde{A} X W_0) W_1),$$

Loss function is:

$$\mathcal{L} = -(\sum_{d \in V_d} Y_d \ln Z_d + \lambda \sum_{l \in V_l} Y_l \ln Z_l),$$

The first part is the cross-entropy loss over all labeled texts and the second part is the loss over all labels, As labels are embedded in the same semantic space as the texts, we could also perform classification over labels to make sure the learned label embeddings are meaningful and explainable.

GNN

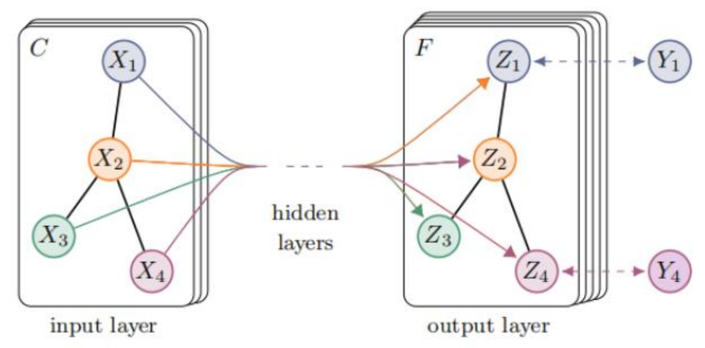
Graph Convolutional Networks

A Graph Convolution Network (GCN) is a multi-layer neural network that directly operates on graph structures and learns node embeddings based on their neighbors [17]. Consider a graph G = (V, E), where V and E are the sets of nodes and edges respectively. In our paper, V consists of three types of nodes including word nodes Vw, text nodes Vd and label nodes Vl . We consider weighted edges between nodes, and denote the adjacency matrix of the graph as A ∈ [0, 1]|V | × |V | , where Ai,j = 0 indicates that node i and node j are not connected, otherwise Ai,j indicates the weight of the edge between i and j.

We denote X ∈ R |V | ×m as the feature matrix constructed by features of all nodes in the graph, where m is the dimension of the feature. Denote Ae = D− 1 2 AD− 1 2 as the normalized symmetric adjacency matrix and D is diagonal degree matrix where Di,i = P j Ai,j , then the propagation rule of the multilayer GCN can be written as:

$$H^{(l+1)} = f(\widetilde{A} H^{(l)} W_l),$$

where H(l) ∈ R |V | ×d stacks the d-dimensional hidden vectors of all the nodes at the l-th layer, and H(0) = X. Wl ∈ R d×d is the trainable parameter matrix of layer l, and f(·) is the ReLU activation function.



Subgraph Module

Word-Word Subgraph: The Word-Word subgraph, denoted as Gww = (Vw, Eww), is a word-word co-occurrence graph where Vw represents the vocabulary of words in the whole corpus, Eww is the set of edges between words. We utilize the point-wise mutual information (PMI) value between word vi and vj as the edge weight wi,j , which can be computed as:

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

where

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

Word-Text Subgraph: The Word-Text subgraph, denoted as Gwd = (Vw ∪ Vd,Ewd) is a bipartite network where Vd is the set of texts and Vw is the word vocabulary. Ewd is the set of edges between words and texts. The weight between word vi and text dj is calculated as the term frequency-inverse document frequency (TF-IDF) score of word vi in text dj , i.e, Ai,j = TF-IDFvi,dj . This subgraph captures the word co-occurrences at the text level

Text-Label Subgraph: The Text-Label subgraph, denoted as Gdl = (Vd ∪ Vl , Edl) is a bipartite network where Vd is the set of texts and Vl is the set of labels in the corpus. Edl is the set of edges between texts and the corresponding labels. If text di is associated with the label lj , we connect an edge between them and the weight is a hyper-parameter to be tuned. Results are shown in experiment

Experiments

Table 2: Comparisons on the datasets.

Model	R8	R52	Ohsumed
TF-IDF + LR	93.74 ± 0.00	86.95 ± 0.00	54.66 ± 0.00
CNN-rand	94.02 ± 0.57	85.37 ± 0.47	43.87 ± 1.00
CNN-non-static	95.71 ± 00.52	87.59 ± 00.48	58.44 ± 1.06
LSTM	93.68 ± 0.82	85.54 ± 1.13	41.13 ± 1.17
LSTN(pretrain)	96.09 ± 0.19	90.48 ± 0.86	51.10 ± 1.50
Bi-LSTM	96.31 ± 0.33	90.54 ± 0.91	49.27 ± 1.07
PV-DBOW	85.87 ± 0.10	78.29 ± 0.11	46.65 ± 0.19
PV-DM	52.07 ± 0.04	44.92 ± 0.05	29.50 ± 0.07
fastText	86.04 ± 0.24	71.55 ± 0.42	14.59 ± 0.00
fastText(bigrams)	82.95 ± 0.03	68.19 ± 0.04	14.59 ± 0.00
SWEM	95.32 ± 0.26	92.94 ± 0.24	63.12 ± 0.55
LEAM	93.31 ± 0.24	91.84 ± 0.23	58.58 ± 0.79
Graph-CNN-C	96.99 ± 0.12	92.75 ± 0.22	63.86 ± 0.53
Graph-CNN-S	96.80 ± 0.20	92.74 ± 0.24	62.82 ± 0.37
Graph-CNN-F	96.89 ± 0.06	93.20 ± 0.04	63.04 ± 0.77
TextGCN	97.07 ± 0.10	93.56 ± 0.18	68.36 ± 0.56
our model(label-text connection)	$\mathbf{97.37 \pm 0.17}$	$\mathbf{94.15 \pm 0.10}$	$\mathbf{69.10 \pm 0.20}$
our model's variant(label-word connection)	$\mathbf{97.10 \pm 0.07}$	$\mathbf{93.91 \pm 0.11}$	$\mathbf{68.93 \pm 0.17}$

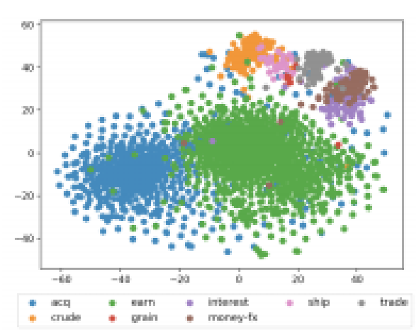


Fig. 3: 1st layer's embedding

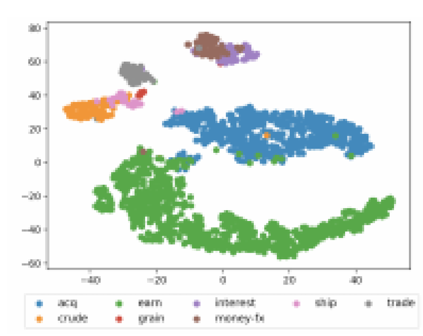


Fig. 4: 2nd layer's embedding

- Our complete approach Label-incorporated GNN achieves the best classification performances on three data size settings. Label-incorporated GNN outperforms other text classification algorithms with averages of about 1-2 percent accuracy improvements on three datasets.
- We could see that after two layers of graph convolution, our labelincorporated GCN could make different category's embedding clearly distinguished

Conclusion

- We build a novel heterogeneous graph convolutional network for text classification by adding label nodes to the graph. . To the best of our knowledge, this is the first work to incorporate labels in GCN when building the graph
- In the heterogeneous graph, both text and label embeddings are learned in the same semantic space, based on which an auxiliary classification loss function of the label embeddings.
- Experimental results demonstrate the superior performance of the proposed model over baselines on several text classification benchmark datasets. GME.

Future

- In the future, we will apply our models to other scenarios and applications leading to a more general framework.