BTech Project
Mid - Semester Examination
Network Representation Learning

Anket Kotkar
180101037
Supervisor: Prof. Sanasam Ranbir Singh

Outline

- Domain of research
- Literature survey
- Problems (limitations) of the existing literature
- Planned solution

Domain of Research

(Network Representation Learning)

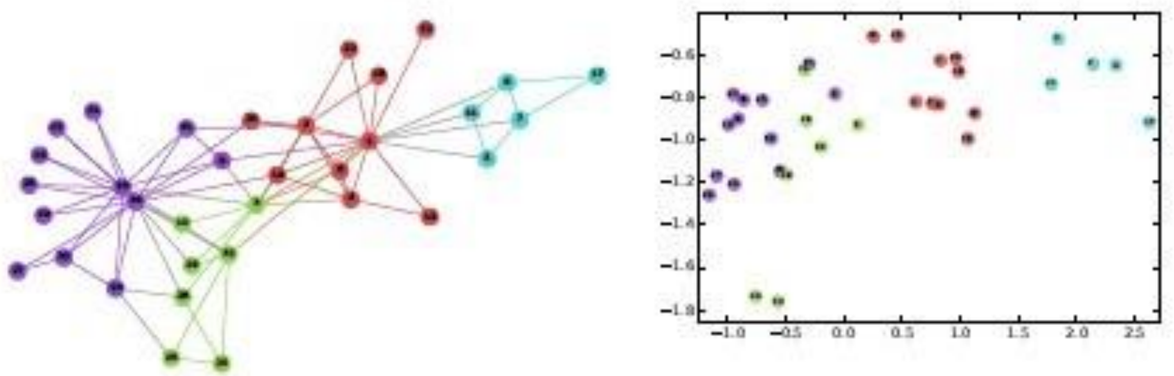
Introduction

- Complex relationships modeled as Information Network
- Emerging Domain based applications
- Perform network analysis for downstream applications
- Network representation learning for network analysis

Definition: Network Representation Learning

Given an information network $G = (V, E, X, Y)$, by integrating network structure in E, vertex attributes X, vertex labels Y (if available), the task of network representation learning is to learn a mapping function $f : v \rightarrow r_v \in R^d$, where r_v is the learned vector representation of the vertex v and d is the dimension of the learned representations.

Generated Sample Representation



(a) Input: Karate Graph (b) Output: Representation

Literature Survey

Network Representation Learning Surveys

- Networks have grown tremendously from thousands of node network to current million and billion node networks.
- This huge networks have fueled for constant improvements in NRL architectures from using mere special matrices to represent graphs to using transformers today for many tasks.
- Having numerous emerging applications and ever increasing network structures, NRL is a field having scope to explore many new horizons.

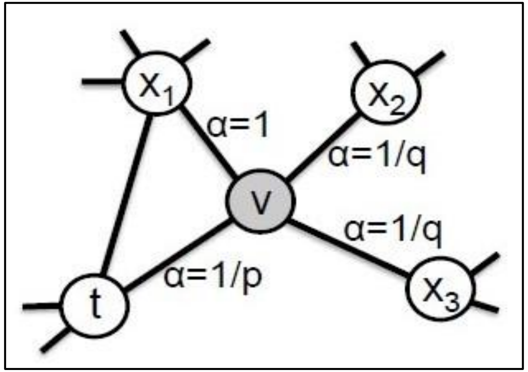
Survey - References

Network Representation Learning Paradigm Surveys:

1. P. Goyal and E. Ferrara, “Graph embedding techniques, applications, and performance: A survey,” CoRR, vol. abs/1705.02801, 2017
2. Zhang, D., Yin, J., Zhu, X., & Zhang, C. (2018). Network representation learning: A survey. IEEE transactions on Big Data

Random Walk Methods

- Using sequences of words to train the models has shown great success in NLP domain, e.g. **Skip-gram** or **CBOW** Model.
- Random Walk base methods proposes to adapt the methodology in graph context by generating truncated random sequences of nodes by traversing over the edges.
- These sequences are used to learn the node embeddings using NLP techniques.



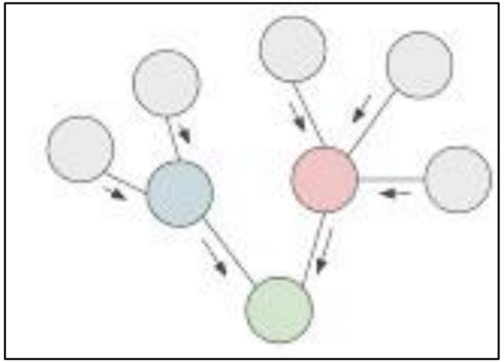
Random Walk Methods - References

Random Walk Based Method References:

1. B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in KDD, 2014, pp. 701–710
2. A. Grover and J. Leskovec, “Node2vec: Scalable feature learning for networks,” in KDD, 2016, pp. 855–864

GNN Methods

- Neural networks that can be directly applied to graph structured data are called Graph Neural Networks.
- Computing through several hidden layers, GNN determine the node embedding of each node, by looking at the information on its neighboring nodes.
- Intuition of GNN is that nodes are naturally defined by their neighbors and connections.



GNN Methods - References

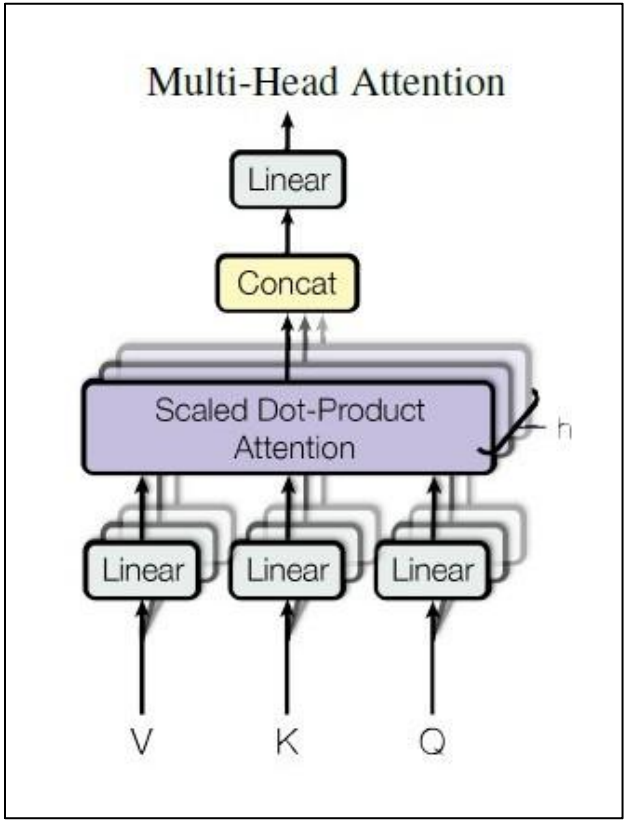
GNN based method reference:

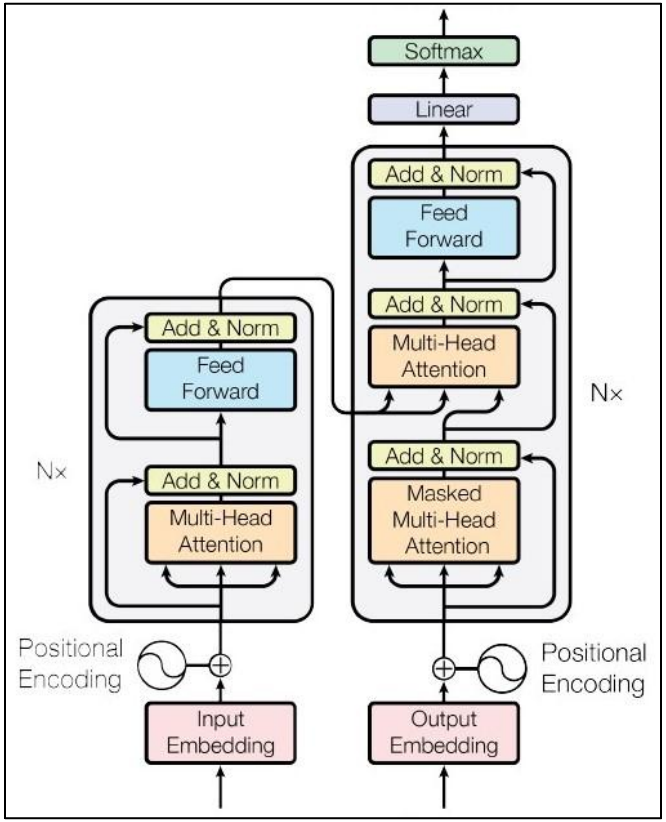
1. [Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016
2. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 1025–1035 (2017)

Transformer Methods

- Attention function computes a weighted sum of the values, where weights are computed by a compatibility function of key and query.
- Transformer uses attention mechanism along with positional encoding to generate inductive model.
- Extending transformer architecture to Graphs, raw feature embeddings and positional embeddings generated using graph structure are fed into transformer to learn node representation.

Transformer - Diagram





Transformer Methods - References

Transformer based models references:

1. Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. AAAI Workshop on Deep Learning on Graphs: Methods and Applications, 2021
2. Zhang, J.; Zhang, H.; Sun, L.; and Xia, C. 2020. Graph-Bert: Only Attention is Needed for Learning Graph Representations. arXiv preprint arXiv:2001.05140

Existing Literature Shortcomings

Shortcomings

- In general, whenever NRL is performed, only the network topology is considered. Along with structural data, nodes and edges have attributes too which is not considered when node or edge representations are learnt.
- How to incorporate that data in network learning is something which needs to be explored more.

Shortcomings

- There are a plethora of information available for Heterogeneous graphs but very few works systematically incorporates the meta-information.
- Heterogeneous graph Bert is the latest Transformer to have incorporated node and edge types.
- To incorporate something sequential like paths or metapaths or metagraphs - we need sequence based methods like GraphBERT or Transformers.

Planned Solution

Objective

To incorporate the node and edge attributes in the network representation learning task.

So what I mean by node attributes and where to start?

Node / Edge Attributes

- ❖ Node attributes contain node type, node creation timestamp. For network specific information, it may include textual data, image data, relations to other entities.
- ❖ Edge attributes include edge type and edge creation timestamp. Same kind of pair of nodes may have different type of edges between them. Some weight may also be associated with them by default.
- ❖ As GraphBERT is most recent architecture providing SoTA result, I choose to work with GraphBERT.

GraphBERT

- Using 4 type of pre processed embeddings (raw feature embedding, Weisfeiler-Lehman embedding, relative positional embedding and hop based embedding), it generates the node representations.
- Using residual connections, deep architecture have been developed.
- Contextualized node representation of target node are generated by fusion function at last layer.

GraphBERT Architecture

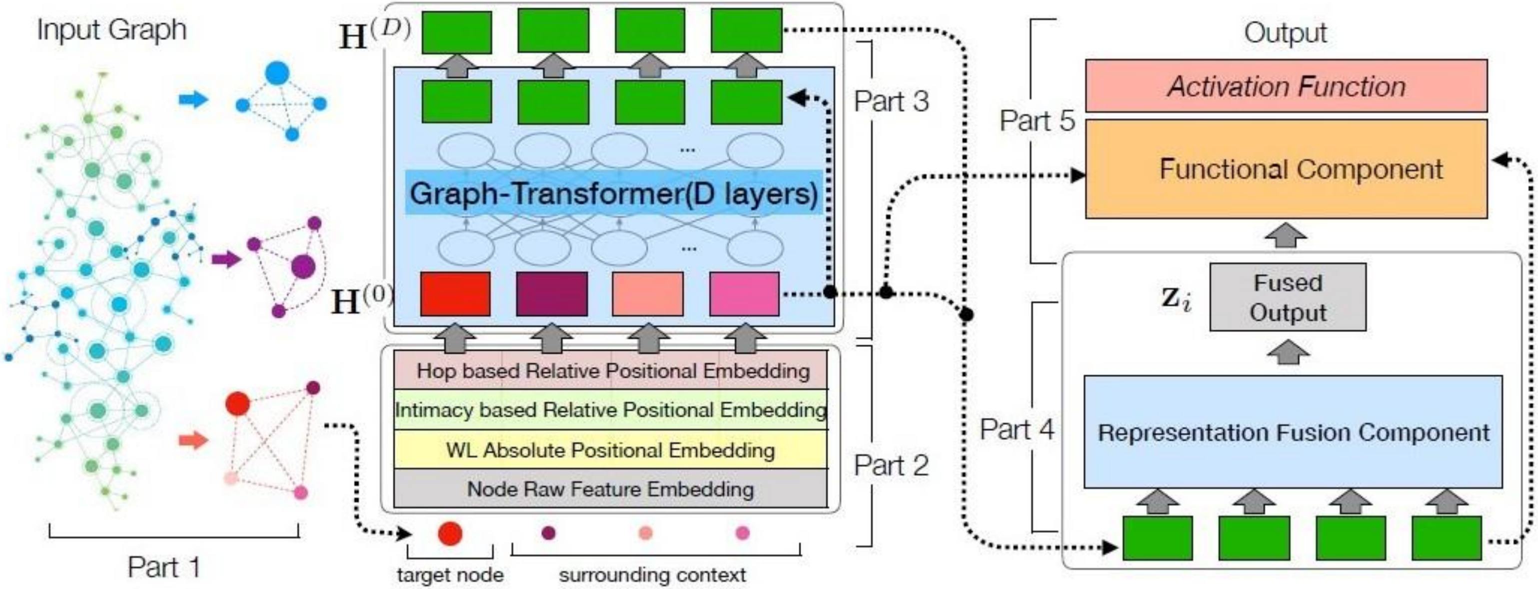


Figure 1: Architecture of the GRAPH-BERT Model. (Part 1: linkless subgraph batching; Part 2: node input vector embeddings; Part 3: graph transformer based encoder; Part 4: representation fusion; Part 5: functional component. Depending on the target application task, the function component will generate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

Proposed Plan

To realize the objective, I want to experiment the following changes:

1. Improving generated raw embeddings
2. Architectural innovations
3. Exploring pre-training strategies as a way to incorporate meta-information during training

Questions?

Thank you