Australian National University

Learning Attributed Subgraph Matching

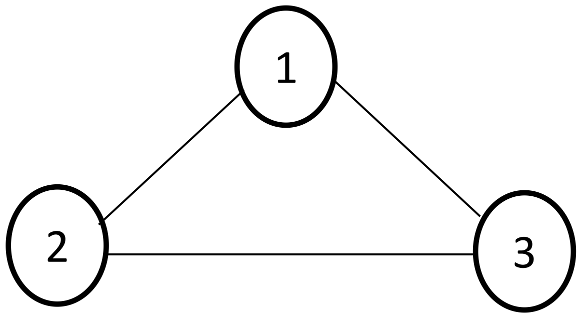
Yanxi Lu

Supervisor: Dr. Qing Wang

Outline

- Introduction and problem definition
- Existing methods and Motivation
- Methodology
- Experiments and Results
- Future Work

Graph



Nodes {1, 2, 3}

**Edges {(1, 2),
(2, 3),
(1, 3)}**

Enough information for real tasks?

Attributed graphs

Richer graph representations

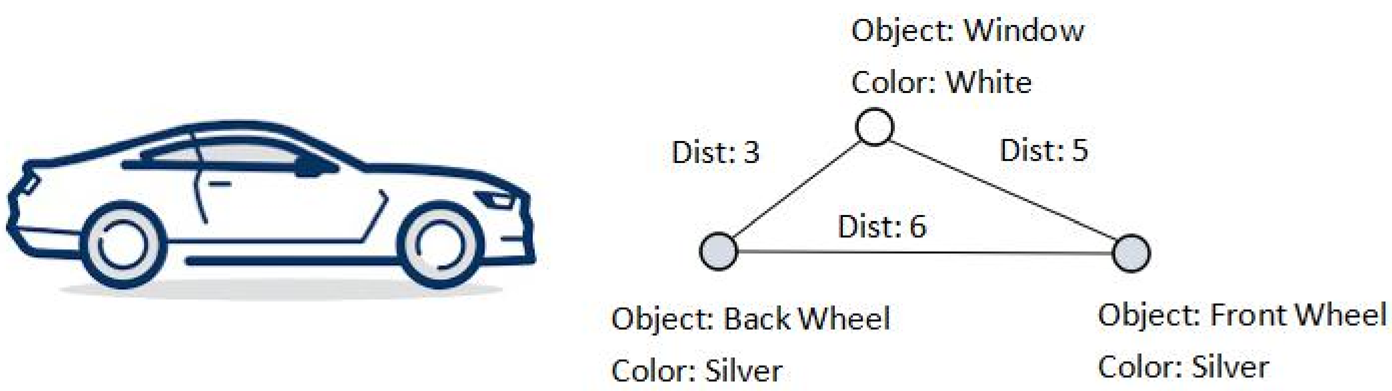


Figure 1: Example Attributted Graph

Application of Attributed Graph

- Pattern representation in images

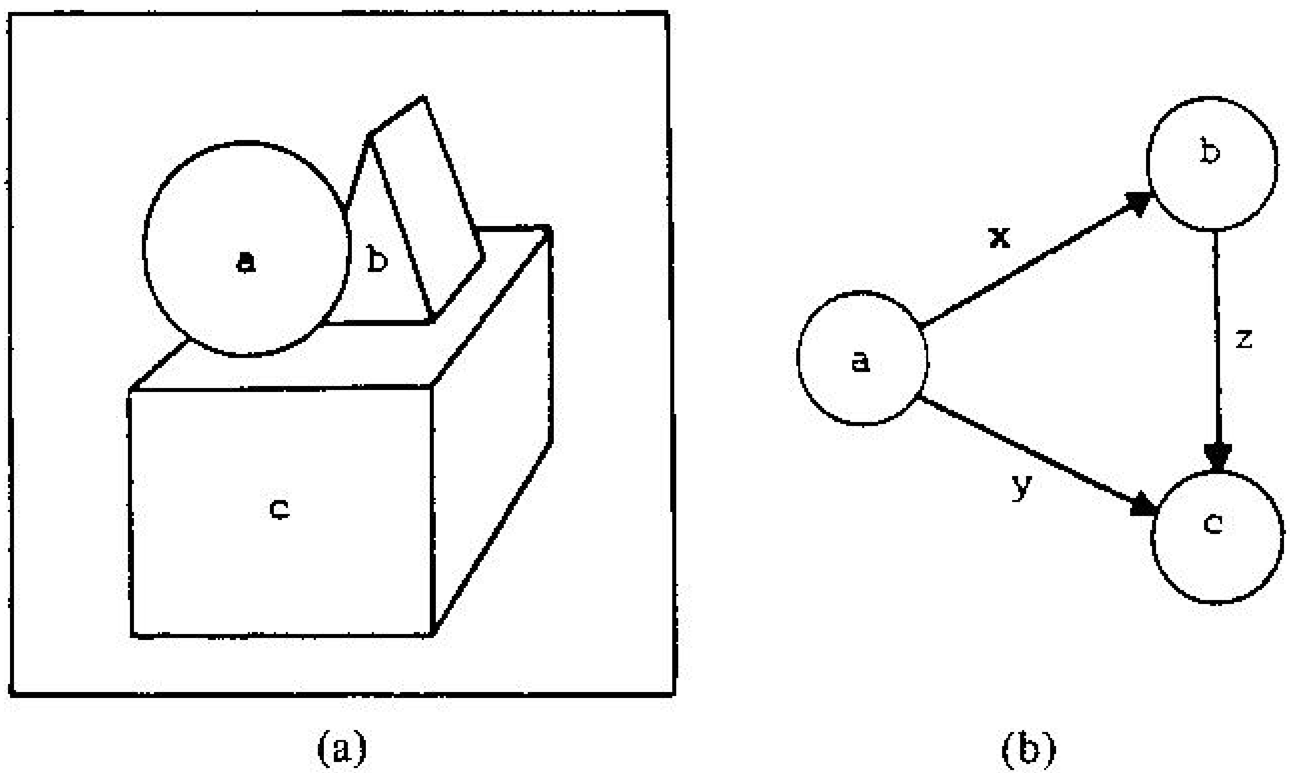


Figure 2: Attributed graph representation of image features[2]

Application of Attributed Graph

Social system

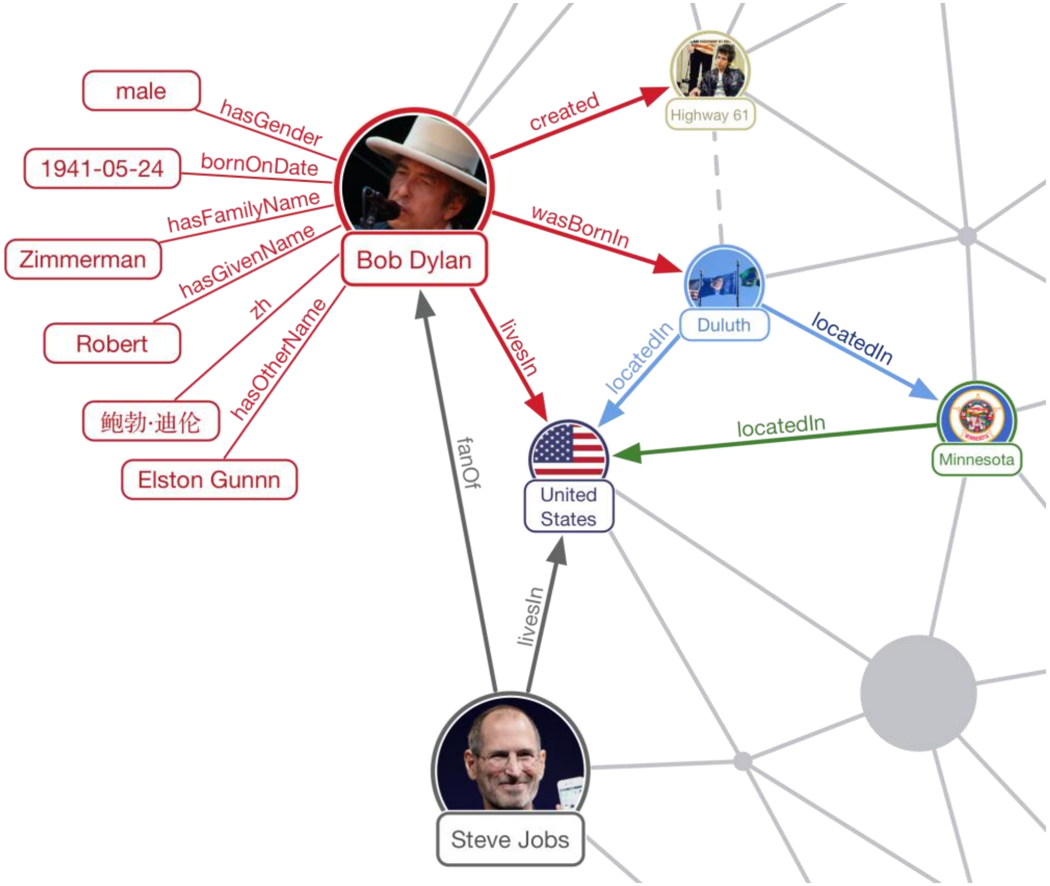


Figure 3: Attributed graph representation of social system[1]

Australian National University

Attributed Graph Matching

Given: 2 attributed graphs

Find: pairwise matching of the nodes across graphs

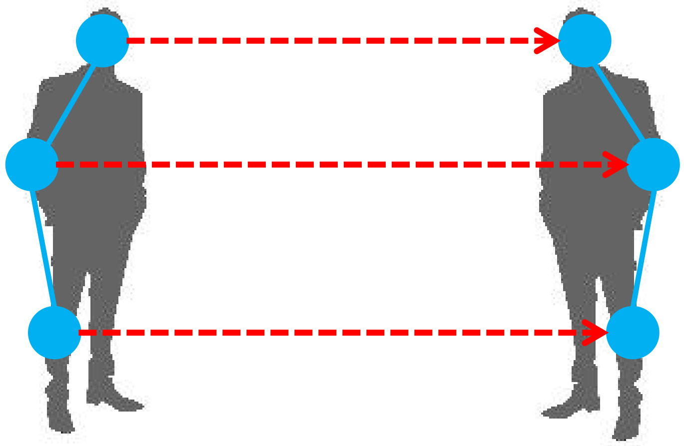


Figure 4: attributed graph matching example[3]

NP-complete

Attributed Graph Matching

Applications

- Computer Vision
  - Object recoginition
- Medicine
  - Diagnostics
- Biology
  - Biometric identification
- NLP
  - Document matching
- Recommender System

...

Inexact Attributed graph Matching

Observation: exact matches don't always exist, find the best match

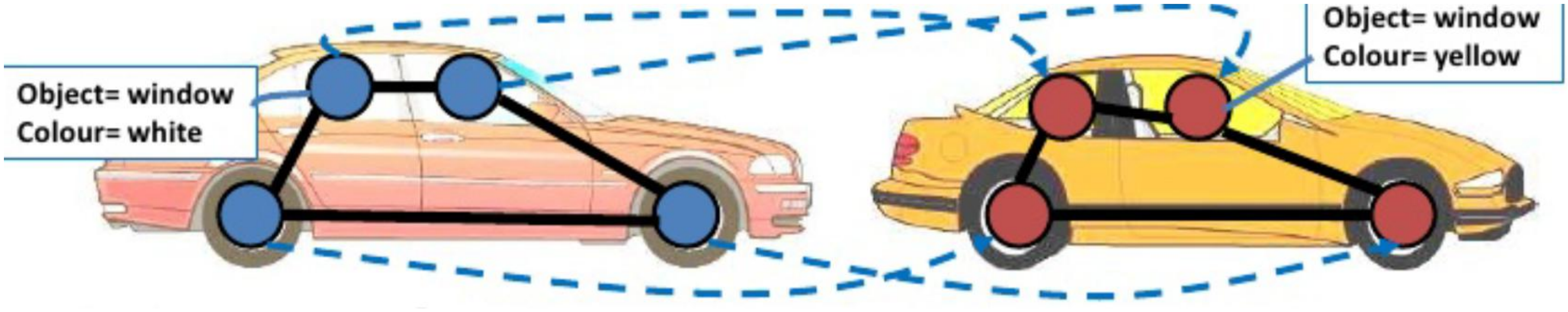


Figure 5: Inexact attributed graph matching example[4].

Attributed subgraph Matching

Find best match among in the graph(both structural and attibute)

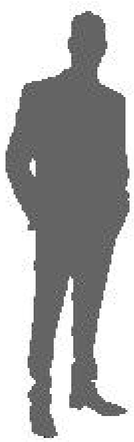


Figure 6: Crowd[5].

Problem Definition

Given: 2 attributed graphs G and G', as well as a (query) subgraph g from G

Find: The best matching subgraph *in* G'

NP-hard
Gap in existing methods

Existing Approaches

- Possible to enumerate all the candidate subgraphs, but inpractical.
- Existing approaches are all approximations

Existing Approaches

Index-based

1. Develop index functions for nodes to capture node information.

2. Apply approximation algorithms to find the optimal matching.

pros: index usually intuitive, e.g. node neighbourhood

cons: Require handcraft index function
Do not generalize well

Existing Approaches

Graph kernels

- ML kernel methods applied on graphs.
- Measures graph similarity.
- stackable
- Subgraph matching kernel, Optimal Assignment(OA) kernel

pros: many available kernels

cons: handcraft kernel function

Existing Approaches

Substructure-similarity based

1. Compute similarity for every pairs of substructures across graphs

2. Matching based on substructure similarity

pros: Can use ML to learn similarity measure

cons: Prone to error
Hard to reason about matching process
Space complexity

Our model

Purpose: To mitigate the problems in current approaches

- Use neural network to learn node representations and similarity
- considering the query subgraph as a whole

Our model

An overview

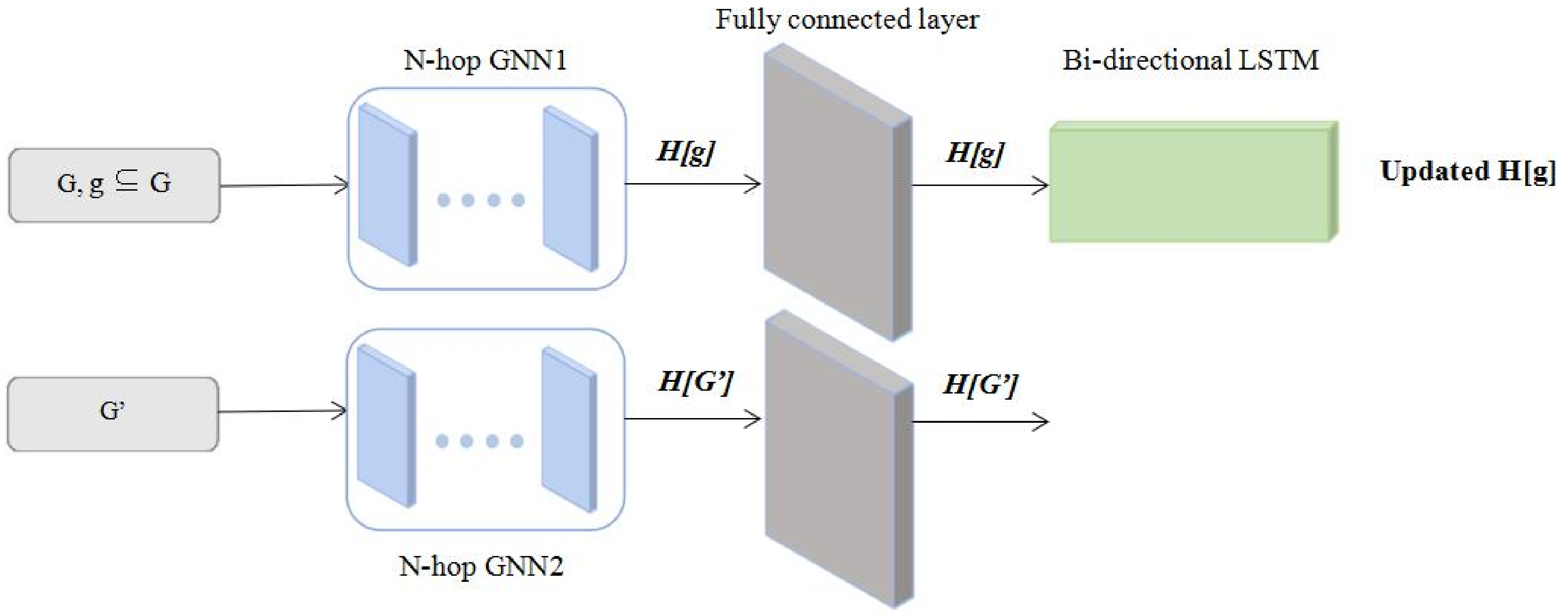


Figure 7: Model Overview. H(g) is the set of node embedding for nodes in g

Training stage

Novel idea: Train with pairs of matching subgraphs

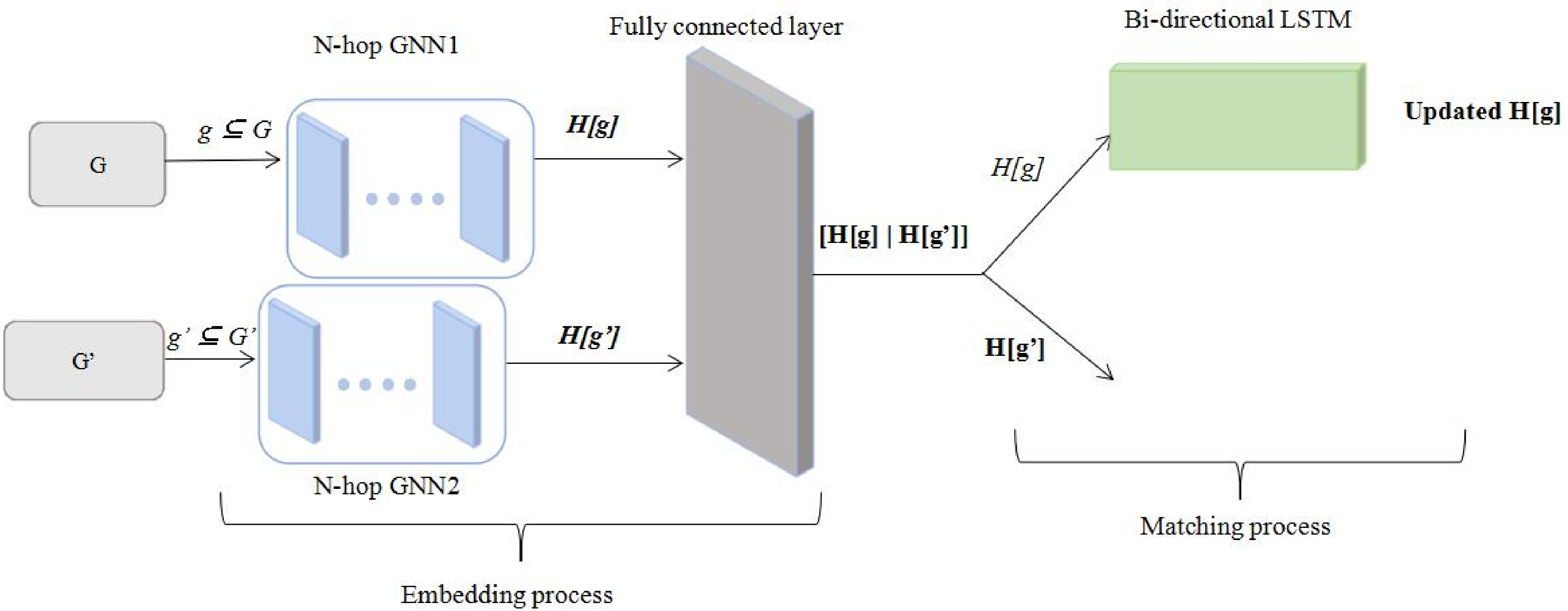


Figure 8: Model for the training stage.

Embedding process

- Graph Neural Network(GNN)

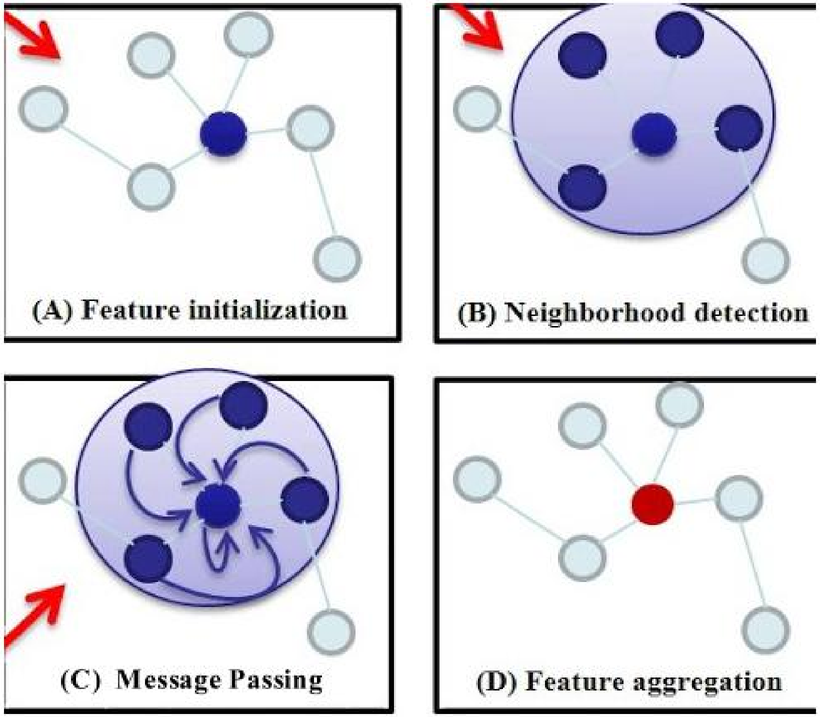


Figure 9: An illustration of 1-degree (hop) graph neural network[6]. The network works based on the principle of message passing.

Embedding process

- Fully connected layer
  - Combine the separate node embeddings, learn two graphs jointly

- Embedding Loss
  - L_e = difference in embeddings between graphs in the training pair

Matching process

- Inspired by NLP
- In sentences, words are closely correlated.
- Same for nodes in the same graph
- Novel idea: Treat graphs as sequence of nodes
- Learn the sequence
- LSTM: capture long-short term dependencies

Matching process

- How to sequence the nodes?
  - random sequence

Bi-directional LSTM learns dependencies in both directions

Matching process

- Bi-directional LSTM(BiLSTM)

Only g is input

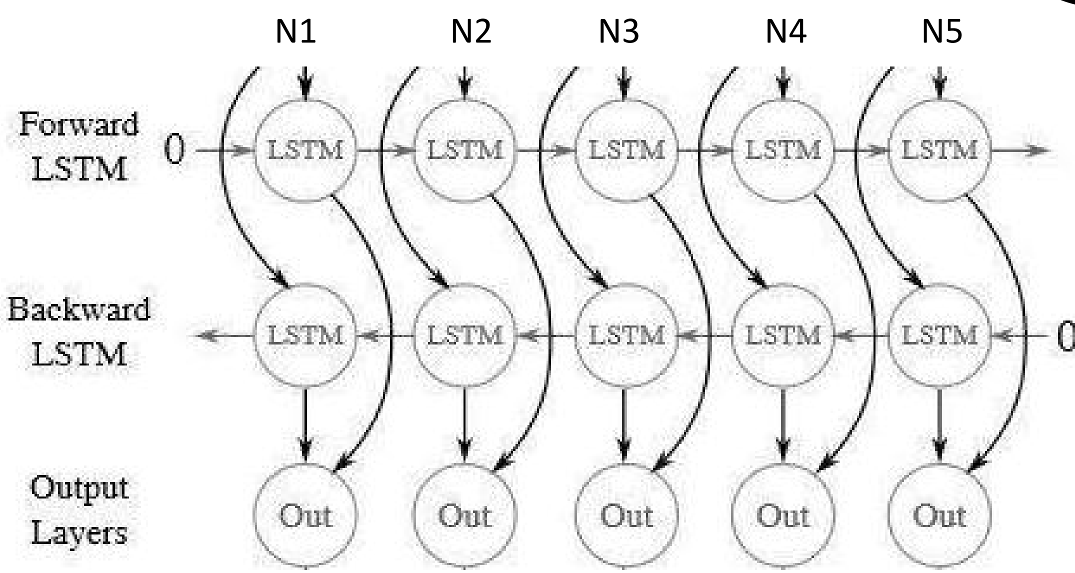


Figure 10: 1 layer Bi-directional LSTM used in the model.[7] The input is one subgraph in each pair.

Matching process

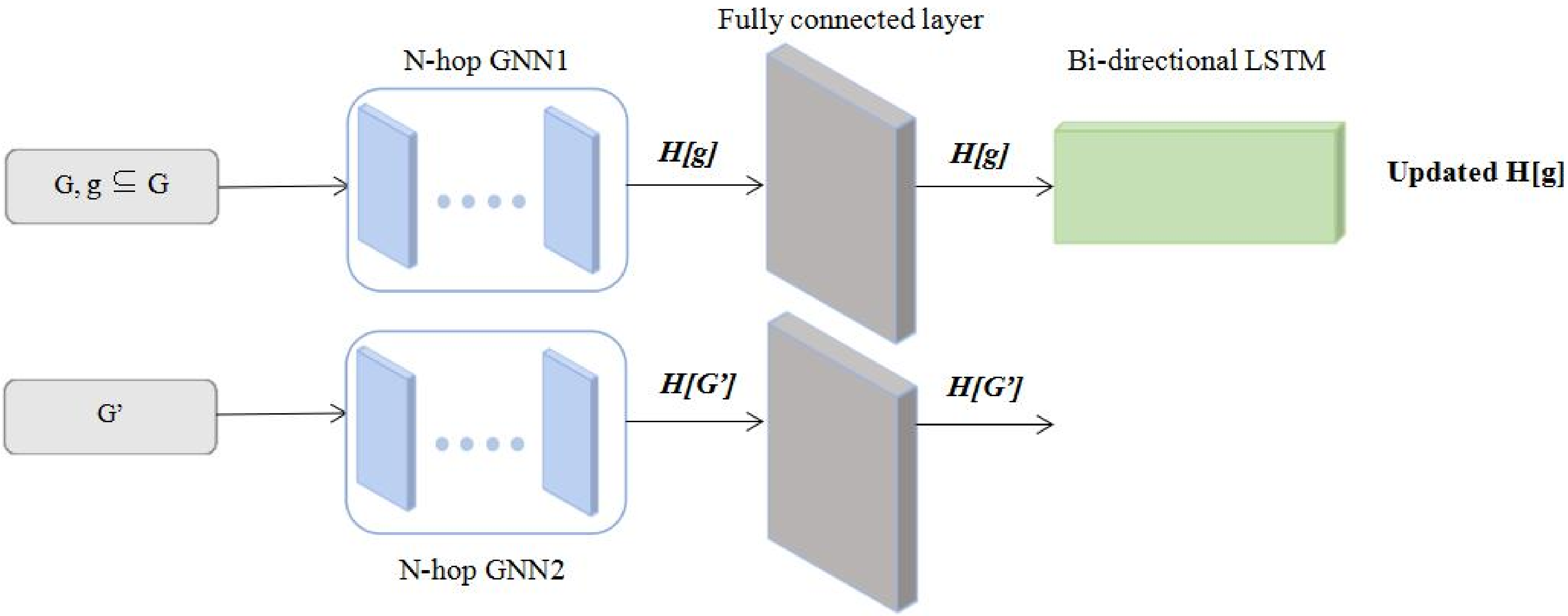
- Finding the best match
  - look for closest neighbour for each node using its embedding

- Matching Loss:
  - Let g'' be the best matching found,
    L_M = difference in distance matrix between query graph g and g''.

- Total loss:
  $L_E + \gamma L_M$ where γ is a hyperparameter, default is 0.5.

Testing stage

- Only have the query subgraph



Experiments

- Three datasets:
  1. Synthetic dataset
  2. Cora
  3. MovieLens

- Three early (~2010) baselines:
  1. TALE(index-based)
  2. Optimal Assignment(OA) kernel
  3. LG method(substructure-similarity based).

- Measure:
  1. Accuracy
  2. Precision

Synthetic Dataset

- Randomly generate 200 pairs of matching subgraphs of size 3, 4, 5 with attributes Color ∈ {R, G, B}
- Each pair obtained by randomly generating one and get the other with small modifications
- Aggregate the pairs to form the large graphs

Cora Dataset

- Classification dataset
- Entries: 2708 machine learning papers
- Each has 1433 features denoting the presence of 1433 keywords
- Class: type of paper
- Randomly extract 200 pairs of matching subgraphs of size 3, 4, 5 using a greedy approach.

MovieLens Dataset

- Dataset recording movie rating
- Entries: movies
- Each has 22 features denoting its information and rating
- Randomly extract 200 pairs of matching subgraphs of size 3, 4, 5.

Experimental Result

- Result for our model on different datasets, taking average of 10 runs

Dataset	Subgraph size	Accuracy	Precision
Synthetic	3	0.61	0.70
Synthetic	4	0.57	0.70
Synthetic	5	0.50	0.62
Cora	3	0.62	0.72
Cora	4	0.52	0.68
Cora	5	0.48	0.66
MovieLens	3	0.59	0.71
MovieLens	4	0.52	0.64
MovieLens	5	0.40	0.59

Results: Ok but not exceptional

For different datasets, scalablility changes

Comparison with baselines

- Synthetic dataset: Simple attributes, Little number of nodes

Table 4.1: Accuracy comparison on the four models on the 40 instances in the synthetic dataset

Subgraph size	Our model	LG	OA kernel	TALE
3	0.61	0.52	0.49	**0.75**
4	0.57	0.43	0.45	**0.70**
5	0.50	0.29	0.38	**0.63**

Second among the four methods

Comparison with baselines

- Cora dataset: Very complex attributes, Large number of nodes

Table 4.2: Accuracy comparison on the four models on the 40 instances in the Cora dataset

Subgraph size	Our model	LG	OA kernel	TALE
3	**0.62**	0.39	0.28	0.56
4	**0.52**	0.34	0.19	0.49
5	**0.48**	0.32	0.15	0.44

The best model in this set of experiment.

Result analysis

- For a model which introduces novel ideas, achieved moderate performance
- Performance problem-dependent, probably attributed to the loss function
- Some error may be resulted by the dataset processing step(the matching pairs found not accurate)
- Naive graph sequencing process

Future Work

- Compare with more recent baselines
- Try other models for processing sequential data
- Try embedding into other spaces
- Include edge attributes
- Involve heterogeneous graphs

Thank you!

References

[1]: Image from: https://medium.com/@brkyataman/knowledge-graph-and-youtube-29d259fd3dc1

[2]: Image from: A. Wong and M. You, Entropy and Distance of Random Graphs with Applications to Structural Pattern Recognition

[3]: Image from: https://www.kissclipart.com/silhouette-person-thinking-clipart-silhouette-clip-10cl2p/

[4]: Image from: https://www.slideshare.net/mobile/raul_A/attributed-graph-matching-of-planar-graphs

[5]: Image from: https://huaban.com/pins/1112150236/

[6]. Image from: Jun Wu, Jingrui He, Jiejun Xu, https://www.kdd.org/kdd2019/

[7]. Image from: https://www.sohu.com/a/283978749_717210