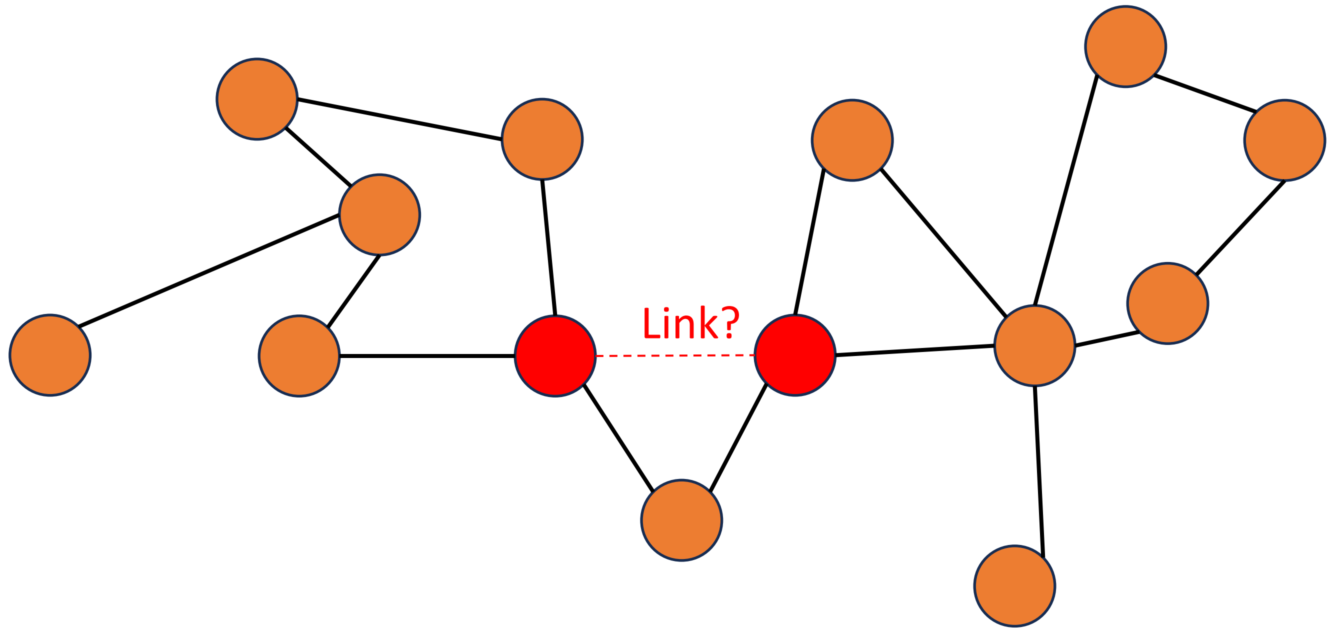
SEAL: Link Prediction Based on Graph Neural Networks

Authors: Muhan Zhang and Yixin Chen

Presenter: Dhroov Pandey

Link Prediction Problem



Applications in Recommendation systems, Knowledge graph completion, metabolic network reconstruction, etc.

Heuristics Based Approach

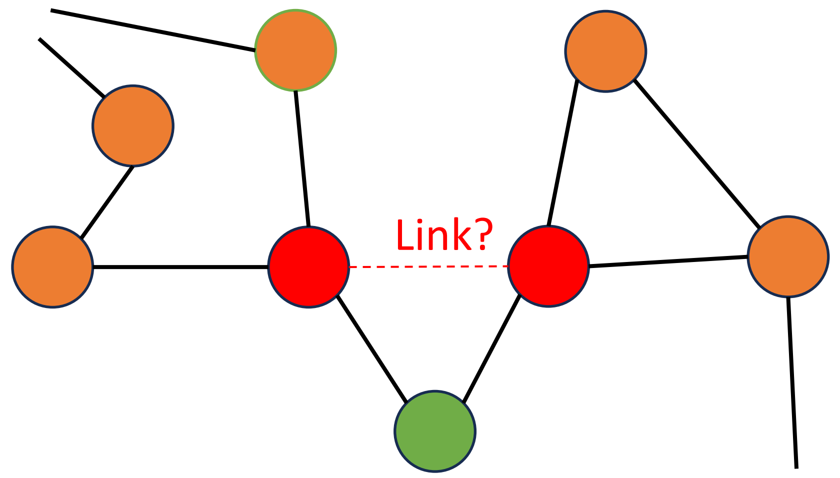
Compute a Node similarity score based on some heuristic related to the neighboring nodes.

Heuristic Degree is classified based on the size of the neighborhood (in hops).

First order (1-hop) heuristics:
Common Neighbor, Preferential attachment

Second Order (2-hop) heuristics:
Adamic-Adar (AA), Resource Allocation

High Order (k-hop k>3) heuristics:
Katz, Rooted PageRank, SimRank



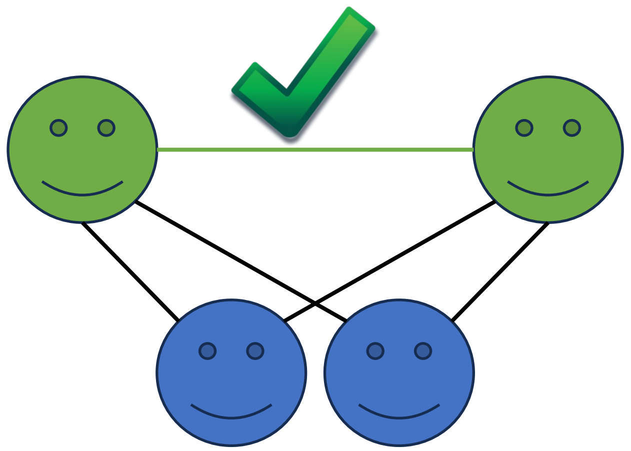
Common Neighbor Heuristic

Limitations with Heuristic Based approach

Requires a lot of Trial & Error to find the right heuristic for a particular network.

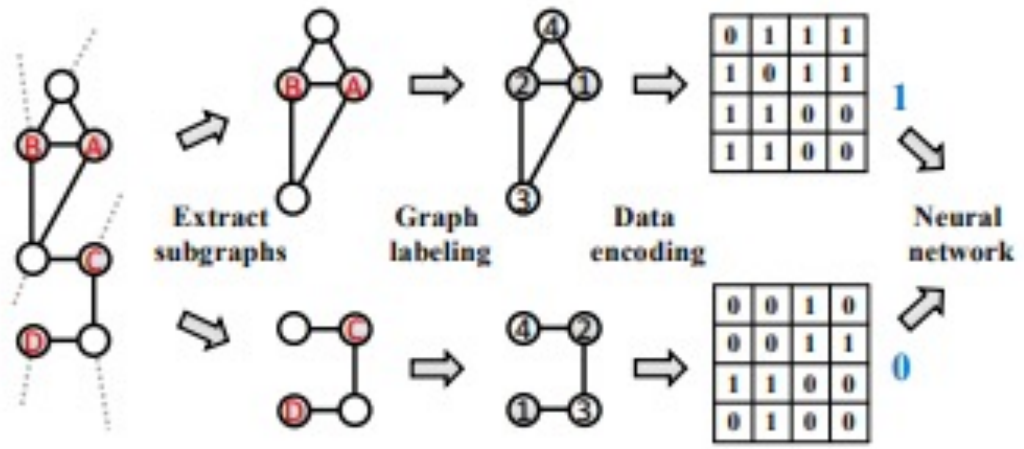
E.g., Common Neighbor heuristic works well in social networks but fails in protein-protein interaction (PPI) networks

This makes the approach non-generalizable since every network will require its own heuristic.



Using DNN to automatically study graph structure heuristics

- First architecture to use this method – Weisfeiler-Lehman Neural Machine
- Achieved state of the art prediction performance
- Uses Weisfeiler-Lehman graph labeling kernel to encode structural information in node label order
- Limitations:
- DNN accepts fixed sized tensors, leading to loss of information due to truncating or dilution due to padding.
- Adjacency matrix representations can't fully encapsulate explicit graph features.



Latent Feature Extraction

Certain matrix factorizations extract low dimensional latent features of the network. E.g. Matrix factorization, stochastic block model, etc.

Other embedding techniques like DeepWalk, and Node2Vec implicitly factorize the matrix.

The latent features can be combined with explicit and structural features to generate meaningful representations.

Computational Limitations

- Very large graphs can be computationally infeasible for Neural processing.

- Can process subgraphs around the target nodes to encapsulate graph context.

- However, this might seem to cause a loss of high-order heuristic learning, and high
Order heuristics perform better than low order ones.

SEAL shows that high order heuristic can be sufficiently approximated by local enclosing subgraphs.

First Contribution: γ −decaying theory

$Assertion$: $High\ order\ \gamma$ −decaying heuristics can be approximated using a smaller h-hop neighborhood

Preliminaries:

$Theorem$: $h - order\ heuristic\ for\ (x,y)\ can\ be\ calculated\ using\ an\ enclosing\ h - hop\ subgraph\ denoted\ by\ G_{x,y}^{h}\ which\ contains\ the\ h - hop\ neighbourhoods\ of\ x\ \&\ y$

Def: $A\ \gamma - decaying\ heuristic\ for\ (x,y)\ has\ the\ form$:

$$H(x,y) = \eta \sum_{l=1}^{\infty} \gamma^{l} f(x,y,l)\ ;\ \gamma \epsilon [0,1], f\ is\ a\ nonnegative\ function\ of\ x, y, \&\ l$$

$Theorem$: $A\ \gamma - decaying\ heuristic\ H(x,y) = \eta \sum_{l=1}^{\infty} \gamma^l f(x,y,l)\ can\ be\ approximated\ using\ G_{x,y}^h$ $and\ the\ approximation\ error\ decreases\ by\ \theta\left(e^h\right)\ if\ f\ statisfies\ the\ following\ properties$:

1. $f(x,y,l) \leq \lambda^l\ where\ \lambda < \frac{1}{\gamma};\ and$
2. $f(x,y,l)\ is\ calculable\ from\ G_{x,y}^h\ \forall l \in \{1,2,\dots,g(h)\}, where\ g(h) = ah + b, \&\ a,b \in \mathbb{N}, a > 0$

$Proof$: $Let\ the\ approximated\ heuristic\ \mathcal{H}(x,y) = \eta \sum_{l=1}^{g(h)} \gamma^l f(x,y,l)$,

$then\ the\ approximation\ error\ can\ be\ shown\ as$

$$|H(x,y) - \mathcal{H}(x,y)| = \eta \sum_{l=g(h)+1}^{\infty} \gamma^l f(x,y,l) \leq \eta \sum_{l=ah+b+1}^{\infty} \gamma^l \lambda^l = \eta(\gamma\lambda)^{ah+b+1}(1-\gamma\lambda)^{-1}$$

$Therefore, a\ large\ a\ value\ and\ a\ small\ \gamma\lambda\ value\ will\ lead\ to\ faster\ error\ reduction$

Proving that Katz index is $\gamma - decaying$

$Katz\ Index\ is\ defined\ as\ Katz_{x,y} = \sum_{l=1}^{\infty} \beta^l |walks^{<l>}(x,y)| = \sum_{l=1}^{\infty} \beta^l |[A^l](x,y)|\ where, walks^{<l>}(x,y)$
$is\ the\ set\ of\ length\ l\ walks\ between\ x\ \&\ y, and\ A^l\ is\ the\ adjacency\ matrix\ to\ the\ lth\ power$

$Lemma$: $Any\ walk\ between\ x\ \&\ y\ with\ length\ l \leq 2h+1\ is\ included\ in\ G_{x,y}^h$

$Proof$: $Let\ w = <x, v_1, v_2, \ldots, v_{l-1}, y>\ be\ a\ walk\ of\ length\ 2h+1, so, for\ any\ v_i \in w, either\ d(v_i, x)$
$\leq h\ or d(v_i, y) \leq h, and\ therefore, v_i \in \{h - hop\ neighborhood\ of\ x\ or\ y\}\ which\ means\ v_i \in G_{x,y}^h$

$Setting\ \gamma = \beta\ \&\ \eta = 1\ and\ f(x,y,l) = \left|walks^{<l>}(x,y)\right|, by\ lemma,$
$\left|walks^{<l>}(x,y)\right|\ is\ calculable\ from\ G_{x,y}^h\ and\ so\ it\ satisfies\ property\ 2.$

$It\ satisfies\ property\ 1\ because$
$\left|walks^{<l>}(x,y)\right|\ is\ bounded\ by\ d^l\ where\ d\ is\ the\ maximum\ node\ degree\ of\ the\ network;$

$Setting\ \lambda = d\ yields\ a\ \gamma - decaying\ heuristic\ as\ long\ as\ d < \frac{1}{\beta}, in\ most\ cases\ \beta$
$\leq 5e-4\ and\ the\ condition\ is\ fulfilled.$

Important Conclusion

The paper contains proofs regarding PageRank & SimRank being $\gamma - decaying\ heuristics$

Therefore, high hop neighborhoods are not required for approximating high order heuristics and the neural networks can utilize low h-hop neighborhoods to yield high performance.

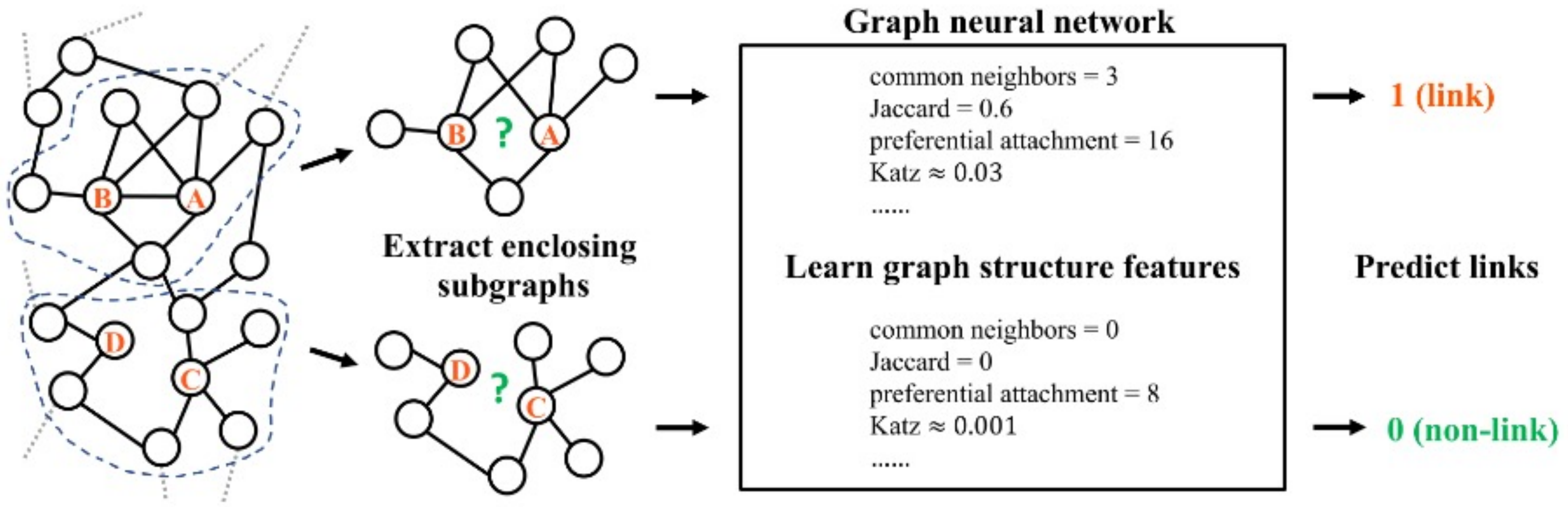
Intuition for the theory:
It is reasonable that nodes located far away in a network would have little impact on the target nodes compared to closer ones.

SEAL: Implementation

3 step process:

1. Subgraph Extraction
2. Node Information Matrix Construction
3. GNN Training

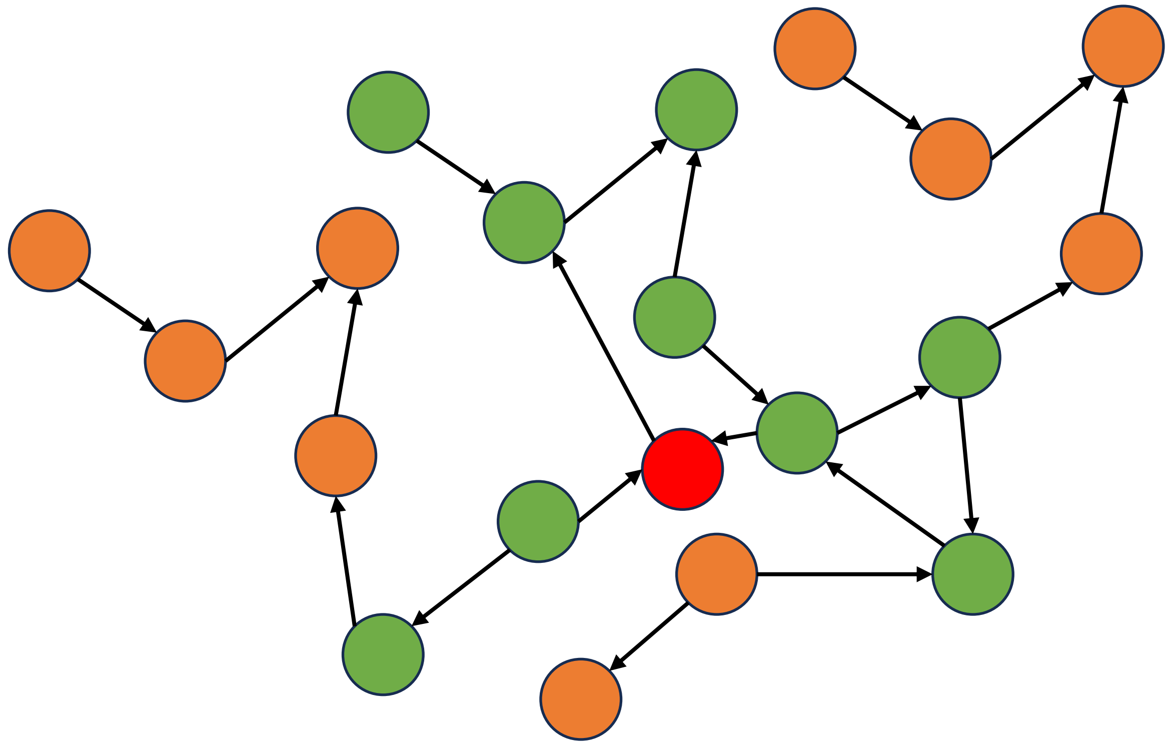
SEAL Execution Overview



Subgraph Extraction

Capture the h-hop neighborhood of both the link prediction nodes and union the sets.

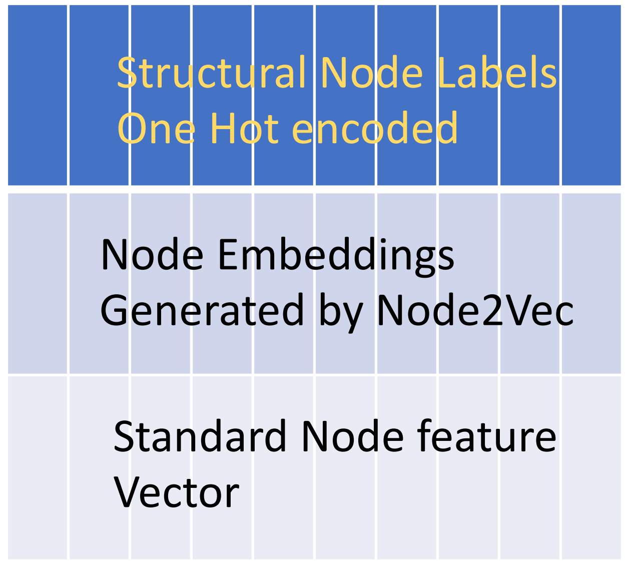
Can be extracted using BFT.



Node Information Matrix Construction

3 Main Components:

1. Structural Node Labels
2. Node Embeddings(Node2Vec)
3. Node Attributes

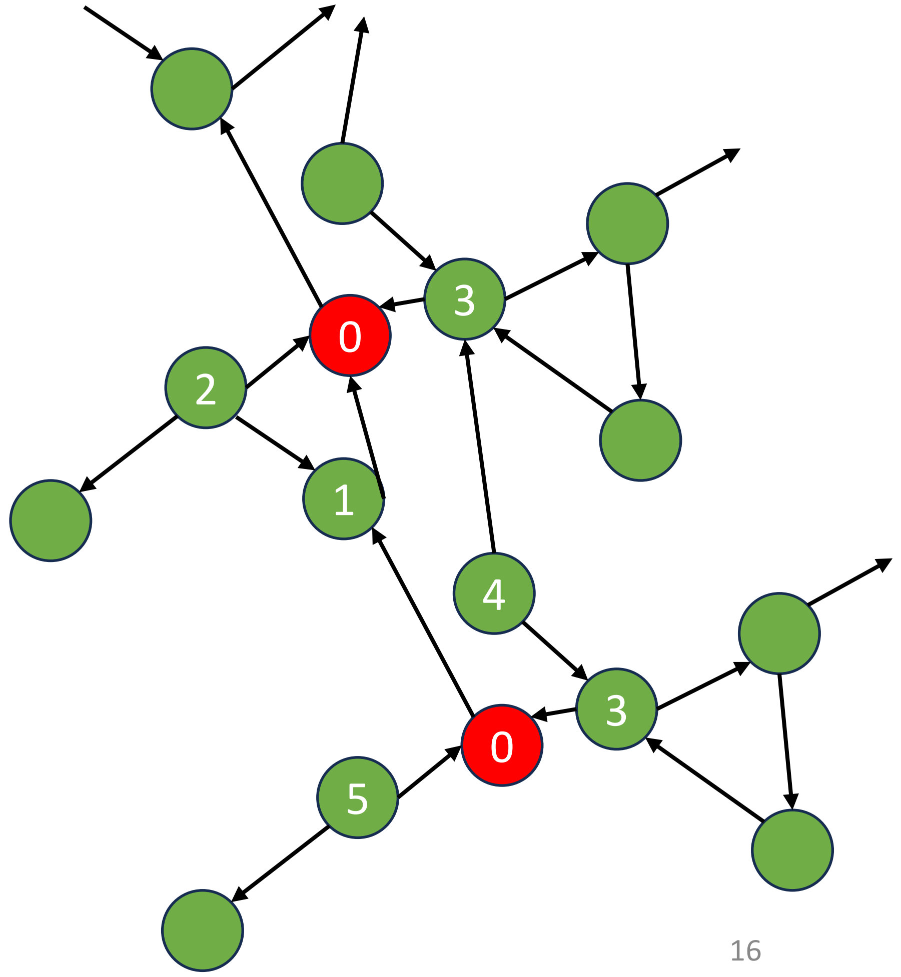


Structural Node Labeling

Encodes information regarding the topology of the subgraph.
Assign labels based on distance from both the central nodes.

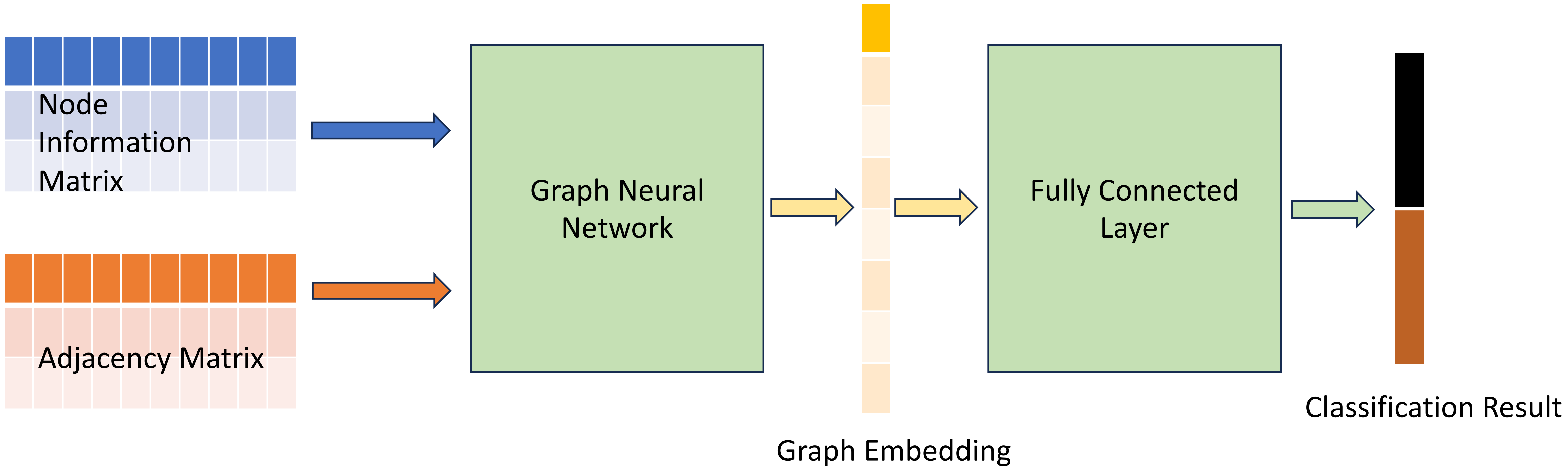
Novel Methodology and **second Contribution:**
Double Radius Node Labeling (DRNL)

DRNL uses a function f: $\mathbb{R}^2 \rightarrow \mathbb{R}$ which maps from (d(i,x), d(i,y)) to labeling 1,2,…,k $\forall i \in subgraph$; where d(m,n) is the distance between the nodes m & n

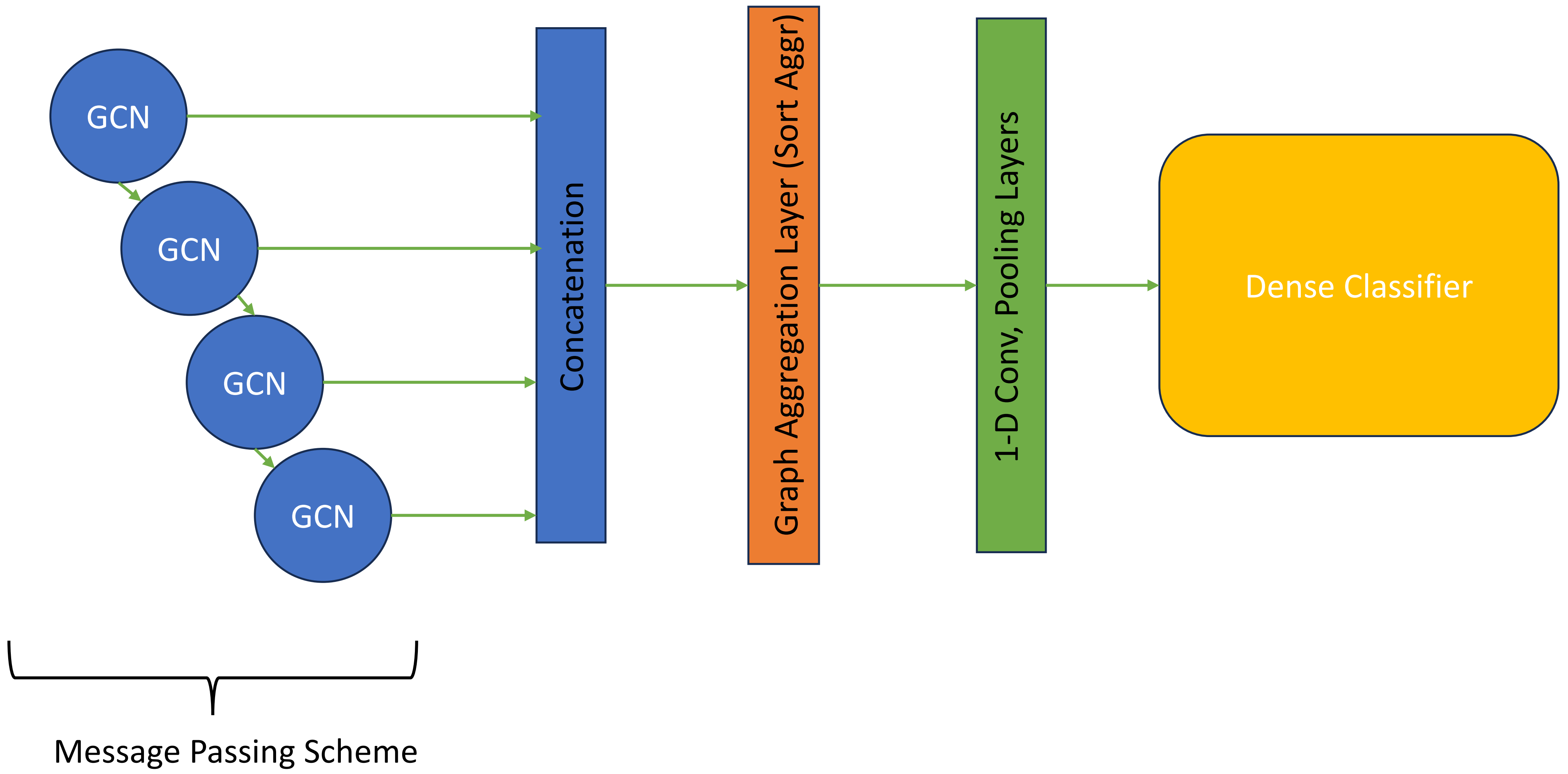


GNN

SEAL uses a standard DGCNN GNN and doesn't make any innovations in this domain, leaving it to future work. The Node Information Matrix and the Adjacency Matrix are passed as inputs to the GNN, which processes them through some convolution layers, followed by aggregation layers which generate a graph embedding vector, which is sent to a fully connected classifier.



DGCNN Architecture


GCN
GCN
GCN
GCN
Concatenation
Graph Aggregation Layer (Sort Aggr)
1-D Conv, Pooling Layers
Dense Classifier
Message Passing Scheme