

# 그래프 노드 중요도 예측 과업: 모델 정리 및 그래프 연결 구조의 중요성 분석

조우성\*, 송승헌\*, 이재구<sup>o</sup>

## Graph Node Importance Estimation Tasks : A Review of Models and Analyzing Importance of Topology

Woo-seong Cho\*, Seung-heon Song\*, Jae-koo Lee<sup>o</sup>

### 요 약

그래프 (Graph) 데이터에서 노드의 중요도는 다양한 곳에서 사용된다. OTT (Over-The-Top)나 전자 상거래 서비스에서 사용자에게 추천할 제품을 선정하기 위해 또는 지식 그래프 (Knowledge Graph)에서 질의 검색 시 우선 순위를 결정하기 위해, 한정된 네트워크 자원을 할당하기 위한 우선순위를 정하기 위해 노드 중요도를 사용한다. 본 논문은 학습 방법 및 모델이 사용할 수 있는 데이터의 범위(입력 중요도, �지의 종류, 노드 중심성)에 따라 노드 중요도 예측 모델을 정리 및 비교하였다. 다루는 모델은 라벨이 있어야 하는 지도 학습 기반으로 한정되어 있다. 따라서 라벨을 얻지 못하거나 라벨이 적은 상황에서 모델을 사용하기 위해 미세 조정을 포함한 귀납적 실험을 수행했다. 한편, 원본 데이터와 목표 데이터의 노드 특징 벡터는 일반적으로 다르게 정의된다. 따라서 원본 데이터에서의 가중치가 목표 데이터에는 적절하지 않을 수 있다. 이에 노드 특징 벡터가 성능에 미치는 영향을 알아보고자 이를 제거한 상태로 학습하는 실험 또한 수행하였다. 실험 결과 대체로 성능 저하가 크지 않았다. 이는 노드 중요도 예측에서 그래프 연결 구조 (Topology)의 중요성을 시사하며, 라벨을 얻지 못하거나 라벨이 적은 상황에서도 귀납 학습을 통해 고품질의 노드 중요도를 얻을 수 있는 것을 시사한다. 이러한 연구 결과는 다양한 분야에서 노드 중요도 예측이 필요한 연구에 도움이 될 것이다.

**키워드 :** 노드 중요도 예측, 그래프, 그래프 신경망, 귀납 학습

**Key Words :** Node Importance Estimation, Graph, Graph Neural Network, Inductive Learning

### ABSTRACT

The importance of nodes in graph data is utilized across various fields. It is used for selecting products for user recommendations in OTT (Over-The-Top) platforms and e-commerce services, determining priority in query searches within knowledge graphs, and prioritizing the allocation of limited network resources. This paper summarizes and compares node importance estimation models, considering the training method and the scope of data used, including input importance, edge types, and node centrality. The models covered in this

※ 본 논문은 2022년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.RS-2022-00167194, 미션 크리티컬 시스템을 위한 신뢰 가능한 인공지능). 이 성과는 정부(과학기술정보통신부)의 재원으로 한국 연구재단의 지원을 받아 수행된 연구임(No.RS-2023-00212484, 복잡한 실제 주행환경에서 설명 가능한 움직임 예측)

♦ First Author : Kookmin University Department of Computer Science, chowoos@kookmin.ac.kr, 학생회원

° Corresponding Author : Kookmin University Department of Computer Science, jaekoo@kookmin.ac.kr, 정회원

\* Kookmin University Department of Computer Science, song1248@kookmin.ac.kr, 학생회원

논문번호 : 202308-067-A-RN, Received August 30, 2023; Revised October 17, 2023; Accepted January 8, 2024

paper are limited to supervised learning methods. So, for situations where labels are unavailable or sparse, we conduct inductive learning with finetuning to use the models. Additionally, as the node feature vectors are defined differently between source and target data, we also train and test the models without node feature vectors to investigate their impact. The results show that overall performance degradation was generally not significant. This highlights the importance of graph topology in node importance estimation, suggesting that high-quality node importance can be obtained through inductive learning even in situations where labels are unavailable or sparse. The insights from this research can greatly contribute to studies that require node importance estimation in various fields.

### 1. 서론

그래프 (Graph) 데이터는 주어진 요소들의 관계 정보를 표현, 분석 및 이해하는 데 중요한 도구로 사용되고 있다. 최근 OTT (Over-The-Top)나 전자 상거래 서비스가 발전함에 따라 사용자에게 추천할 제품을 선정하는 모델의 필요성이 증가하고 있다<sup>[1]</sup>. 또한 인공지능 개인 비서는 질의 검색 시 대담 우선순위를 결정하는데 지식 그래프 (Knowledge Graph)를 사용한다. 이들 모두 노드 중요도를 사용하여 성능을 높일 수 있다. 한편, 한정된 네트워크 자원을 할당하기 위한 우선순위를 정하는 데도 사용된다. 이는 실시간 통신, 비디오 스트리밍 등과 같이 코로나 이후 사용이 지속해서 증가하고 있는 애플리케이션에서 특히 중요하다. 그럼에도 불구하고, 우리가 아는 한 노드 중요도 예측 모델은 아직 체계적으로 정리되지 않았다. 따라서 본 논문에서는 노드 중요도 예측 모델을 정리하고자 한다. 표 1과 같이 모델 학습 방법 및 입력 중요도, 엣지의 종류, 노드 중심성 등 사용하는 데이터의 범위에 따라 정리하였다.

대부분의 노드 중요도 예측 모델은 라벨이 있는 데이터가 필요한 지도 학습 기반이다. 하지만 노드 중요도

라벨을 얻는 것은 어려우며, 실제로 노드 중요도 예측 과업은 라벨이 제공되지 않는 상황이 가정된다. 귀납 학습을 통해 이런 상황에서 기존 지도 학습 기반 모델의 활용성 또한 확인한다.

본 논문은 다음과 같은 의미를 가진다.

1. 노드 중요도 예측에 특화된 정리 논문을 제안했다.
2. 모델들을 학습 방법뿐만 아니라 데이터를 얼마나 활용하는지에 따라서 구분했다.
3. 노드 중요도 예측에서 그래프 구조의 중요성을 분석했다.
4. 노드 중요도 예측 모델에 대한 귀납 학습 가능성을 통신 그래프를 통해 확인했다.

본 논문은 다음과 같은 순서로 진행된다. 2, 3장에서는 사용한 데이터 집합과 노드 중요도 예측 모델들을 설명한다. 4장에서 실험 내용 및 결과를 정리 후 5장에서 본 논문을 요약하며 마친다.

표 1. 노드 중요도 예측 모델 비교  
Table 1. Difference between Node Importance Estimation Models

Data Used \ Model	PR <sup>[9]</sup>	PPR <sup>[9]</sup>	APPNP <sup>[12]</sup>	RGCN <sup>[13]</sup>	CompGCN <sup>[14]</sup>	GAT <sup>[15]</sup>	GENI <sup>[7]</sup>	RGTN <sup>[10]</sup>
Importance		✓	✓				✓	✓
Relation				✓	✓		✓	✓
Centrality	✓	✓					✓	✓
Method								
Non-Trainable	✓	✓						
Convolution			✓	✓	✓			
Attention						✓	✓	
Transformer								✓

## II. 배경지식

### 2.1 그래프 데이터

본 논문에서 사용하는 데이터 집합은 지식 그래프 형태이다. 지식 그래프는 방향 그래프로, 출발 노드를 주체 (Subject)라 하고 도착 노드 (Node)를 대상 (Object)이라 칭하며, 이들을 잇는 엣지 (Edge)는 관계 (Relation) 혹은 술어 (Predicate)라 한다. 이때 주체, 관계, 대상을 묶어 트리플 (Triple)이라 부르며 영어 앞 글자를 따 (*s, r, o*)와 같이 나타낸다. 본 논문에서는 FB15k<sup>[2]</sup>, TMDb<sup>[3]</sup>, IMDB<sup>[4]</sup>의 세 가지 지식 그래프 데이터 집합과 분산 응용프로그램을 실행하는 컴퓨터 간의 통신 그래프인 CISCO<sup>[5]</sup>를 사용한다. 표 2에 이들에 대한 특징이 정리되어 있으며, 상세 내용은 다음과 같다.

FB15k는 FreeBase<sup>[6]</sup>의 부분 집합이다. 이는 일반 지식 기반의 트리플과 각 개체의 설명이 포함된 데이터 집합이다. 따라서 다른 지식 그래프에 비해서 많은 개수의 엣지 종류를 갖는다. 각 개체는 위키피디아 페이지와 연결되는데, Park 등<sup>[7]</sup>은 30일 동안의 해당 페이지 방문자 수를 수집하여 노드 중요도로써 활용하였다.

TMDb는 TMDb 5000 데이터 집합으로부터 만들어진 영화에 관련된 지식 그래프이다. 노드는 영화, 배우, 출연진, 제작진, 그리고 영화사들로 이루어져 있다. 영화의 인기도를 노드 중요도로써 활용한다.

IMDB는 또 다른 영화에 관련된 지식 그래프이다. 노드는 영화, 장르, 출연진, 제작진, 배급사, 그리고 배급된 국가들로 이루어져 있다. IMDB 웹사이트에서는 해당 영화에 대한 투표를 진행하는데, 투표수를 노드 중요도로써 활용한다. IMDB는 사용한 데이터 집합 중 가장 커다란 지식 그래프이다.

CISCO는 서로 다른 분산 응용 프로그램을 실행하는 기기 간의 통신을 나타낸다. 따라서 노드는 각 기기를 구분하지 않고 나타내며, 엣지는 기기 간의 통신을 나타낸다. 수집된 기업에 따라 총 22개의 그래프가 존재하여 표 2와 후술할 실험 결과는 평균치이다.

CISCO는 노드 중요도가 사전에 정의되어 있지 않기

때문에 본 논문에서는 많은 패킷을 주고받은 기기가 중요하다는 가정으로 송/수신한 총 패킷의 양을 노드 중요도로써 정의하였다. 노드 특징 또한 정의되어 있지 않아, 128차원의 벡터를 1로 채워 정의하였다.

### 2.2 그래프 신경망과 노드 중요도 예측 과업

그래프 신경망(Graph Neural Network, GNN)<sup>[8]</sup>은 그래프 데이터를 입력으로 하는 인공 신경망 (Artificial Neural Network)의 한 종류이다. 인공 신경망에서는 정형 데이터 형태의 입력에 뉴런의 가중치를 적용하여 특징을 추출한다. 이와 대조적으로, GNN은 가중치뿐 아니라 이웃 노드의 정보를 활용하여 각 노드의 특징 벡터를 갱신하고, 특징을 추출한다.

GNN의 과업은 노드, 그래프, 엣지 단위로 나뉜다. 노드나 그래프의 속성을 분류 (Classification)<sup>[12,13,15]</sup>하거나 회귀 (Regression)<sup>[7,8,10]</sup>하는 과업과 노드 간 연결 가능성을 예측하는 연결 예측 (Link Prediction)<sup>[13,14]</sup> 과업이 있다. 전자는 노드 특징 벡터나 한 그래프 내의 모든 노드 특징 벡터를 요약 (Readout)한 임베딩 벡터 (Embedding Vector)를 활용하여, 후자는 노드 특징 벡터 간의 관계를 분석하여 과업을 수행한다.

본 논문에서는 노드 회귀 과업의 하나인 노드 중요도 예측 (Node Importance Estimation)<sup>[7,10]</sup> 과업에 초점을 맞추고 있다. 이는 2.1에서 소개한 것처럼 다양하게 정의된 노드 중요도 *s*를 예측하는 과업이다.

노드 중요도 예측 과업은 학습 불가능한 모델인 PR (PageRank)<sup>[9]</sup>을 처음으로 발달하여 왔으며, 현재는 GENI (GNN for Estimating Node Importance), RGTN (Relational Graph Transformer Network)<sup>[10]</sup> 등 GNN 기반 학습 가능한 모델들이 제안되어 있다.

### 2.3 전도성 학습 및 귀납 학습

본 논문에서 다루는 중요도 예측 모델은 모두 지도 학습 기반이다. 즉 학습과 테스트를 하나의 데이터 집합에서 진행하는 전도성 학습 (Transductive Learning)으로 모델이 학습된다.

하지만 실제로 노드 중요도 예측 과업은 라벨이 없는

표 2. 지식 그래프 데이터 집합 요약  
Table 2. A summary of the knowledge graph datasets

Dataset	# Nodes	# Edges	# Relations	Input Score Type	# Node with Scores
FB15k[2]	14,951	592,213	1,345	# Pageviews	14,105 (94%)
TMDb[3]	114,805	761,648	34	Movie popularity	4,803 (4%)
IMDB[4]	1,124,995	9,729,868	30	# Votes for movies	202,538 (18%)
CISCO[5] (Avg.)	30,295	914,686	2	# of packets	30,295.05 (100%)

상황에서 수행되는 비지도 학습이 가정된다. 따라서 라벨이 있는 원본 (Source) 데이터를 이용해 모델을 사전 학습한 후 라벨이 없는 목표 (Target) 데이터의 라벨을 얻어내는 귀납 학습 (Inductive Learning)을 도입한다. 목표 데이터에 적은 양의 라벨이 주어지면 이를 이용하여 사전 학습된 모델을 미세 조정 (Finetuning)을 통해 추가 학습을 할 수도 있다.

귀납 학습은 서로 다른 그래프(도메인)에서도 모델이 잘 동작해야 하는 도메인 적응 (Domain Adaptation) 과업이기도 하다. 이때 가장 큰 전제는 사전 학습에 사용된 원본 데이터와 실제로 사용할 목표 데이터가 유사해야 한다는 것이다<sup>[11]</sup>. 이를 확인하기 위해 4장의 실험에서 데이터 집합 간의 유사성을 구조적 (Topological), 의미적 (Semantic)의 두 가지로 측정하였다. 구조적 유사성은 그래프 구조를, 의미적 유사성은 노드로 사용된 코퍼스 (Corpus)를 벡터화하여 계산하였다.

### III. 노드 중요도 예측 모델

#### 3.1 학습 불가능한 노드 중요도 예측 모델

##### 3.1.1 PageRank (PR) (9)

웹은 개별 웹페이지를 노드로, 하이퍼링크를 엣지로 하여 그래프로 표현될 수 있다. PR은 특정 웹페이지의 중요도를 들어오는 엣지가 많을수록 중요하다고 가정하여 계산한다 (식 1).

$$s_i = \beta \sum_{j \rightarrow i} \frac{s_j}{d_j} + (1 - \beta) \left[ \frac{1}{N} \right]_{1 \times N} \quad (1)$$

$d_j$ 는 노드  $j$ 의 차수로, 노드  $j$ 에서 나가는 엣지의 수이다.  $\beta$ 는 무작위 이동과 순간이동의 비율을 정하는 상수이며,  $N$ 은 노드의 수를 의미한다. 순간이동은 우변의 두 번째 항으로 표현되며, 식의 수렴성을 보장한다. 식 (1)을 반복 계산하여 얻은 수렴 값을 최종 노드 중요도로 사용한다.

개별 노드 중요도를 구하는 식 (1)을 전체 노드에 대해 식 (2)와 같이 표현할 수 있다.

$$s = \beta Ms + (1 - \beta) \left[ \frac{1}{N} \right]_{N \times N} \quad (2)$$

$s$ 는 노드 전체의 중요도,  $M$ 은  $M_{ij} = 1/d_j$ 인 열 확률 행렬이다. PR은 직관적인 아이디어로 그래프에서 노드 중요도를 계산한 초기 모델이라 할 수 있다. 그러나 그래

프 구조에 따라 값이 고정되는 학습 불가능한 모델이며, 그래프 구조 외의 정보를 활용하지 못하는 한계가 있다.

##### 3.1.2 PPR (Personalized PR) (9)

PPR은 순간이동을 특정 노드에 특정 확률로 하도록 변경한 모델이다. 순간이동 행렬을  $R$ 이라 하면 식 (3)과 같이 표현된다.

$$s = \beta Ms + (1 - \beta)R \quad (3)$$

개별 노드의 중요도를 이용하여  $R$ 을 정의하면 특정 페이지에 가중치를 부여할 수 있다. 하지만, 여전히 그래프 구조에 따라 값이 고정되는 학습 불가능한 모델이며, 개별 노드의 중요도 외의 정보는 활용하지 못하는 한계가 있다.

#### 3.2 학습 가능한 노드 중요도 예측 모델

본 논문에서 다루는 학습 가능한 노드 중요도 예측 모델은 GNN을 기반으로 한다. 본 장에서는 GNN의 노드 특징 벡터의 갱신 방법에 따라 노드 중요도 예측 모델을 정리하였다.

##### 3.2.1 합성곱 기반

그래프 합성곱 신경망 (Graph Convolutional Neural Network, GCN)은 식 (4)와 같이 이웃 ( $\mathcal{N}$ ) 노드의 특징을 응집 (Aggregation)하여  $l$ 번째 층에서 노드  $i$ 의 노드 특징 벡터  $H_i^l$ 을 갱신한다.  $\sigma$ 는 활성화 함수 (Activation Function),  $W$ 는 가중치이다.

$$H_i^{l+1} = \sigma \left( \sum_{j \in \mathcal{N}_i^{(l)}} H_j^l W^l \right) \quad (4)$$

이를 정규화 (Normalized)된 인접 행렬  $\hat{A}$ 와 전체 노드 특징 행렬  $H$ 에 대해 식 (5)와 같이 나타낼 수 있다.  $\hat{A}$ 는 인접 행렬  $A$ 에 자기-연결 (Self-loop)을 추가한  $\tilde{A}$ 와 이에 대한 차수행렬  $\tilde{D}$ 에 대해  $\hat{A} = \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}$  로 정의된다.

$$H^{l+1} = \sigma \left( \hat{A} H^l W^l \right) \quad (5)$$

대부분의 합성곱 기반 모델은 위 식을 기반으로 한다.

### 3.2.1.1 APPNP (Approximated PPR Propagation of Neural Predictions)<sup>[12]</sup>

GCN에서 층의 개수를 늘리게 되면, 데이터가 과도하게 평활해지는 문제 (Over Smoothing)가 있다. 이는 모든 갱신된 노드 특징 벡터가 비슷해지는 것을 의미한다. APPNP는 노드 특징 벡터 갱신 후에 PPR 전파를 진행하여 특정 노드에 대한 가중치를 잃지 않도록 해서 이 문제를 해결한다. 이를 위해 순간이동 행렬을 입력 노드 특징 행렬  $X$ 에 2개 층의 순방향 신경망 (Feed Forward Network, FFN)  $f_\theta$ 를 적용한  $R = f_\theta(X)$ 로 정의한다. APPNP는 식 (3)의 열 확률행렬  $M$  대신 정규화된 인접행렬을 사용하여 식 (6)과 같이 노드 특징을 갱신하며,  $H^p$ 로는  $R$ 을 사용한다.

$$H^{l+1} = \beta \hat{A} H^l + (1 - \beta)R \quad (6)$$

APPNP는 기존 GCN과 달리 노드 특징을 갱신하는 과정과 갱신된 노드 특징으로부터 중요도를 예측하는 과정이 분리되어 평활 등의 문제를 줄이고자 하였다. 하지만 다양한 종류의 관계(엣지)를 반영하지 못하는 한계점이 있다.

### 3.2.1.2 RGCN (Relational GCN)<sup>[13]</sup>

RGCN은 기존 GCN이 다양한 종류의 관계를 반영하지 못하는 문제를 해결하기 위해 고안되었다. 식 (7)과 같이 식 (4)에 합 기호를 하나 추가한 형태이다.

$$H_i^{l+1} = \sigma \left( \sum_{r \in R} \sum_{j \in \mathcal{N}_i^r} \frac{1}{d_{i,r}} W_r^l H_j^l + W_T^l H_i^l \right) \quad (7)$$

밑 첨자  $r$ 은 관계의 종류를,  $T$ 은 자기-연결을 나타낸다. 그러나 관계의 종류가 많아질 경우  $W$ 의 개수가 증가하여 모델의 복잡도가 급증하고, 과적합 (Overfitting)이 발생하기 쉽다. 따라서 가중치 행렬을 식 (8)과 같이 기저 (Basis)와 블록 대각 분해 (Block-Diagonal Decomposition)를 이용하였다.  $V_b$ 는 전체  $B$ 개 중  $b$ 번째 기저 벡터,  $a_{rb}$ 는  $r$ 번째 관계의  $V_b$ 에 대한 계수이다.

$$W_r^l = \sum_{b=1}^B a_{rb}^l V_b^l \quad (8)$$

RGCN은 다양한 관계를 적은 변수로 효율적으로 다루기 위해 과적합을 방지한 모델이나, 너무 많은 수의 관계를

다룰 땐 매개변수가 너무 많아지는 문제가 있다. 또한 입력 노드의 중요도를 받지 못하는 한계가 있다.

### 3.2.1.3 CompGCN (Composition GCN)<sup>[14]</sup>

CompGCN은 노드와 엣지에 대한 임베딩 (Embedding)을 구성 연산 (Composition Operation)으로 합치고 엣지의 방향에 따라 세 개의 가중치만을 사용하여 매개변수를 효과적으로 줄였다.

CompGCN은 트리플 형태의 데이터 집합으로 이뤄진 지식 그래프 자료형에 초점을 맞춰서, 대상 ( $o$ )에 대한 임베딩(노드 특징 벡터)을 식 (9)와 같이 주체 ( $s$ )와 관계 ( $r$ ) 특징 벡터의 구성 연산 ( $\phi$ )으로 임베딩한다.

$$H_o = \phi(H_s, H_r) \quad (9)$$

CompGCN은 기존의 관계  $R$ 에 역전 관계  $R^m$ 와 자기-연결  $T$ 를 추가로 정의하고, 각각 서로 다른 가중치 ( $W_{\lambda(r)}$ )를 적용하여 식 (10)과 같이 노드 특징 벡터를 갱신한다.

$$H_i^{l+1} = f \left( \sum_{(j,r) \in \mathcal{N}_i} W_{\lambda(r)} \phi(H_j^l, H_r^l) \right) \quad (10)$$

$$H_r^{l+1} = W_{rel} H_r^l \quad (11)$$

또한, 식 (11)과 같이 관계에 대한 특징 벡터도  $W_{rel}$ 로 갱신하여 관계를 더욱 효과적으로 다룬다.

CompGCN은 RGCN보다 좀 더 효율적으로 서로 다른 관계들을 다룬 모델이다. 하지만 개별 노드의 중요도를 입력으로 받지 못하는 한계가 있다.

### 3.2.2 어텐션 (Attention) 기반

GCN 기반의 모델은 대상 노드에 대한 이웃 노드의 가중치가 두 노드의 차수 곱의 제곱근으로 일정하다. 반면 어텐션 기반의 모델은 두 노드 간의 상대적인 가중치 또한 학습한다.

#### 3.2.2.1 GAT (Graph Attention Network)<sup>[15]</sup>

가장 기본적인 모델은 GAT이다. Bahdanau의 어텐션 메커니즘<sup>[16]</sup>과 비슷하게 어텐션 계수  $\alpha$ 는 식 (13)과 같이 계산된다. 여기서 노드  $s$ 에 대한 노드  $j$ 의 중요도를 뜻하는  $e_j$ 는 식 (12)와 같이 계산된다.  $\tilde{a}$ 는 어텐션 메커니즘에 해당하는 한 층의 완전 연결 층이며,  $W_{\tilde{a}}$ 는 변환 행렬 (Transformation Matrix)이다.  $\parallel$ 는 연결

(Concatenate) 연산자를 의미한다. 이 어텐션 계수를 이용하여 식 (14)와 같이 한 층의 갱신이 이뤄진다.

$$e_{ij} = \text{LeakyReLU}(\vec{a}^T [W_F H_i \parallel W_F H_j]) \quad (12)$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})} \quad (13)$$

$$H_i^{l+1} = \sigma \left( \sum_{j \in \mathcal{N}_i} \alpha_{ij}^{l+1} W^{l+1} H_j^l \right) \quad (14)$$

어텐션 학습 과정의 안정성을 위하여 여러 개의 어텐션을 사용하기도 하며 이를 다중 어텐션 (Multi-Head Attention)이라 한다.

GAT는 GCN에 마스크된 (Masked) 자기-어텐션 (Self-Attention)을 적용하여 효율적으로 성능을 높였지만 다양한 종류의 관계를 반영하지 못하는 한계가 있다.

### 3.2.2.2 GENI (GNN for Estimating Node Importance)<sup>[7]</sup>

GENI는 노드 중심성을 이용해 그래프의 구조적인

정보를 포착하는 한편 노드 특징 벡터가 아닌 스칼라의 노드 중요도를 응집하고 갱신한다.

GENI는 그림 1.a처럼 크게 점수 계산 (Scoring Network), 점수 응집 (Score Aggregation), 중심성 조정 (Centrality Adjustment) 단위로 나뉜다. 이들을 쌓아서 전체 GENI 모델이 만들어지며, 그림 1.b에서는 두 개 층을 각각 2개의 다중 어텐션을 이용하여 쌓았을 때를 예시로 보여주고 있다. 한 층을 지난 후엔 다중 어텐션의 정보를 통합하기 위해 중간 응집 (Intermediate Aggregation) 단위에서 응집하고 다시 나누는 과정을 거치며, 마지막 층에서는 중심성 조정 단위를 통해 차수에 대한 과적합을 방지하고 최종 응집 (Final Aggregation) 단위에서 노드 중요도를 예측한다.

점수 응집 단위에서 노드  $i$ 의 점수는 식 (15)과 같이 어텐션을 기반으로 갱신된다.  $\alpha_{ij}$ 는 GAT와 동일하게 (식 13) 계산되지만  $e_{ij}$ 는 식 (15)와 같이 계산된다. 두 노드뿐 아니라 사이의  $r$ 번째 관계에 대한 임베딩  $\phi(p_{ij}^r)$ 을 같이 연결한다.

$$s_i^{l+1} = \sum_{j \in \mathcal{N}_i} \alpha_{ij}^{l+1} s_j^l \quad (15)$$

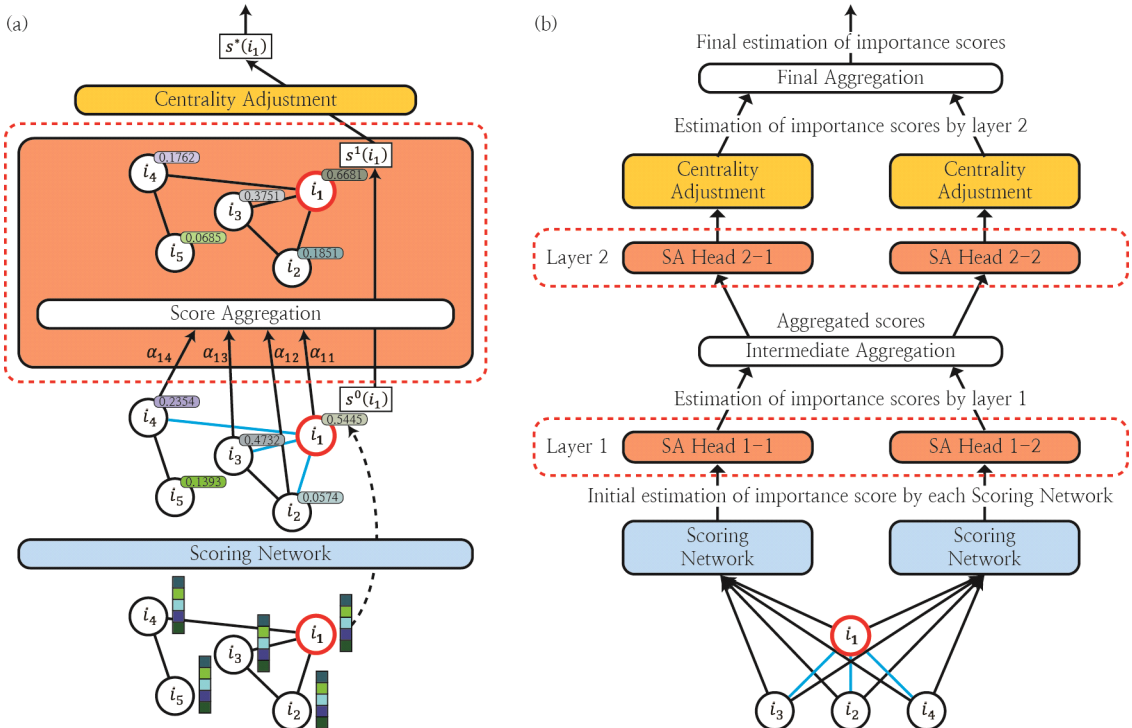


그림 1. GENI[7] 모델 모식도  
Fig. 1. The framework of GENI[7]

$$e_{ij} = \sigma \left( \sum_r \vec{a}^T [s_i \parallel \phi(p_{ij}^r) \parallel s_j] \right) \quad (16)$$

$s^0$ 는 노드 특징 벡터 입력으로 FFN으로 구성된 점수 계산 단위를 거쳐 계산된다.

최종적인 노드 중요도  $s^*$ 는 식 (17)과 같이 마지막 층에서의 점수  $s^l$ 에 유연하게 (Flexible) 조정된 중심성  $c^*$ 를 적용하여 계산된다.

$$s_i^* = \sigma(c_i^* \cdot s_i^l) \quad (17)$$

노드의 중심성은 식 (18)과 같이 노드 차수의 로그 값으로 정의한다.  $\epsilon$ 는 로그의 진수가 0이 되지 않도록 하는 작은 양의 상수이다.  $\gamma$ 와  $\beta$ 는 학습 가능한 매개변수로써 차수에 과적합 되는 것을 방지하기 위해 중심성을 유연하게 조정하는 중심성 조정 단위의 연산이다.

$$c_i = \gamma \cdot \log(d_i + \epsilon) + \beta \quad (18)$$

GENI는 이전 모델들과 다르게 지식 그래프에서 이웃의 정보를 어텐션 기반으로 가중치를 주고, 다중 관계를 고려하며 노드의 중심성과 입력된 중요도를 모두 사용한 모델이다. 또한, 노드의 중심성이 줄 수도 있는 단점을 해소하기 위해 유연함도 추가했다.

### 3.2.3 트랜스포머(Transformer) 기반

트랜스포머 (Transformer)<sup>[17]</sup>는 자연어 처리 분야 (Natural Language Progressing)에서 혁신적인 모델로 개발되었으며, 사진 관련 과업에도 널리 응용되고 있다

[18]. 그래프에서는 트랜스포머 인코더 (Encoder)에 노드 특징 벡터를 토큰 (Token)으로 간주하여 입력하고 갱신한다.

#### 3.2.3.1 RGTN (Relational Graph Transformer Network)<sup>[10]</sup>

RGTN 모델은 구조적 (Structural), 의미적 (Semantic) 트랜스포머 인코더를 병렬로 이용하여 그래프의 구조적, 의미적인 정보를 동시에 포착한다 (그림 2.a).

두 트랜스포머 인코더에는 노드 특징 벡터가 다르게 정의된 같은 구조 (Topology)의 그래프가 들어간다. 노드의 구조적 특징은 Node2Vec<sup>[19]</sup>을 통해 얻어지며, 의미적 특징은 노드의 설명을 트랜스포머-XL (Transformer Fixed Length)<sup>[20]</sup>로 얻는다. 두 인코더는 일반적인 트랜스포머와 동일한 자기-어텐션 기반으로 갱신된다 (그림 2.b).

자기-어텐션 중  $\alpha_{ij}$ 는 식 (13)과 동일하게 계산되며,  $e_{ij}$ 는 식 (19)와 같이 계산된다.

$$e_{ij} = \sum_r \frac{W_Q H_j \cdot W_K H_i^T}{\sqrt{d}} \cdot W_E p_{ij}^r \quad (19)$$

여기서 변환 행렬  $W$ 의 아래 첨자  $Q, K, E$ 는 각각 쿼리 (Query), 키 (Key), 값 (Value)을 의미한다. 값으로 옛지 임베딩을 사용해서  $E$ 라 한다.

노드 특징 벡터는 식 (22)와 같이 다중 어텐션으로 메시지  $M$ 을 생성한다. 그 후 식 (21, 22)처럼 정규화

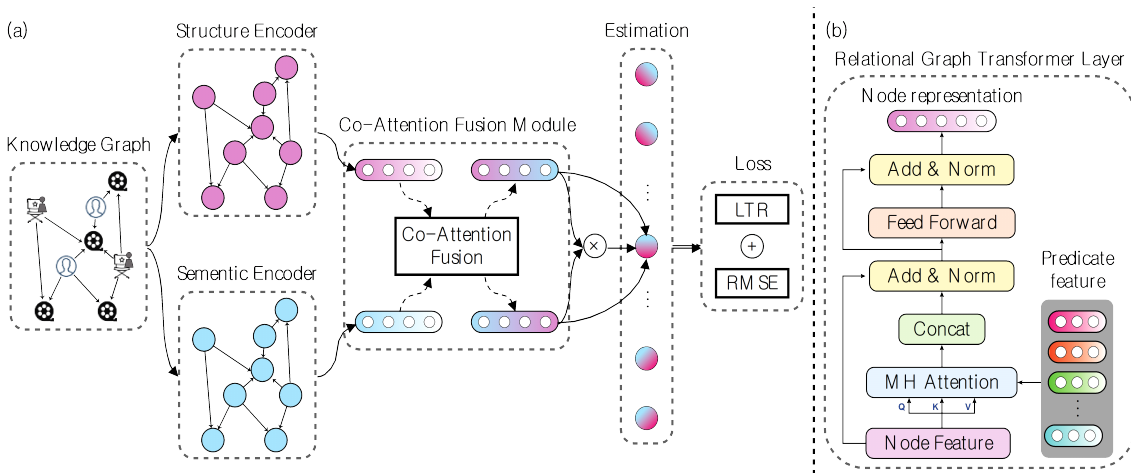


그림 2. RGTN[10] 모델 모식도  
Fig. 2. The framework of RGTN[10]

층 (Normalization Layer, Norm), 잔차 연결 (Residual Connection)과 FFN을 거쳐 노드 특징 벡터가 갱신된다.  $W_0$ ,  $W_v$ 는 변환행렬이다.

$$H_i^{l'} = \text{Norm}(H_i^l + M_i^l) \quad (20)$$

$$H_i^{l'+1} = \text{Norm}(H_i^{l'} + \text{FFN}(H_i^{l'})) \quad (21)$$

$$M_i^l = W_0 \parallel \left( \sum_{j \in \mathcal{N}_i} \alpha_{ij}^{k,l} \cdot W_V^{k,l} H_j^l \right) \quad (22)$$

구조적, 의미적 트랜스포머 인코더를 통해 갱신된 노드 특징 벡터를 각각  $H_i^s, H_i^c$ 로 표기한다. 이 둘은 노드의 서로 다른 특성을 포함하고 있으므로 상호-어텐션 합성 단위 (Co-Attention Fusion Module)로 합친다.

$$z_i^s = W_F H_i^s, \quad z_i^c = W_F H_i^c \quad (23)$$

$$\beta_{sc} = \frac{\exp(W'_Q z_i^s \cdot W'_K z_i^c)^T)}{\sum_{k \in \{s,c\}} \exp(W'_Q z_i^s \cdot W'_K z_i^k)^T)} \quad (24)$$

$$\begin{aligned} \hat{H}_i^s &= \text{Norm}(\text{FFN}(\beta_{ss} z_i^s + \beta_{sc} z_i^c) + H_i^s) \\ \hat{H}_i^c &= \text{Norm}(\text{FFN}(\beta_{cs} z_i^s + \beta_{cc} z_i^c) + H_i^c) \end{aligned} \quad (25)$$

$$s_i^s = W_s^1 \hat{H}_i^s, \quad s_i^c = W_s^2 \hat{H}_i^c \quad (26)$$

여기서  $W$ 는 변환 행렬이고, 식 (24)와 같이 상호-어텐션 계수를 계산하여 잔차 연결, Norm, FFN을 거쳐 갱신된 뒤 (식 25) 구조적 의미적 점수  $s^s, s^c$ 가 계산된다 (식 26).

이렇게 갱신된 두 점수를 이용하여 최종 점수  $s^*$ 를 구한다 (식 27). 이때  $\gamma$ 는 식 (28)과 같이 계산되는 어텐션 계수,  $\lambda$ 는 어텐션 벡터이다.

$$s_i^* = \text{LeakyReLU}(\gamma^s s_i^s + \gamma^c s_i^c) \quad (27)$$

$$\gamma^t = \frac{\exp(\hat{H}_i^t \lambda^T)}{\exp(\hat{H}_i^s \lambda^T) + \exp(\hat{H}_i^c \lambda^T)} \quad (28)$$

전체 손실함수  $L$ 은 MSE(Mean Squared Error)를 최종 점수  $s^*$ 와 구조적, 의미적 점수  $s^s, s^c$ 에 적용한  $L_0, L_1, L_2$ 과 더불어 LTR (Learning-To-Rank) 손실함수 (식 29)를 가중합하여 계산한다 (식 30). 이는 소프트맥스 (Softmax)를 적용한 라벨  $\hat{g}$ 와 예측 점수  $\hat{s}_i^*$  사이의

교차 엔트로피 (Cross Entropy)로 계산된다. 이를 통해 전체적인 노드 중요도 분포 또한 학습할 수 있게 된다.

$$L_i^r = - \sum_{j \in \mathcal{N}_i^r} \hat{g} \log(\hat{s}_i^*) \quad (29)$$

$$L = aL_0 + \frac{b}{2}(L_1 + L_2) + c \left( \frac{1}{|\mathcal{V}_s|} \sum_{i \in \mathcal{V}_s} L_i^r \right) \quad (30)$$

RGTN은 노드의 구조적, 의미적인 정보를 트랜스포머라는 효과적인 구조를 이용해 병렬로 인코딩하여 성능을 높였다. 하지만 의미적 트랜스포머 인코더의 입력을 위해 노드 설명이란 추가 정보를 트랜스포머-XL로 인코딩하는 과정이 필요한 단점이 있다.

## IV. 실험

### 4.1 평가 지표

본 논문에서는 평가 지표로 NDCG@K (Normalized Discounted Cumulative Gain)와 스피어만 (Spearman, SPM), 그리고 OVER@K (Overlap)를 사용하였다. @K는 평가 시 상위 몇 개의 결과를 사용했는지를 의미하며, 100으로 통일한다.

NDCG@K는 DCG@K (Discounted CG)를 정규화한 평가 지표이다. DCG@K는 식 (31)과 같이 추천된 K개의 중요도를 순위에 따라 가중치를 부여하여 더하여 구한다 (식 31).

$$\text{DCG@K} = \sum_{k=1}^K \frac{2^{s_k}}{\log_2(k+1)} \quad (31)$$

모델이 추천한 K개가 아닌 실제 라벨값을 이용하여 DCG@K를 계산한 것이 IDCG@K이며, 이를 이용해 식 (32)와 같이 정규화를 통해 0에서 1 사이의 값 NDCG@K를 구한다.

$$\text{NDCG@K} = \frac{\text{DCG@K}}{\text{IDCG@K}} \quad (32)$$

이때 사용하는 데이터 집합마다 전체 객체의 수가 다르므로 K를 늘려 모든 객체의 순위를 고려하긴 어렵다. 따라서 SPM을 이용한다.

SPM은 식 (33)과 같이 라벨값의 분포  $g$ 와 예측 점수의 분포  $s$  사이의 순위 상관관계를 측정하는 지표이다.



아래 첨자  $r$ 은 순위를 의미한다. 라벨값과 예측 점수가 완전 음의 상관관계를 따르면 -1, 완전 양의 상관관계를 따르면 1의 값을 가진다.

$$SPM = \frac{\sum_i (g_{ri} - \bar{g}_r)(s_{ri} - \bar{s}_r)}{\sqrt{\sum_i (g_{ri} - \bar{g}_r)^2} \sqrt{\sum_i (s_{ri} - \bar{s}_r)^2}} \quad (33)$$

$$OVER@K = \frac{|S_{top K}^{gt} \cup S_{top K}^*|}{K} \quad (34)$$

OVER@K는 식 (34)와 같이 예측한 상위 K개와 실제 상위 K개 사이에 겹치는 객체 수의 비율을 의미한다. OVER@K는 0에서 1 사이이다.

## 4.2 실험

### 4.2.1 모델 및 데이터 집합

PR과 PPR의 경우 NetworkX<sup>[21]</sup>에 구현된 것을 사용

하였다. 해당 모델의 출력인 특정 노드에 도착할 확률을 노드 중요도 예측값으로 사용하였다.  $\alpha$ 는 0.85, 최대 반복 횟수는 100으로 하였다. PPR의 순간이동 벡터는 PR의 결과 중 상위 10%의 노드로 중요도에 비례하게 순간이동 하도록 설정하였다. CompGCN과 GENI, RGTN은 공식 코드를 기반으로 변형하였고, 나머지 GNN 기반의 기준선 모델들은 DGL<sup>[22]</sup>에 구현된 예제를 응용하여 회귀 모델로 변형하였다.

사용한 데이터 집합은 FB15k, TMDB, IMDB와 CISCO의 네 가지이다. IMDB와 CISCO의 노드 특징 벡터 차원은 128차원, FB15k, TMDB는 64차원으로 서로 다르다. 귀납 학습을 위해 제로 패딩 (Zero Padding)을 이용하여 차원을 통일하였다. 각각 노드에 대응되는 위키 페이지의 방문자 수, 영화의 인기도, 영화에 대한 투표수와 패킷의 양을 노드의 중요도로 사용하였다. 8:1:1로 분할하여 각각 학습, 검증, 테스트에 사용하였고, 다섯 개의 서로 다른 분할의 평균 및 분산을 계산하

표 3. 서로 다른 모델, 방법으로 측정한 세 데이터 집합의 노드 중요도 예측 전도성 실험 결과  
Table 3. Experimental transductive results of different models and methods over the three datasets

Method	FB15k[2]			TMDB[3]			IMDB[4]		
	NDCG @ 100	SPM	OVER @ 100	NDCG @ 100	SPM	OVER @ 100	NDCG @ 100	SPM	OVER @ 100
PR[9]	0.1109	0.3169	0.0800	0.2207	0.3417	0.1900	0.1540	0.5198	0.1000
PPR[9]	0.1135	0.3171	0.0800	0.2178	0.3406	0.1900	0.0175	0.3196	0.0100
APPNP[12]	0.8171 ±0.0171	0.2669 ±0.0196	0.1420 ±0.0204	0.8052 ±0.0364	0.5712 ±0.0805	0.3450 ±0.0554	0.7434 ±0.0742	0.2140 ±0.0354	0.2780 ±0.0449
APPNP[12]*	0.8192 ±0.0160	0.3225 ±0.0160	0.1440 ±0.0233	0.8402 ±0.0089	0.6391 ±0.0130	0.4120 ±0.0598	0.9101 ±0.0219	0.3703 ±0.0044	0.4200 ±0.0210
RGCN[13]	0.7806 ±0.0478	0.2285 ±0.2583	0.0420 ±0.0643	0.4419 ±0.2078	-0.1049 ±0.3981	0.0920 ±0.1450	0.2198 ±0.0165	-0.1148 ±0.1087	0.0000 ±0.0000
RGCN[13]*	0.7804 ±0.0475	0.2276 ±0.2566	0.0400 ±0.0603	0.4419 ±0.2078	-0.1049 ±0.3980	0.0920 ±0.1450	0.2200 ±0.0165	-0.1126 ±0.1123	0.0000 ±0.0000
CompGCN[14]	0.7638 ±0.0392	0.3349 ±0.0178	<u>0.3080</u> ±0.0747	0.6283 ±0.0311	0.5295 ±0.1310	0.2355 ±0.0333	0.4738 ±0.0592	0.5140 ±0.1210	0.1708 ±0.0581
CompGCN[14]*	0.7588 ±0.0193	0.3117 ±0.0264	0.2604 ±0.0102	0.5818 ±0.0311	0.4712 ±0.1351	0.2528 ±0.0214	0.4615 ±0.0603	0.5028 ±0.0595	0.1486 ±0.0489
GAT[15]	<u>0.8892</u> ±0.0100	0.6322 ±0.0207	0.2260 ±0.0492	0.5254 ±0.2426	-0.0181 ±0.4316	0.1500 ±0.1741	0.3660 ±0.1992	0.0280 ±0.2325	0.0460 ±0.0920
GAT[15]*	0.7684 ±0.0433	0.0494 ±0.0515	0.0660 ±0.0388	0.6063 ±0.2125	0.1109 ±0.3321	0.1820 ±0.1003	0.9117 ±0.0210	0.3969 ±0.0416	0.4300 ±0.0219
GENI[7]	0.8890 ±0.0111	0.6472 ±0.0140	0.3000 ±0.0583	<b>0.9091</b> <b>±0.0101</b>	0.7781 ±0.0180	0.5420 ±0.0248	0.9526 ±0.0249	0.6981 ±0.0078	0.4960 ±0.0326
GENI[7]*	0.8125 ±0.0160	0.3219 ±0.0162	0.1680 ±0.0264	0.8503 ±0.0088	0.6863 ±0.0123	0.4180 ±0.0571	0.9160 ±0.0219	0.4353 ±0.0175	0.4340 ±0.0233
RGTN[10]	<b>0.9267</b> <b>±0.0020</b>	<b>0.7606</b> <b>±0.0100</b>	<b>0.2850</b> <b>±0.0050</b>	0.9064 ±0.0187	<b>0.7828</b> <b>±0.0140</b>	<b>0.5450</b> <b>±0.0350</b>	<b>0.9602</b> <b>±0.0006</b>	<b>0.7087</b> <b>±0.0096</b>	<b>0.5400</b> <b>±0.0141</b>

여 실험 결과를 표 3에 정리하였다.

4.2.2 실험 설정

본 논문의 실험은 크게 전도성 및 귀납 학습으로 나뉜다. 귀납 학습의 경우 학습에 추가로 사용한 목표 데이터의 양에 따라 0, 5, 10, 20, 40, 80, 100%의 7가지로 나뉜다. 학습에 목표 데이터를 사용하지 않아도 준수한 성능을 보여 (표 5) 목표 데이터를 일부 사용해 더욱 준수한 성능을 얻을 수 있는지 확인하고자 하였다.

한편, 실제 상황에서는 노드 특징 벡터가 사용자에게 따라 다르게 정의되어 원본과 목표 데이터의 노드 특징 벡터의 각 요소 의미가 달라 원본 데이터에서 학습된 가중치가 목표 데이터에서는 효과적이지 않을 수 있다. 이에 노드 특징 벡터의 모든 요소를 1로 대체하여 노드 특징 벡터를 사용하지 않는 상황을 추가하였다.

따라서 본 논문의 실험은 학습과 테스트 데이터에서 노드 특징 벡터의 사용 유무에 따라 표 4와 같이 6가지로 나뉜다. "O"와 "X"는 각각 노드 특징의 사용 유무를 지칭한다.

전도성 학습의 (1)과 (2)번 실험 결과는 표 3에 정리하였다. 모델 옆에 \*가 붙은 것이 (2)번 실험의 결과이다. 사용한 모델 중 RGTN은 그래프의 구조적, 의미적 정보를 모두 갖고 오기 위해 노드 특징을 따로 정의한 모델이기 때문에 이를 사용하지 않는 상황은 적절하지 않아 제외하였으며, PR과 PPR은 노드 특징을 사용하지 않는 모델이기 때문에 제외하였다.

귀납 학습 (3)번 실험의 결과는 표 5에 정리되어 있으며, 그림 3은 (3)과 (4)번 실험의 목표 데이터양에 따른 7가지를 그래프로 나타낸 것이다. 마지막으로 (5)와 (6)번 실험의 결과는 표 8에 정리되어 있다.

CISCO는 정답이 없는 실제 상황, 특히 통신 시나리오에서 귀납 학습을 통한 적용이 가능한지 알아보기 위해 도입하였다. 따라서 전도성 학습은 하지 않았으며,

표 4. 전도성 및 귀납 학습 실험 설정

Table 4. The details of transductive and inductive learning

		Train		Test (Target)
		Source	Target	
Transductive	(1)	-	O	O
	(2)	-	X	X
Inductive	(3)	O	O	O
	(4)	X	O	O
	(5)	O	X	X
	(6)	X	X	X

노드 특징이 정의되지 않았기 때문에 (5)와 (6)번 실험만 진행하였다. CISCO 실험 결과도 그림 3과 표 8에 같이 나와 있다.

4.2.3 데이터 집합 간의 유사성 및 노드 수 비교

귀납 학습에는 원본과 목표 데이터 집합이 서로 유사하다는 전제가 중요하다<sup>[14]</sup>. 따라서 사용한 네 데이터 집합 간의 유사성을 구조적, 의미적 두 가지로 비교하였다.

구조적 유사성은 그래프 구조 자체의 유사성을 의미한다. Graph2Vec<sup>[23]</sup>으로 128차원으로 임베딩한 그래프 간의 코사인 유사도 (Cosine Similarity)를 구조적 유사성으로 정의하였다.

의미적 유사성은 그래프의 노드로 사용된 단어들로 구성된 코퍼스의 유사성을 의미한다. 각각의 코퍼스로 훈련된 Word2Vec<sup>[24]</sup> 모델로 단어들을 128차원의 벡터로 임베딩 하고 전체 단어에 대해 임베딩 벡터의 평균을 각 데이터 집합의 임베딩 벡터로 사용하였다. 의미적 유사성 또한 이렇게 만든 임베딩 벡터 간의 코사인 유사도로 정의하였다. 단, CISCO에서는 각 노드가 익명의 기기를 지칭하기 때문에 코퍼스를 구성할 수 없어 의미적 유사도 계산에서 제외하였다. 각 유사도는 표 6에

표 5. GENI[7] 모델을 사용한 귀납 학습 결과와 전도성 학습과의 성능 차이

Table 5. The results of inductive learning on GENI[7] and differences from transductive learning

Dataset		NDCG@100		Diff. <sup>1</sup> (%)	SPEARMAN		Diff. <sup>1</sup> (%)
Source	Target	Transductive	Inductive		Transductive	Inductive	
FB15k[2]	TMDB[3]	0.9091	0.7373	18.90	0.7781	0.5112	34.30
	IMDB[4]	0.9526	0.7963	16.41	0.4960	0.3167	36.15
TMDB[3]	FB15k[2]	0.8890	0.8160	8.21	0.6472	0.3176	50.93
	IMDB[4]	0.9526	0.9199	3.43	0.4960	0.4731	4.62
IMDB[4]	FB15k[2]	0.8890	0.7747	12.86	0.6472	0.2427	62.50
	TMDB[3]	0.9001	0.7937	11.82	0.7781	0.5436	30.14

<sup>1</sup> Difference: (Transductive - Inductive) / Transductive

표 6. 사용한 데이터 집합 간의 구조적, 의미적 유사도  
Table 6. The topological and semantic similarity between datasets

T \ S	FB15k[2]	TMDB[3]	IMDB[4]
FB15k[2]	1	0.3265	0.0643
TMDB[3]	0.8668	1	0.2976
IMDB[4]	0.9694	0.7227	1
CISCO[5]	0.7809	0.6872	0.7562

나와 있다. 이상단 (S)는 의미적 유사도, 최하단 (T)는 구조적 유사도를 뜻한다.

### 4.3 실험 결과 및 분석

#### 4.3.1 전도성 학습

표 3은 전도성 학습, (1)과 (2)번 실험의 결과를 보여 준다. 굵은 글씨는 성능이 가장 좋은 결과를, 밑줄은 그다음으로 좋은 결과를 나타낸다. 한 가지 경우를 제외 하고 가장 많은 정보를 사용한 RGTN의 성능이 가장 우수하였고, 그다음이 GENI였다. 즉 노드 중요도 예측 과업에서는 많은 정보가 중요함을 알 수 있다.

하지만 노드 특징을 사용하지 않은 경우에도 성능 저하가 크지 않다. GENI 모델에서 보면 NDCG@100 성능에 대한 오차가 각각 8.61%, 6.47%, 3.84%밖에 차이 나지 않는다. 즉, 노드 중요도 예측 과업에서 노드 자체의 특징보다는 그래프의 구조, 노드가 어떻게 연결 되어 있는지가 중요함을 의미한다.

또한 PR과 PPR은 비록 NDCG@100 성능은 낮지만, SPM과 OVER@100 성능이 다른 모델보다 대체로 높다. 두 모델이 그래프 구조 자체에 집중한 모델인 만큼 전체적인 경향을 잘 예측했기 때문으로 보인다.

#### 4.3.2 데이터 집합 간의 유사도 및 노드 수 비교

이전의 연구에서 미세 조정의 성능에 데이터 집합 간 유사도와 크기 차이가 영향을 미치는 것이 확인됐다<sup>[11]</sup>. 따라서 귀납 학습 실험 결과를 분석하기 전에 유사도와 크기를 먼저 비교할 필요성이 있다. 표 6의 유사성을  $sim_{ij}^m$ 으로 나타낸다면 이들 간의 대소 관계는 식 (35~37)과 같다.

$$sim_{FI}^S > sim_{FT}^S > sim_{TI}^S \quad (35)$$

$$sim_F^S > sim_I^S > sim_T^S \quad (36)$$

$$sim_{FT}^C > sim_{TI}^C > sim_{FI}^C \quad (37)$$

표 7. 사용한 양에 따른 데이터 집합 별 노드 개수  
Table 7. The number of nodes in each dataset by each %

Data Used	FB15k[2]	TMDB[3]	IMDB[4]	CISCO[5]
5%	747	5,740	56,249	1,515
10%	1,495	11,480	112,499	3,030
20%	2,990	22,961	224,999	6,059
40%	5,980	45,922	449,998	12,118
80%	11,960	91,844	899,996	24,236
100%	14,951	114,805	1,124,995	30,295

여기서 아래 첨자  $F, T, I$ 는 각각 FB15k, TMDB, IMDB를 지칭하며 위 첨자  $s, c$ 는 각각 구조적, 의미적 유사성을 지칭한다. 즉 식 (35)와 식 (37)은 FB15k, TMDB, IMDB 간의 구조적, 의미적 유사성을, 식 (36)은 CISCO와 FB15k, TMDB, IMDB 간의 구조적 유사성의 대소 관계를 나타낸다.

또한 표 7을 보면 TMDB가 FB15k보다 7.7배, IMDB가 TMDB보다 9.8배, FB15k보다 75배 많은 노드를 갖고 있다. 더 많은 노드에서 사전 학습된 모델은 과적합 (Overfitting)되어 미세 조정이 어려울 것이다. 실제로 IMDB로 사전 학습한 모델은 다른 경우에 비해 미세 조정이 잘되지 않음을 확인하였다.

특히 CISCO는 노드 개수가 적은 편이다. 평균 개수도 FB15k 다음으로 적으며, 제일 적은 그래프는 86개 밖에 되지 않고 제일 많은 그래프도 68,724개로 TMDB보다 적다. 이에 CISCO로의 미세 조정은 오히려 학습을 하지 않았을 때보다 떨어지는 것을 확인하였다.

#### 4.3.3 귀납 학습

그림 3은 귀납 학습의 (3)과 (4)번 실험을 학습에 사용한 목표 데이터 집합의 양에 따른 성능을 나타낸다. x축은 목표 데이터 집합의 양, y축은 성능을 나타내며 실선은 각 목표 데이터 집합의 전도성 (1)번 실험 결과를 나타낸다. 범례의 윗줄은 사전 학습에 원본 데이터의 노드 특성을 사용한 경우, 아랫줄은 사용하지 않은 경우를 지칭한다.

먼저 그림 3.a는 목표 데이터가 w/FB15k일 때의 귀납 학습 결과이다. 목표 데이터를 쓰지 않았을 때 NDCG@100 성능은 TMDB에서가 IMDB보다, 그리고 w/에서가 w/o에서보다 더 높다. 구조적 유사성은 IMDB와 더 높지만, 노드 수 차이가 크고, 의미적 유사성이 TMDB와 더 높기 때문이다. 전체적인 경향성을 나타내는 SPM 성능은 IMDB에서 학습된 구조적 정보를 이용하여 원본 데이터가 IMDB일 때 항상 높음을

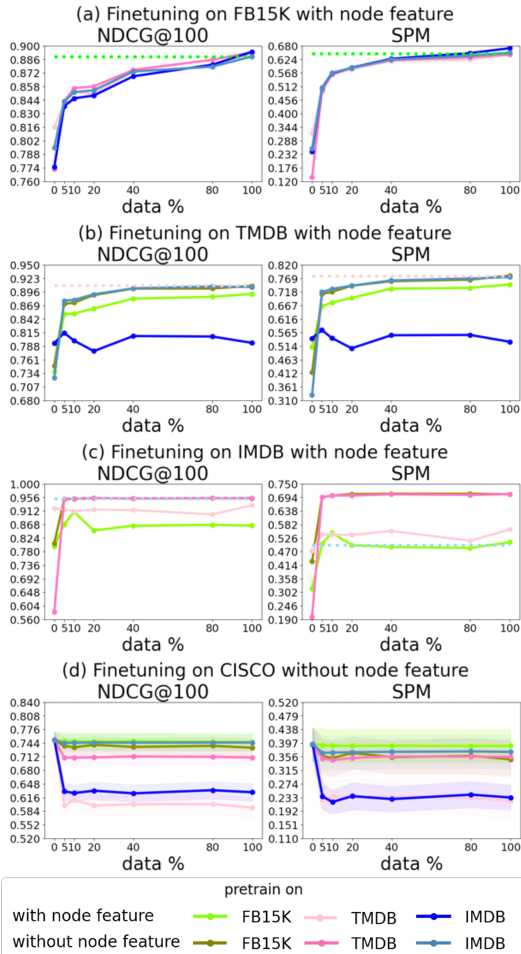


그림 3. GENI[7] 모델에서의 귀납 학습 결과  
Fig. 3. The results of inductive learning on GENI[7]

볼 수 있다. 목표 데이터를 모두 사용하여 노드 수 차이를 최대한 줄였을 때 NDCG@100 성능은 w/IMDB에서 가장 높다. 이는 IMDB의 구조적 유사성을 최대한 활용하였기 때문이다.

그림 3.b는 목표 데이터가 w/TMDB일 때의 귀납 학습 결과이다. 목표 데이터를 쓰지 않았을 때 NDCG@100 성능은 w/IMDB에서 가장 높다. FB15k와의 유사도가 더 높지만, 노드 수로부터 이점을 받았기 때문이다. 학습에 목표 데이터를 사용하게 되면서 구조적인 정보가 더 의존하는 것을 볼 수 있으며, 목표 데이터를 모두 사용했을 땐 FB15k로부터의 구조적 정보를 더 잘 활용하게 되었다.

마지막으로 그림 3.c는 목표 데이터가 w/IMDB일 때의 귀납 학습 결과이다. 그림 3.a와는 반대로 FB15k는 노드가 너무 적기 때문에 좀 더 노드가 많은 w/TMDB

에서의 NDCG@100 성능이 더 높다. 하지만 그림 3.a와 같이 SPM 성능은 FB15k의 구조적 정보에서 이점을 받아 전체적으로 높음을 볼 수 있다.

위 결론들을 토대로 귀납 학습은 그래프의 의미적 유사성보다는 구조적 유사성에 좀 더 의존적이며, 노드 수 또한 중요함을 알 수 있다.

한편 표 8은 귀납 학습의 (5)와 (6)번 실험 결과를 나타낸다. CISCO 데이터 집합은 전도성 학습을 진행하지 않았기 때문에, 원본 데이터로는 사용되지 않았으며, 목표 데이터로 사용된 경우는 그림 3.d와 같은 결과이다. 표시된 숫자는 학습에 사용한 목표 데이터양 7가지와 원본 데이터의 노드 특성 사용 여부 2가지, 총 14가지에서의 실험 결과에 대한 평균±분산을 나타낸다. CISCO 데이터 집합을 제외한 나머지 세 데이터 집합에서는 원본 데이터의 노드 특징 사용 유무 및 목표 데이터로 추가 학습을 하는 것과 관계없이 성능이 일정함을 확인했다. 이는 그래프 구조가 비슷하다면, 목표 데이터에서 귀납 학습을 통해 준수한 성능을 얻을 수 있음을 보여주는 결과이다.

해당 결과는 큰 이점을 가진다. 하나의 그래프에서 학습된 모델을 이용해서 다른 그래프에 쉽게 적용할 수 있기 때문에, 그래프 자체가 변하거나 여러 종류의 그래프를 포함하고 있는 네트워크를 사용하는 통신 분야에 특히 유용하게 적용될 수 있을 것으로 보인다.

#### 4.3.4 통신 시나리오 응용

본 논문에서는 네트워크 그래프인 CISCO 데이터 집합을 이용하여 유용성을 확인하였다. 먼저 그림 3.d에서 귀납 학습에 사용한 CISCO 데이터 집합의 양에 따른 (5)와 (6)번 실험에 대한 결과를 확인할 수 있다. 다른 데이터 집합과 달리, CISCO는 22개의 그래프에 대

표 8. GENI[7] 모델에서 특성을 사용하지 않았을 때의 모든 귀납 학습 결과  
Table 8. The results of inductive learning without node feature of target dataset

Dataset		NDCG@100	SPEARMAN
Target	Source		
FB15k[2]	TMDB[3]	0.8125±0.0001	0.3219±0.0004
	IMDB[4]	0.8125±0.0001	0.3218±0.0004
TMDB[3]	FB15k[2]	0.8505±0.0002	0.6858±0.0007
	IMDB[4]	0.8504±0.0002	0.6856±0.0008
IMDB[4]	FB15k[2]	0.9157±0.0002	0.4363±0.0027
	TMDB[3]	0.9159±0.0002	0.4343±0.0032
CISCO[5]	FB15k[2]	0.7381±0.0056	0.3616±0.0142
	TMDB[3]	0.7164±0.0143	0.3585±0.0146
	IMDB[4]	0.7456±0.0021	0.3730±0.0086

한 평균값을 보여주고 있기 때문에  $\pm 0.1\sigma$ 에 해당하는 부분을 음영으로 표시하였다.

대체로 노드 특징을 사용한 FB15k에서 사전 학습했을 때의 성능이 가장 높으며, 노드 특징을 사용한 TMDB에서 사전 학습했을 때의 성능이 가장 낮은 것을 볼 수 있다. 또한 노드 특징을 사용하지 않은 경우 IMDB, FB15k, TMDB 순으로 성능이 높은 것을 볼 수 있는데 이는 4.3.3에서 확인한 것처럼 그래프 구조와 노드 개수 차이에 영향을 받은 결과이다. 식 (36)에서 확인할 수 있듯, CISCO는 구조적으로 FB15k와 제일 유사하며, 그다음 IMDB, TMDB 순으로 유사하다. 또한 표 6에서 보듯 노드 수는 FB15k 다음으로 적다.

먼저 노드 특징을 활용한 경우 FB15k에서가 제일 높고, IMDB와 TMDB 순으로 높다. 이는 FB15k보다 CISCO의 노드가 많아 FB15k에서 학습한 정보를 잘 활용하여 CISCO에 맞게 미세 조정이 되었지만 다른 두 데이터 집합에서는 그러지 못한 것이다. 한편 노드 개수가 가장 많은 IMDB에서 성능이 제일 낮지 않은 것은 구조적 유사성에서 이점을 가져왔기 때문이다.

노드 특징을 활용하지 않은 경우에도 구조적 유사성과 성능이 높은 순서 간의 관계를 찾을 수 있다.

## V. 요약 및 결론

노드 중요도 예측 과업은 그래프 데이터를 분석하는데 중요한 과업이며, 여러 서비스에서 제품을 추천할 때 지식 그래프에서 질의 검색 시 우선순위를 결정하는데 활용되고 있다. 본 논문은 이런 노드 중요도 예측 과업을 위한 모델들을 모델 구조 및 사용하는 데이터 범위에 따라 구분하였다. 또한 GENI에서 귀납 학습을 진행한 결과 원본 데이터와 목표 데이터가 구조적으로 유사할 때 더 좋은 성능을 보임을 확인하였고, 이에 더해 노드 수 또한 중요함을 확인하였다.

지도 학습 기반의 모델만 제안된 노드 중요도 예측 과업은 실제로 비지도 학습이 가정된다. 하지만 본 논문을 통해 확인한 전이 학습 및 귀납적 테스트의 가능성은 이런 차이를 보완해 줄 것으로 기대된다. 본 논문에서는 이를 통신 분야 데이터 집합인 CISCO를 통해 실제로 유용함을 입증하였으며, 이뿐 아니라 다른 데이터 집합에도 적용될 수 있을 것이다.

## References

[1] J. Lee and B. Ko, "Knowledge graph-based music recommender system using lyrics

keyword extraction," *J. Korean Data And Inf. Sci. Soc.*, vol. 33, no. 6, pp. 937-949, Nov. 2022.

[2] A. Bordes, et al., "Translating embeddings for modeling multi-relational data," *Advances in NIPS*, vol. 26, Stateline, USA, Dec. 2013.

[3] The Movie Database, *TMDB Movie*, <https://www.kaggle.com/tmdb/tmdb-movie-metadata>.

[4] Amazon.com, *IMDb Datasets*, <https://www.imdb.com/interfaces>

[5] O. Madani, et al., "A dataset of networks of computing hosts," in *Proc. 2022 ACM Int. Wkshp. Secur. and Privacy Analytics*, pp. 100-104, Baltimore, USA, Apr. 2022. (<https://doi.org/10.1145/3510548.3519368>)

[6] K. Bollacker, et al., "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proc. 2008 ACM SIGMOD Int. Conf. Manag. Data*, pp. 1247-1250, Vancouver, Canada, Jun. 2008. (<https://doi.org/10.1145/1376616.1376746>)

[7] N. Park, et al., "Estimating node importance in knowledge graphs using graph neural networks," in *Proc. 25th ACM SIGKDD Conf. Knowledge Discovery & Data Mining*, pp. 596-606, Anchorage AK, USA, Jul. 2019. (<https://doi.org/10.1145/3292500.3330855>)

[8] F. Scarselli, et al., "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61-80, Jan. 2009. (<https://doi.org/10.1109/TNN.2008.2005605>)

[9] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer networks and ISDN systems*, vol. 30, no. 1, pp. 107-117, Apr. 1998.

[10] H. Huang, et al., "Representation learning on knowledge graphs for node importance estimation," in *Proc. 27th ACM SIGKDD Conf. Knowledge Discovery & Data Mining*, pp. 646-655, Virtual Event, Singapore, Aug. 2021. (<https://doi.org/10.1145/3447548.3467342>)

[11] C. Raffel, et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *The J. Mach. Learn. Res.*, vol.

21, no. 1, pp. 5485-5551, Jun. 2020.

[12] J. Gasteiger, et al., "Predict then propagate: Graph neural networks meet personalized pagerank," *arXiv preprint arXiv:1810.05997*, 2018.

[13] M. Schlichtkrull, et al., "Modeling relational data with graph convolutional networks," in *The Semantic Web: 15th Int. Conf., ESWC 2018*, pp. 593-607, Heraklion, Crete, Greece, Jun. 2018.  
([https://doi.org/10.1007/978-3-319-93417-4\\_38](https://doi.org/10.1007/978-3-319-93417-4_38))

[14] S. Vashishth, et al., "Composition-based multi-relational graph convolutional networks," *arXiv preprint arXiv:1911.03082*, 2019.

[15] P. Veličković, et al., "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[16] J. K. Chorowski, et al., "Attention-based models for speech recognition," *Advances in NIPS*, vol. 28, Montreal, Canada, 2015.

[17] A. Vaswani, et al., "Attention is all you need," *Advances in NIPS*, vol. 30, Long Beach, USA, 2017.

[18] A. Dosovitskiy, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[19] A. Grover and J. Leskovec, "Node2Vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pp. 855-864, San Francisco, USA, Aug. 2016.  
(<https://doi.org/10.1145/2939672.2939754>)

[20] Z. Dai, et al., "Transformer-XL: Attentive language models beyond a fixed-length context," *arXiv preprint arXiv:1901.02860*, 2019.  
(<https://doi.org/10.48550/arXiv.1901.02860>)

[21] A. Hagberg, et al., "Exploring network structure, dynamics, and function using NetworkX," in *Proc. 7th SciPy2008*, pp. 11-15, Pasadena, CA USA, Aug. 2008.

[22] M. Wang, et al., "Deep graph library: A graph-centric, highly-performant package for graph neural networks," *arXiv preprint*

*arXiv:1909.01315*, 2019.

[23] A. Narayanan, et al., "Graph2Vec: Learning distributed representations of graphs," *arXiv preprint arXiv:1707.05005*, 2017.  
(<https://doi.org/10.48550/arXiv.1707.05005>)

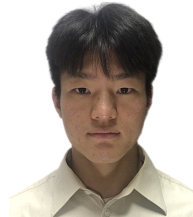
[24] K. W. Church, "Word2Vec," *Natural Lang. Eng.*, vol. 23, no. 1, pp. 155-162, Dec. 2016.  
(<https://doi.org/10.1017/S1351324916000334>)

조 우 성 (Woo-seong Cho)



2021년 2월: 연세대학교 화공생명공학과 졸업  
2022년 9월~현재: 국민대학교 컴퓨터공학과 석사과정  
<관심분야> 그래프 신경망, 노드 중요도 예측, 자기 지도 학습

송 승 현 (Seugn-heon Song)



2023년 2월: 국민대학교 나노전자물리학과 졸업  
2023년 3월~현재: 국민대학교 컴퓨터공학과 석사과정  
<관심분야> 멀티모달, 조밀 예측 (Dense Prediction)

이 재 구 (Jae-koo Lee)



2011~2013년: LG전자 CTO부 문 주임 연구원  
2018년: 서울대학교 전기컴퓨터공학부 박사 졸업  
2018년: SK 텔레콤 DATA 기술원 매니저  
현재: 국민대학교 소프트웨어융합대학 인공지능학부 부교수, 인공지능 (<http://mi.kookmin.ac.kr/>) 연구실 PI  
<관심분야> 인공지능, 기계학습, 심층학습 및 데이터 과학, 자율 주행, 객체 인식, 멀티모달