EXPLAINABLE AUTOMATIC HYPOTHESIS GENERATION VIA HIGH-ORDER GRAPH WALKS

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we study the automatic hypothesis generation (HG) problem, focusing on explainability. Given pairs of biomedical terms, we focus on link prediction to explain how the prediction was made. This more transparent process encourages trust in the biomedical community for automatic hypothesis generation systems. We use a reinforcement learning strategy to formulate the HG problem as a guided node pair embedding-based link prediction problem via a directed graph walk. Given nodes in a node pair, the model starts a graph walk, simultaneously aggregating information from the visited nodes and their neighbors for an improved node pair representation. Then at the end of the walk, it infers the probability of a link from the gathered information. This guided walk framework allows for explainability via the walk trajectory information. By evaluating our model on predicting the links between millions of biomedical terms in both transductive and inductive settings, we verified the effectiveness of our proposed model on obtaining higher prediction accuracy than baselines and understanding the reason for a link prediction.

1 INTRODUCTION

The exponential growth in the number of scientific research publications (Björk et al., 2008; Larsen & Von Ins, 2010) has made the manual process of extracting the connection between biomedical terms (e.g., diseases, chemicals, drugs, and genes) a laborious task. For an increased efficiency, computers have been introduced to the hypothesis generation pipeline for scientific research and have gained popularity in the last decade (Baek et al., 2017; Akujuobi et al., 2020b; Jha et al., 2019; Xun et al., 2017; Spangler, 2015; Akujuobi et al., 2020a). With the increasing interest in automatic hypothesis generation (HG), several systems have been proposed. Smalheiser & Swanson (1998); Swanson & Smalheiser (1997); Hristovski et al. (2006) utilize a simple A-B-C approach based on a closed graph triad principle for inferring connections between two terms. Although a logical and straightforward approach, it fails to capture complex relationships between terms. To mitigate the limitations of the A-B-C approach, several studies have applied machine learning techniques for automatic HG. These techniques include text mining (Spangler et al., 2014; Spangler, 2015), association rules (Hristovski et al., 2006; Gopalakrishnan et al., 2016; Weissenborn et al., 2015), classification and clustering (Sybrandt & Safro, 2018; Srihari et al., 2007; Baek et al., 2017), and others (Jha et al., 2019; Xun et al., 2017; Shi et al., 2015; Akujuobi et al., 2020b).

However, with the growing discussion on the trustworthiness of machine learning models (Chatzimparmpas et al., 2020; Varshney, 2019; Toreini et al., 2020; Samek et al., 2020), especially in the biomedical field (Yuan et al., 2021; Michael et al., 2018), it is not enough to only predict yes or no about the connectivity of term pairs. There is a need for transparent and explainable models that also can provide insights on how the prediction was made. Understanding what drives a prediction is essential for determining targeted practical study (Lundberg et al., 2017). To the best of our knowledge, we are the first to focus on the study of explainable automatic hypothesis generation.

Definition 1. Walk-based explainable hypothesis generation: *Given an unlinked pair of nodes $< u, v >$ and a graph $G = \{V, E\}$, the target is to infer the probability of a link between u and v (a hypothesis is generated between these nodes) via a guided graph walk, such that the trajectories explains the link prediction.*

Our proposed approach is named *Walk-based Explainable Hypothesis Generation* (WEHG). To infer the connectivity of the terms $\{u, v\}$, WEHG initializes a graph walk of length T on the graph G

starting from u and v respectively. At each step $t; t = 1 \cdots T$ of the walk, WEHG learns an embedding that integrates together the information obtained from previous walk steps, the current node attribute, and node attributes of neighboring nodes at time t. If well-guided rather than walking randomly, WEHG moves purposefully from one node to another to better understand the meaning of the starting node u or v. At the terminal step, WEHG embeds its understanding of the nodes by a vector $h_{u,v}^T$ for link prediction. To make an accurate prediction, WEHG is expected to have an appropriate policy to determine the walk sequence, which provides a semantically meaningful embedding vector for the node pair and explains the prediction of a link. We design WEHG as a decision-making agent whose policy is learned by a reinforcement learning-based attention mechanism.

In summary, our contributions are:

1.) We structure the hypothesis generation task as a node pair embedding task via graph walk on attributed graphs. This framework facilitates the direct encoding of the intrinsic information from the nodes' neighborhood, thereby leading to improved predictive performance and explanation.

2.) We design WEHG to learn the graph walk policy in a reinforcement learning-based attention mechanism. WEHG is flexible to be used for any edge (node pair) embedding task on attributed graphs, although we apply it for hypothesis generation in this work. Meanwhile, WEHG has an advantage to be usable in both inductive and transductive settings.

2 Related Work

2.1 Hypothesis Generation

Ideas, research results, and knowledge are often presented to the public in scientific literature. To identify unnoticed but discoverable connections via automated systems, Smalheiser & Swanson (1998) proposed the A-B-C model based on closed entity linkages for knowledge discovery. This strategy has been further developed by Hristovski et al. (2006); Weissenborn et al. (2015); Gopalakrishnan et al. (2016) to mitigate the limitations of the simple A-B-C model such as it's high dependence on just the entity connections. Machine learning techniques such as Latent Dirichlet Allocation (LDA), clustering, and text mining have also been introduced to the process of building automatic hypothesis generation systems (Sybrandt & Safro, 2018; Srihari et al., 2007; Akujuobi et al., 2020b;a; Jha et al., 2019; Xun et al., 2017; Srinivasan & Libbus, 2004; Spangler et al., 2014). In this paper, we take a different approach in the automated HG problem formulation using guided walks on graphs. We focus on introducing flexibility and explainability in the framework by exploring the information space to better capture essential information for node pair relationships in transductive and inductive settings.

2.2 Network Embedding

The area of network embedding has gained significant attention in the recent decade. Several Graph Neural Networks (GNN) have been proposed for node/graph embedding (Kipf & Welling, 2016; Perozzi et al., 2014; Grover & Leskovec, 2016; Yang et al., 2016; Hamilton et al., 2017). Graph-based hypothesis generation task can be formulated as an edge (node pair) embedding task. A simple solution would be joining the embedding of two nodes linked by one edge (Perozzi et al., 2014; Grover & Leskovec, 2016). However, its simplicity has a disadvantage as it fails to simultaneously capture complex information from its structural and semantic attributes. Hence its performance is dependent on the quality of the separately learned attributes. Some other complex models embed node pairs and their context in the same space (Zhao et al., 2016), or learn node and edge embedding vectors jointly (Gao et al., 2019). This node pair embedding problem can also be reduced to a knowledge graph task of jointly learning the embedding of a (relation) edge joining a pair of nodes (head and tail) (Lin et al., 2015; Ji et al., 2016; Shi & Weninger, 2017; Wang et al., 2014; Bordes et al., 2013). However, these methods focus on independent node representation learning and lack explainability of the model's predictive performance.

Recent works have structured graph problems as reinforcement learning tasks. Lee et al. (2018) proposed using a graph walk for multi-class graph classification, through attention on the graph structural composition. This proposed method (GAM) is based on recurrent walks with random restarts and assumes that the nodes are tagged (attached labels), hence evaluates graph label prediction per step iteration. The assumptions and learning method are not feasible in our problem setting as we

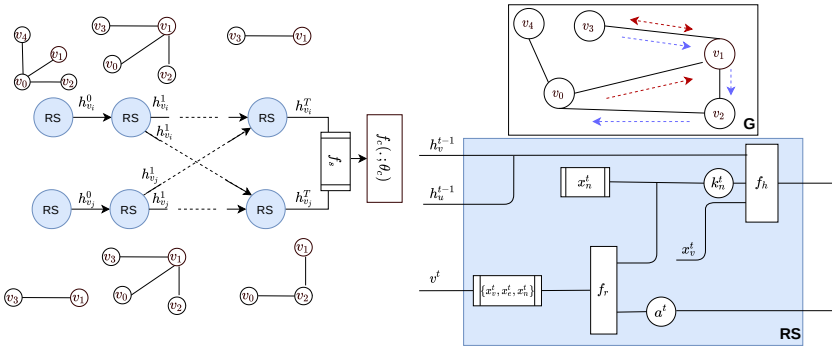

Figure 1: Pipeline of our WEHG method. The example graph given in G shows the walking process for inferring the relationship between $\{v_0, v_3\}$. The red arrows indicate the walking $v_0 \rightarrow v_1 \rightarrow v_3 \rightarrow v_1$, while the blue arrows show the walking $v_3 \rightarrow v_1 \rightarrow v_2 \rightarrow v_0$.

do not have such information and differ in task. The aforementioned method also lacks explainability as it focus on an uninterpretable latent semantic vector space. Reinforcement learning has also found application in explainable systems especially in explainable recommendation (Xian et al., 2020; 2019; Fu et al., 2020), explainable conversation systems (Yang et al., 2020; Moon et al., 2019) and explainable KG reasoning (Bhowmik & de Melo, 2020). However, these works are specifically tailored to their respective task and not applicable to node pair(/edge) embedding problems.

3 METHODOLOGY

3.1 MODEL OVERVIEW

The proposed model makes node pair relationship predictions based on information aggregated from walks on a given graph. In this framework, information is modeled as a sequential process of moving from one graph node to another, and the choice of which node to visit is determined with the goal to efficiently explore the neighborhood to provide essential information for the node pair relation prediction. Specifically, given a node pair $S_{u,v} = \{u, v\}$, the model starts a walk from each node in the pair (i.e., u and v) respectively. At each step of the walk, the model decides which node from the observed neighborhood to move to next. We define the observed neighborhood as a sample of one-hop neighbors of the current node v_t. At the end of the graph walk process, the final context representation $h^T_{u,v}$ is then used to conduct node pair connectivity prediction. The pipeline of the walking process in WEHG is illustrated in Fig. 1, and the inference process is given in Algo. 1.

WEHG is a recurrent model composed of three main neural network blocks. Two networks in the recurrent step (RS) section are $f_h(.;\theta_h)$ for updating the learned context during the graph walks and $f_r(.;\theta_r)$ for recurrently estimating the relevance of the node neighbors during the sequential graph walks. At the end of the graph walk, the prediction network $f_c(.;\theta_c)$ is used to infer the node pair connection. An example of the graph G snippet presented in Fig. 1 is explained as follows: Given a node pair $\{u = v_0, v = v_3\}$, the model runs a simultaneous graph walk for both nodes in the pair. At step t, the agent decides the next node to visit at time $t + 1$ by evaluating the importance of the nodes in the observed neighborhood at step t using the relevance network $f_r(.;\theta_r)$. This process takes as input the previous aggregated information $h^{t-1}_{u,v} = f(h^{t-1}_u, h^{t-1}_v)$, the node attribute x^t_v, and the attributes of the immediate node neighbors x^t_n and edges x^t_e. f is an aggregation function. The relevance network then generates a score for each observed node neighbor in the range of 0 to 1. Since the score indicates the relevance of a node neighbor to the given node, the next node v^{t+1} to visit is sampled from its neighbors in proportion to their relevance scores. Along with the previous walk context representation $h^{t-1}_{u,v}$, the current node embedding x^t_v and the relevance scores from $f_r(.;\theta_r)$ are also used in the updating of context representation in the form of the weighted neighborhood aggregation k^t_n (see section 3.2 for more details). The context representation h^t_v is updated until $t = T$, for v_0 following the walk sequence $v_0 \rightarrow v_1 \rightarrow v_3 \rightarrow v_1$, and for v_3 following the walk sequence $v_3 \rightarrow v_1 \rightarrow v_2 \rightarrow v_0$.

The final node pair representation vector $h^T_{v,u}$ summarizing the respective information obtained from the graph walk is obtained as $h^T_{u,v} = f_s(h^T_u, h^T_v)$, where $f_s(.)$ can be a simple linear aggregation function (such as addition) or a more complex neural network function. The aggregated graph walk information $h^T_{u,v}$ is then passed to the classification network $f_c(.; \theta_c)$ modeled as a single-layer neural network, to predict the node pair connection. It is worth mentioning that WEHG is also applicable in an inductive setting, where node pair relationships can be predicted among new nodes.

3.2 Formulation as Markov Decision Process

As discussed in section 3.1, WEHG aggregates information from walks on the graph. At each step, the model updates the aggregated information and decides on the next graph step. The sequential process in WEHG can be framed as a Markov Decision Process (MDP), which provides a coherently appropriate solution to the sequential decision-making problem by providing the most beneficial actions for potential gains in the future based on a current environment observation at time t. However, the sampled one-hop neighborhood observation at each graph walk step presents a barrier in a direct MDP setup. This barrier is due to the introduced noise in the sampled one-hop observation and the inability of the observation to capture the complete topological environment in a large graph. Partially Observable MDP (POMDP) provides a medium to sidestep the limitations mentioned above for partially observable environments. Hence, our problem formulation fits a POMDP setup.

To address the incompleteness of observations, we augment the current observation by integrating the information from the previous steps in the walk trajectory. This information is encoded recurrently by Gated Recurrent Unit (GRU) (Cho et al., 2014) to capture the recurrent dependency in the graph walk path on the graph. For v at step t, we update h^t_v by $h^t_v = f_h(h^{t-1}_{u,v}, x^t_v, k^t_n; \theta_h)$, which has GRU at its core and is formulated as :

$$\begin{aligned} z^t &= \sigma_g(W^z[x^t_v + k^t_n] + U^z h^{t-1}_{u,v} + b^z), \\ r^t &= \sigma_g(W^r[x^t_v + k^t_n] + U^r h^{t-1}_{u,v} + b^r), \\ \hat{h}^t &= \sigma_{\hat{h}}(W[x^t_v + k^t_n] + r^t \circ U h^{t-1}_{u,v} + b), \\ h^t_v &= z^t \circ \hat{h}^t + (1 - z^t) \circ h^{t-1}_{u,v}. \end{aligned} \tag{1}$$

where x^t_v is the current node attribute of node v, k^t_n is the weighted aggregate of the current one-hop node neighbor of v (see Eq. (2)), $\circ$ and $+$ denote element-wise multiplication and addition, respectively. The parameters W^z, W^r, W, U^z, U^r, and U are learnable weights, which are all presented as θ_h, including the bias vectors b^z, b^r, b as well. This framework allows for message passing between the simultaneous graph walks for u and v via memory sharing between the simultaneous graph walks.

At the end of each walk ($t = T$), $f_h(.; \theta_h)$ produces the embedding of the full trajectory started from each node in the node pair: h^T_u and h^T_v respectively. To predict the node pair connectivity, the node pair is represented as an aggregation $h^T_{u,v} = f_s(h^T_u, h^T_v)$. Then $h^T_{u,v}$ is given to the prediction network $f_c(.; \theta_c)$ for link prediction.

Actions to take in walking. At each step t, we model the move to the next node as an action a^t based on the output of the relevance network, $\varphi^t_v = f_r(h^{t-1}_{u,v}, x^t_v, x^t_e, x^t_n; \theta_r)$, where given a node v in a node pair, u denotes the other node in the pair. The relevance score φ^t_v expresses the relevancy of the one-hop neighbor nodes to the current node v^t. It is thus used for the next-node selection and also for the recurrent neighborhood aggregation update. We model the relevance network using a sigmoid activation function which places the values of φ^t between 0 and 1 for each neighboring node. For the sake of better exploration, a stochastic policy π is adopted to make the choice of the next node v^{t+1} to visit via sampling under the categorical distribution $P = Cat(.|\varphi^t)$.Based on the relevance weighting, the aggregation of relevant neighboring nodes is conducted as:

$$k^t_n = \sum_v x_v \times \varphi^t_v \quad v \in N(v^t), \tag{2}$$

where $N(v^t)$ is the set of sampled nodes in the one-hop neighborhood of the current node v^t. The aggregation of neighbors in Eq.(2) is a weighted sum of the neighbors according to their relevance.

Rewards to the actions. We adopt a delayed reward system, where the agent receives a reward only at the end of the graph walk. Specifically, for a labeled pair, the agent gets a reward $r_t = 1$ at the last step T, if the link prediction between the nodes in the pair at the end ($t = T$) is correct and

$r_t = -1$ otherwise. The agent gets no reward or penalty ($r_t = 0$) for unlabeled pairs. This reward framework allows freedom in the unlabeled node pairs' inference, as an unlabeled pair can be positive or negative.

3.2.1 POLICY TRAINING

The eventual outcome of reinforcement learning is the policy mapping from the observation space to the action space, $\pi(a^t|\beta_{1:t};\theta)$. In our setting, the variable $\beta_{1:t}$ represents the partial observations along the walk trajectory until step t, and is modeled as the augmented observation $\{h_{u,v}^{t-1}, x_n^t, x_v^t, x_n^v\}$. Parameters θ_r and θ_h are both policy relevant, because $f_h(.;\theta_h)$ provides $f_r(.;\theta_r)$ with the context embedding $h_{u,v}^{t-1}$ to determine the next action, and $f_r(.;\theta_r)$ sends k_n^t to $f_h(.;\theta_r)$ updating h_v^t with the relevant neighbors. The parameters $\theta = \{\theta_r, \theta_h\}$ are thus updated in training jointly to maximize the reward accumulation.

Algorithm 1: Calculate the probability of connection between nodes in node pair $S_{u,v} = \{u, v\}$. u denotes the opposite node (i.e., $u = v_j$ if $v = v_i$ and vice versa)

Input: Graph G, a node pair $S_{u,v} = \{u, v\}$, initial vector for each node h_u^0 and h_v^0 (vectors of zeros), node and edge embeddings $\{x, x_e\}$
Result: $p_{i,j}$: the connectivity probability score for the node pair $S_{i,j}$

```
for t ← 1 ··· T do
    for each node v ∈ S_{u,v} do
        Obtain the embeddings x_v^t of the current node v^t; x_e^t
          for edges connecting to v^t; and x_n^t for neighboring
          nodes;
        Calculate the relevance score to each neighbor
          φ_v^t = f_r(h_{u,v}^{t-1}, x_v^t, x_e^t, x_n^t; θ_r);
        Sample next node v^{t+1} to visit from a categorical
          distribution Cat(.|φ^t) over the relevant neighbors;
        Extract the relevant neighbor information k_n^t;
        update the history vector
          h_v^t = f_h(h_v^{t-1}, h_u^{t-1}, x_v^t, k_n^t; θ_h) ;
    end
end
Aggregate the learned embeddings h_{u,v}^T = f_s(h_u^T, h_v^T);
Return p_{u,v} = f_c(h_{u,v}^T; θ_c)
```

However, computing the policy objective: $\mathcal{L}_R(\theta) = \mathbb{E}_{(\pi;\theta)}\big[\sum_{t=1}^{T} r_t\big]$ is hard in practice (Lee et al., 2018) due to the difficulty of calculating the expectation over joint probability distribution of walk paths. Hence, using the trick of log derivative to change the gradient of the expectation to the expectation of the gradient, the REINFORCE algorithm for POMDP Williams (1992) could take gradients of the objective as following:

$$\begin{aligned}\nabla_\theta \mathcal{L}_R &= \sum_{t=1}^{T} \mathbb{E}_{p(\beta_{1:T};\theta)}[\nabla_\theta \log \pi(a_t|\beta_{1:T};\theta) R] \\ &\approx \frac{1}{M} \sum_{i=1}^{M} \sum_{t=1}^{T-1} \nabla_\theta \log \pi(a_t^i|\beta_{1:t}^i;\theta) \gamma^{T-t} R^i. \end{aligned} \tag{3}$$

The variable $\gamma \in (0, 1]$ is a discount factor giving more preference to actions performed closer to the last step (i.e., $t = T$). The roll-out sequences β^i's are obtained from running the agent π_θ for $i = 1, ..., M$ episodes; and R^i is the reward to go of the episode i.

The gradient estimate of reward observation and action can be obtained by sampling several roll-outs, each running the agent for T walk steps. This is made possible by breaking down the joint distribution of the trajectory. This process provides feedback to the policy and guides for an enriched and better parameter space exploration.

The differentiable relevance network f_r and recurrent network f_h, represented as neural networks, are updated based on the backpropagated policy gradient information. Hence, any policy gradients corresponding to high rewards are higher weighted, making roll-outs with higher rewards more likely. However, high variance from sampling still exists, though the estimate is an unbiased one. The reward setup alleviates this problem in sampled trajectories by reducing the reward collected at the intermediate steps of roll-outs. It may, however, suffer from high variance due to sampling and may cause the policy to be unstable. A baseline could be used to reduce the magnitude of the estimate, resulting in reduced variance and an unbiased estimate. The classification network is trained in supervised using a binary cross-entropy loss while the relevance network is trained using REINFORCE. The whole model is trained end-to-end.

Table 1: Statistics of evaluation node pairs. The positive pairs are those with confirmed true connections, while the unknown pairs are those without confirmed true connections.

	#Nodes	Inductive		Transductive		Unfiltered	
		#Positive pairs	#Unknown pairs	#Positive pairs	#Unknown pairs	#Positive pairs	#Unknown pairs
Virology	42,870	35,363	729,256	488,711	3,167,809	524,074	3,897,065
COVID-19	65,275	174,708	2,232,254	3,132,283	3,703,709	3,306,991	5,935,963
Cancer	64,083	44,901	822,196	893,556	5,005,188	938,437	5,827,384

4 EXPERIMENTS

4.1 DATASETS AND EXPERIMENTAL SETUP

We construct evaluation graph datasets from the title and abstract of papers published in the biomedical fields from 1949 to early 2021. The nodes of the graphs correspond to the scientific terms found in the papers, while the edges linking two nodes indicate the co-occurrence in a sentence or title. Our study focuses only on the co-occurrence relation and leaves the relationships' multi-dimensional properties for future study. The three graph datasets in evaluation are constructed from papers in popular biomedical research areas, namely *COVID-19*, *Cancer*, and *Virology*.

To set up the training and testing data, we split the graph by publication year. The connections made in papers published till 2015 are used for training, while the model is tested on connections made in papers published from 2016 to 2021. Therefore in testing, the *positive* pairs, i.e., those linked in papers published in the time window [2016 - 2021] can exist as links : 1) between nodes already existing in the training graph G, i.e., $\{v_i, v_j\}; v_i \in G, v_j \in G$; 2) between a new term and an existing term, i.e., $\{v_i, v_j\}; v_i \notin G, v_j \in G$; or 3) between two new nodes $\{v_i, v_j\}; v_i \notin G, v_j \notin G$. Unlinked node pairs in G are considered as *unlabeled*. The statistics of the datasets used for evaluation are shown in Table 1.

The experiments were conducted on Amazon AWS Linux system using Python[1]. We extract term descriptions and convert them to a 200-dimensional feature vector by applying the latent semantic indexing (LSI) method to obtain the node feature vector x_v^t. The sentences where two terms co-occur are used as the edge attributes. The edge feature vectors are also obtained using the LSI technique. Our method is implemented using the Tensorflow library. Each GPU-based experiment was conducted on a single NVIDIA T4 Tensor Core GPU. For evaluations reported in this paper, proposed models are trained for 5 epochs with a parameter set ($d = 128, T = 5, M = 3, lr = 5e^{-3}$). Parameter search was performed for the proposed model and baseline methods. We assume the graph to be undirected.

4.2 QUANTITATIVE STUDY

4.2.1 COMPARISON METHODS

To evaluate the performance of our proposed model, we compare it against several state-of-the-art (SOTA) methods, which have publicly available implementation codes, to avoid unfair evaluations due to potential faulty implementation. These baseline methods include:

Embedding Based Methods. This group of methods extract node embedding from models trained in unsupervised ways: text embedding (using LSI), Node2Vec, and GraphSAGE. For the TextGraph feature, we concatenate the text and Node2Vec features. The node embeddings in each pair are then concatenated and fed through a logistic regression layer (a single layer perceptron) for link prediction.

Signed Networks. These are SOTA methods for link prediction using signed networks. We mark the observed pairs with positive edges and the non-observed pairs with negative edges in this setup. The methods evaluated are nSNE (Song et al., 2018) and SGCN (Derr et al., 2018).

tNodeEmbed (Singer et al., 2019). Leverages the temporal information in graphs to create rich node representations. We modified the original model to handle unseen nodes in inductive learning.

TransR (Lin et al., 2015), TransH (Wang et al., 2014), TransE (Bordes et al., 2013). These are knowledge graph completion methods, which are modified for this task by conducting triplet {head,tail,relationship} classification.

[1]The implementation codes and all datasets will be available in public.

Table 2: Evaluation results showing the precision-recall AUC performance on the COVID-19, Virology and Cancer datasets, respectively

	Virology			Cancer			COVID-19		
	Inductive	Transductive	Unfiltered	Inductive	Transductive	Unfiltered	Inductive	Transductive	Unfiltered
Text	0.10	0.37	0.34	0.13	0.42	0.40	0.15	0.73	0.68
GraphSAGE	0.15	0.58	0.55	0.15	0.60	0.58	0.18	0.83	0.81
Node2Vec	_	0.77	0.73	_	0.84	0.81	_	0.92	0.9
TextGraph	_	0.78	0.73	_	0.84	0.81	_	0.93	0.90
tNodeEmbed	_	0.65	0.61	_	0.72	0.69	_	0.86	0.81
TRANSE	_	0.62	0.61	_	0.81	0.81	_	0.72	0.65
TRANSR	_	0.67	0.64	_	0.80	0.79	_	0.69	0.64
TRANSH	_	0.44	0.44	_	0.75	0.65	_	0.67	0.61
SGCN	0.14	0.45	0.42	0.17	0.48	0.46	0.19	0.79	0.75
nSNE	0.44*	0.82	0.78	0.67	0.88	0.86	0.56	0.94	0.92
WEHG (I)	**0.54**	0.85	0.83*	0.69*	0.90	0.89*	0.57	0.96*	0.94
WEHG (RW)	0.42	0.87*	0.83*	0.63	0.91*	0.89*	0.59*	0.96*	0.95*
WEHG	**0.54**	**0.88**	**0.86**	**0.70**	**0.92**	**0.91**	**0.65**	**0.97**	**0.96**

4.2.2 RESULTS

We evaluate WEHG's performance quantitatively on three dataset setups: transductive, inductive, and unfiltered. WEHG (I) denotes the proposed model without the memory sharing. We also show the performance of a random walk version WEHG (RW) by replacing the RL strategy with a random selection strategy. We evaluate link prediction on node pairs with both nodes available in the training graph in the *transductive* setting. For the *inductive* setting, we perform a quantitative evaluation only on the node pairs in which at least a node in the node pair is not in the training graph. The *unfiltered* setup applies no prefiltering on the node pair nodes, hence covering all node pairs setup combinations. Considering the unbalanced nature of the data, we evaluate the performance using the precision-recall AUC (PR-AUC).

The result presented in Table 2 shows that the proposed model outperforms the baseline models across the three datasets. Some baselines can not be evaluated in the inductive setup as these methods are not directly built to handle new unseen data samples. *nSNE*, and *SGCN* were modified for inductive setting. Table 2 shows that combing text features and Node2Vec features (results in the row of TextGraph) did not significantly improve the performance compared to Node2Vec. This underscores the importance of terms' usage (cooccurrence relationship) information extracted from scientific literature in hypothesis generation, as using just the terms' description does not contain enough information in analyzing possible relationships. WEHG non-linearly incorporates the textual and structural information via an attentive graph walk, capturing information from both sources simultaneously and leading to better performance.

SGCN and nSNE are designed for signed networks; hence we incorporate the signs given to the edges to create the node representations. We set the positive-negative signs based on node pair link existence. In a cooccurrence biomedical graph, all edges show a kind of relationship. Hence the task becomes to efficiently estimate the neighborhood nodes' relevancy at each walk step given a node pair. Therefore, aggregating information from all neighboring nodes at each recurrent step will outperform a similar method with a less efficient relevance estimation. This can be seen in the performance of WEHG (RW) and WEHG (I), as WEHG (RW) assigns equal weight to all neighboring nodes. The result also shows that memory sharing across the simultaneous graph walk leads to better representation in the embedding layer and also the relevace estimation for the RL graph walk process, as seen in the performance of WEHG against WEHG (RW).

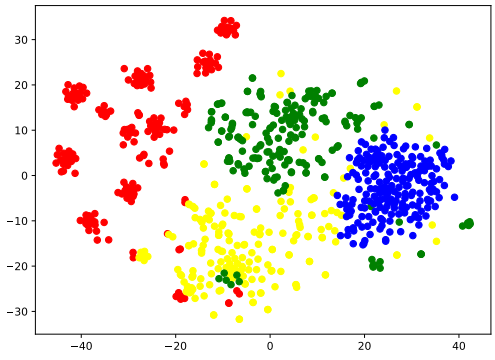

Figure 2: Pair embedding visualization. The blue color denotes the true positive samples, the red points are unobserved negative, the green points are unobserved positive, and the yellow points are false negatives.

Table 4: Walk trajectories of node pairs. The trajectories show the nodes visited in the graph walk for each node in the node pairs.

Term pairs	Terms	Walk Trajectory
Infection Control and Job-related Stress	Infection Control	Fascitides → Air → Spinal Cord Injury → Occupational Therapies → Diet
	Job-related Stress	School → Nursing Faculty → Nursing Personnel → Durable Medical Equipment → Health
Death and Cerebral Concussion	Death	Treatment Delays → Sex → Ganglionic Cyst → False Aneurysm → Tear
	Cerebral Concussion	Trauma → Major Depressive Disorder → Risk Factor → Knowledge → Grouping, Blood
Capillary and SARS COV2	Capillary	Infectivity → Class II Major Histocompatibility Molecules → CD4 Receptors → Re infection → Cholangitides
	SARS COV2	Transmission → Breathing → Angioneurotic Edemas → Operative Surgical Procedure → HEPES [3]
Herbal Medicine and SARS COV2	Medicine	Affects → sodium sulphite → Muriatic Acid → Pneumogastric Nerves → High Blood Pressures
	SARS COV2	Pandemi → Pharmaceutic → Blossom → Protein Binding → Endrofloxicin

Pair Embedding Visualization. To further understand the model performance, We sample and visualize the embeddings of pairs from the COVID-19 data using the t-SNE method Maaten & Hinton (2008). We randomly sample 800 pairs for clear visibility and visualize the learned embeddings in Figure 2. We denote with colors the true label in comparison to the predicted labels. We observe that the true negatives (red) are further apart from the other groups. In contrast, the false positives (green) and false negatives (yellow) are both closer to the true positives (blue). This observation keeps with the theory that the unobserved links are a mixture of positive and negative pairs as although we can say with certainty that observed pairs are connected, we can not say that unobserved nodes should not be connected with similar certainty for real-world data. In real-world biomedical data, several factors such as missing data, lack of sufficient studies, or human mistakes can affect the observation of node pair connections.

4.3 Quanlitative Analysis

To evaluate the proposed model's performance on new diseases, we first manually gathered a list of investigated diseases linked to the COVID-19. We then check if the model was able to predict the existence of these links. We also note that the COVID-19 infection node object was not available in the training graph. Hence the model sees the COVID-19 virus as a new term and tries to predict its relationship with the terms shown in Table 3. We compare the prediction score of WEHG to the second-best method (nSNE). We observe that although the nSNE could predict the relationship between COVID-19 infection and most of the terms, WEHG gave higher probability scores showing more confidence in the existence of a link. We delved further into the result to understand the low probability score for the terms with lower scores. The mean score of the true negatives for the nSNE model is 0.11, more than twice that of WEHG, which is 0.04.

Table 3: Case analysis of preselected test cases with scores (from WEHG and nSNE) showing whether the terms should be linked to COVID-19.

Diseases	Prediction Score	
	WEHG	nSNE
Cytokine	0.93	0.79
Consumption Coagulopathies	0.58	0.23
Cancer of Lung	0.79	0.68
Cancer of the Breast	0.80	0.68
Obesity	0.91	0.66
Soreness, Muscle	0.36	0.32
Insomnias	0.50	0.26
Pink Eyes	0.24	0.40
Hypercytokinemias	0.13	0.24
Smell Loss	0.68	0.13
Taste Loss	0.57	0.13

Prediction Explainability via graph walk. We propose that the analysis of the nodes visited in the graph walk for both nodes can further explain the prediction. This ability is one of the key advantages of the proposed model. We train a model using the COVID-19 dataset and extract the walk trajectory in predicting the node pair relationships on the test data. We performed three case studies to analyze the explainability performance of the proposed model. First, we analyze the nodes visited in the walk trajectory of sampled true positive pairs from the test data. We present the visited

[3] N-(2-Hydroxyethyl)piperazine-N'-(2-ethanesulfonic acid)

node in the graph walk of four selected sampled pairs (based on interest in the community) in Table 4. We observe that the nodes visited during the walk capture the relationship between the pairs from the walk path analysis.

Analyzing the pair <*Infection control and Job-related stress*> we see the nodes visited in the walk trajectory for *infection Control* are related to *job-related Stress*. For instance, good air quality, good diet, and availability of occupational therapies can help control work-related stress. The nodes visited in the trajectory of job-related stress include places where job-related stress can be prevalent and need to be controlled (i.e., schools and medical facilities).

For the pair <*Death and Cerebral concussion*>, the nodes in the graph walk trajectory for *death* are indirectly related to fatal cerebral concussions. For instance, treatment delays can cause the Death of a patient with a cerebral concussion. Ganglionic cyst, false aneurysm, and tear can be a result of trauma related to concussion. Most of the nodes visited in the walk trajectory for the cerebral concussion walk are related to cerebral concussion causes or results.

The result's explainability can also be seen in the last two pairs where the nodes visited in the trajectories provide hints on the relationship between the two nodes in each pair. For the pair <*Capillary and SARS COV2*>, the *capillary* (a small blood vessel) is prone to be attacked by infections. Cholangitides, where capillaries can play a significant role, is related to SARS COV2 due to the effect of SARS COV2 on the lungs (Praveen et al., 2020). For the pair <*Medicine and SARS COV 2*>, pneumogastric Nerves (also known as vagus nerves) are vital nerves as they interface with the parasympathetic control of the heart, lungs, and digestive tract. Due to its functions, several works have been dedicated to finding the effects of SARS COV2 on the nerves (Selma, 2020; Asad et al., 2020).

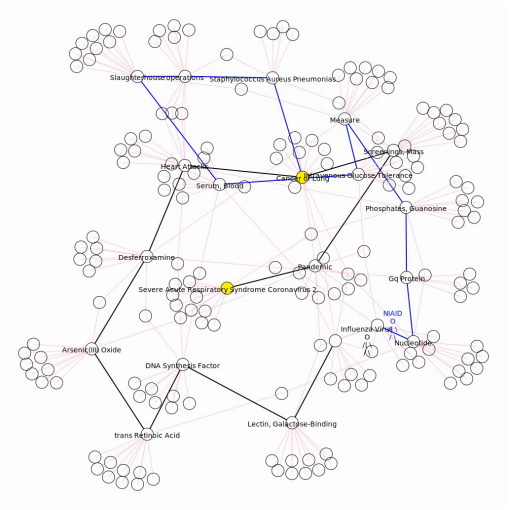

Figure 3: Subgraph showing trajectories of walks starting from each node in the node pair (yellow nodes) and terminating at the nodes with the stick figures.

Next, Figure 3 which shows the trajectory of the graph walk agent on the graph for term pairs <Lung cancer and SARS COV2>. We set a walk length of 10 to show how the agent explores the graph. We see that the walk of SARS COV2 intersects the trajectory of the *Lung cancer* walk at the node *Lung cancer*. From research, we do know that the COVID-19 affects the lung (Praveen et al., 2020), with difficulty in breathing being one of the symptoms. These trajectories show that the model can walk attentively on the graph using the learned policy.

Furthermore, we sent a list of 100 correctly predicted term pairs to a team of domain experts to evaluate the prediction insights and validity. With respect to the generated insights based on the trajectories, the average percentage of correct insights was 0.73.

5 Conclusion

This paper studies the hypothesis generation problem and proposes WEHG - a node pair representation learning method for attributed graphs based on guided graph walk via reinforcement learning. In this framework, an agent chooses the most relevant nodes for information aggregation recurrently. The walk decision is determined by considering the previous walk context, the current node content, the node its one-hop neighborhood. The accumulated information from the nodes in the graph walk sequence is finally aggregated and used to predict the probability of the nodes being connected. The quantitative experiments and analyses show that WEHG outperforms several state-of-the-art methods in both inductive and transductive settings. The qualitative analyses also show the effectiveness of the proposed method.

REFERENCES

Uchenna Akujuobi, Jun Chen, Mohamed Elhoseiny, Michael Spranger, and Xiangliang Zhang. Temporal positive-unlabeled learning for biomedical hypothesis generation via risk estimation. *arXiv preprint arXiv:2010.01916*, 2020a.

Uchenna Akujuobi, Michael Spranger, Sucheendra K Palaniappan, and Xiangliang Zhang. T-pair: Temporal node-pair embedding for automatic biomedical hypothesis generation. *IEEE Transactions on Knowledge and Data Engineering*, 2020b.

Zain Ul Abideen Asad, Adam Przebinda, Amna Mohyud Din Chaudhary, Samid Farooqui, Houssein Youness, and Stavros Stavrakis. The role of vagus nerve stimulation in sepsis. *Bioelectronics in Medicine*, 3(4):51–62, 2020.

Seung Han Baek, Dahee Lee, Minjoo Kim, Jong Ho Lee, and Min Song. Enriching plausible new hypothesis generation in pubmed. *PloS one*, 12(7):e0180539, 2017.

Rajarshi Bhowmik and Gerard de Melo. Explainable link prediction for emerging entities in knowledge graphs. In *International Semantic Web Conference*, pp. 39–55. Springer, 2020.

Bo-Christer Björk, Annikki Roos, and Mari Lauri. Global annual volume of peer reviewed scholarly articles and the share available via different open access options. In *ELPUB*, pp. 178–186, 2008.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NeurIPS*, pp. 2787–2795, 2013.

Angelos Chatzimparmpas, Rafael M Martins, Ilir Jusufi, Kostiantyn Kucher, Fabrice Rossi, and Andreas Kerren. The state of the art in enhancing trust in machine learning models with the use of visualizations. In *Computer Graphics Forum*, volume 39, pp. 713–756. Wiley Online Library, 2020.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Tyler Derr, Yao Ma, and Jiliang Tang. Signed graph convolutional networks. In *ICDM*, pp. 929–934, 2018.

Zuohui Fu, Yikun Xian, Ruoyuan Gao, Jieyu Zhao, Qiaoying Huang, Yingqiang Ge, Shuyuan Xu, Shijie Geng, Chirag Shah, Yongfeng Zhang, et al. Fairness-aware explainable recommendation over knowledge graphs. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 69–78, 2020.

Zheng Gao, Gang Fu, Chunping Ouyang, Satoshi Tsutsui, Xiaozhong Liu, Jeremy Yang, Christopher Gessner, Brian Foote, David Wild, Ying Ding, et al. edge2vec: Representation learning using edge semantics for biomedical knowledge discovery. *BMC bioinformatics*, 20(1):306, 2019.

Vishrawas Gopalakrishnan, Kishlay Jha, Aidong Zhang, and Wei Jin. Generating hypothesis: Using global and local features in graph to discover new knowledge from medical literature. In *BICOB*, pp. 23–30, 2016.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *KDD*, pp. 855–864, 2016.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, pp. 1024–1034, 2017.

Dimitar Hristovski, Carol Friedman, Thomas C Rindflesch, and Borut Peterlin. Exploiting semantic relations for literature-based discovery. In *AMIA annual symposium proceedings*, volume 2006, pp. 349, 2006.

Kishlay Jha, Guangxu Xun, Yaqing Wang, and Aidong Zhang. Hypothesis generation from text based on co-evolution of biomedical concepts. In *KDD*, pp. 843–851, 2019.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. Knowledge graph completion with adaptive sparse transfer matrix. In *AAAI*, 2016.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Peder Larsen and Markus Von Ins. The rate of growth in scientific publication and the decline in coverage provided by science citation index. *Scientometrics*, 84(3):575–603, 2010.

John Boaz Lee, Ryan Rossi, and Xiangnan Kong. Graph classification using structural attention. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1666–1674, 2018.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, 2015.

Scott M Lundberg, Bala Nair, Monica S Vavilala, Mayumi Horibe, Michael J Eisses, Trevor Adams, David E Liston, Daniel King-Wai Low, Shu-Fang Newman, Jerry Kim, et al. Explainable machine learning predictions to help anesthesiologists prevent hypoxemia during surgery. *bioRxiv*, pp. 206540, 2017.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

K Yu Michael, Jianzhu Ma, Jasmin Fisher, Jason F Kreisberg, Benjamin J Raphael, and Trey Ideker. Visible machine learning for biomedicine. *Cell*, 173(7):1562–1565, 2018.

Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 845–854, 2019.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *KDD*, pp. 701–710, 2014.

Sharma Praveen, Kumar Ashish, ShriHari Anil Anikhindi, Bansal Naresh, Singla Vikas, Shivam Khare, and Arora Anil. Effect of covid-19 on pre-existing liver disease: What hepatologist should know? *Journal of Clinical and Experimental Hepatology*, 2020.

Wojciech Samek, Grégoire Montavon, Sebastian Lapuschkin, Christopher J Anders, and Klaus-Robert Müller. Toward interpretable machine learning: Transparent deep neural networks and beyond. *arXiv preprint arXiv:2003.07631*, 2020.

Pr Sakhri Selma. Everything is in the vagus nerve: what is the relationship between chronic fatigue syndrome (cfs) and coronavirus? *Cancer Case Reports*, 1:1–4, 2020.

Baoxu Shi and Tim Weninger. Proje: Embedding projection for knowledge graph completion. In *AAAI*, 2017.

Feng Shi, Jacob G Foster, and James A Evans. Weaving the fabric of science: Dynamic network models of science's unfolding structure. *Social Networks*, 43:73–85, 2015.

Uriel Singer, Ido Guy, and Kira Radinsky. Node embedding over temporal graphs. *arXiv preprint arXiv:1903.08889*, 2019.

Neil R Smalheiser and Don R Swanson. Using arrowsmith: a computer-assisted approach to formulating and assessing scientific hypotheses. *Computer methods and programs in biomedicine*, 57(3):149–153, 1998.

Wenzhuo Song, Shengsheng Wang, Bo Yang, You Lu, Xuehua Zhao, and Xueyan Liu. Learning node and edge embeddings for signed networks. *Neurocomputing*, 319:42–54, 2018.

Scott Spangler. *Accelerating Discovery: Mining Unstructured Information for Hypothesis Generation*. Chapman and Hall/CRC, 2015.

Scott Spangler, Angela D Wilkins, Benjamin J Bachman, Meena Nagarajan, Tajhal Dayaram, Peter Haas, Sam Regenbogen, Curtis R Pickering, Austin Comer, Jeffrey N Myers, et al. Automated hypothesis generation based on mining scientific literature. In *KDD*, pp. 1877–1886, 2014.

Rohini K Srihari, Li Xu, and Tushar Saxena. Use of ranked cross document evidence trails for hypothesis generation. In *KDD*, pp. 677–686, 2007.

Padmini Srinivasan and Bisharah Libbus. Mining medline for implicit links between dietary substances and diseases. *Bioinformatics*, 20(suppl_1):i290–i296, 2004.

Don R Swanson and Neil R Smalheiser. An interactive system for finding complementary literatures: a stimulus to scientific discovery. *Artificial intelligence*, 91(2):183–203, 1997.

Justin Sybrandt and Ilya Safro. Validation and topic-driven ranking for biomedical hypothesis generation systems. *bioRxiv*, 2018. doi: 10.1101/263897.

Ehsan Toreini, Mhairi Aitken, Kovila Coopamootoo, Karen Elliott, Carlos Gonzalez Zelaya, and Aad van Moorsel. The relationship between trust in ai and trustworthy machine learning technologies. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 272–283, 2020.

Kush R Varshney. Trustworthy machine learning and artificial intelligence. *XRDS: Crossroads, The ACM Magazine for Students*, 25(3):26–29, 2019.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, volume 14, pp. 1112–1119, 2014.

Dirk Weissenborn, Michael Schroeder, and George Tsatsaronis. Discovering relations between indirectly connected biomedical concepts. *Journal of biomedical semantics*, 6(1):28, 2015.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Yikun Xian, Zuohui Fu, S Muthukrishnan, Gerard De Melo, and Yongfeng Zhang. Reinforcement knowledge graph reasoning for explainable recommendation. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pp. 285–294, 2019.

Yikun Xian, Zuohui Fu, Handong Zhao, Yingqiang Ge, Xu Chen, Qiaoying Huang, Shijie Geng, Zhou Qin, Gerard De Melo, Shan Muthukrishnan, et al. Cafe: Coarse-to-fine neural symbolic reasoning for explainable recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 1645–1654, 2020.

Guangxu Xun, Kishlay Jha, Vishrawas Gopalakrishnan, Yaliang Li, and Aidong Zhang. Generating medical hypotheses based on evolutionary medical concepts. In *ICDM*, pp. 535–544, 2017.

Kai Yang, Xinyu Kong, Yafang Wang, Jie Zhang, and Gerard De Melo. Reinforcement learning over knowledge graphs for explainable dialogue intent mining. *IEEE Access*, 8:85348–85358, 2020.

Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. *arXiv preprint arXiv:1603.08861*, 2016.

Bo Yuan, Ciyue Shen, Augustin Luna, Anil Korkut, Debora S Marks, John Ingraham, and Chris Sander. Cellbox: Interpretable machine learning for perturbation biology with application to the design of cancer combination therapy. *Cell systems*, 12(2):128–140, 2021.

Na Zhao, Hanwang Zhang, Meng Wang, Richang Hong, and Tat-Seng Chua. Learning content–social influential features for influence analysis. *International Journal of Multimedia Information Retrieval*, 5(3):137–149, 2016.