DOI: 10.20472/EFC.2022.016.015

ELDA XHUMARI
University of Tirana, Faculty of Natural Sciences, Department of Informatics, Albania

SUELA MAXHELAKU
University of Tirana, Faculty of Natural Sciences, Department of Informatics, Albania

ENDRIT XHINA
University of Tirana, Faculty of Natural Sciences, Department of Informatics, Albania

A REVIEW OF KNOWLEDGE GRAPH AND GRAPH NEURAL NETWORK APPLICATION

Abstract:

Many learning activities include working with graph data, which offers a wealth of relational information between parts. Modeling physical systems, learning molecular fingerprints, predicting protein interfaces, and diagnosing illnesses all need the use of a model that can learn from graph inputs. In other fields, such as learning from non-structural data such as texts and images, reasoning on extracted structures (such as phrase dependency trees and image scene graphs) is a major topic that requires graph reasoning models. Graph neural networks (GNNs) are neural models that use message transmission between graph nodes to represent graph dependency. Variants of GNNs have recently showed ground-breaking performance on a variety of deep learning tasks.
This paper represents a review of the literature on Knowledge Graphs and Graph Neural Networks, with a particular focus on Graph Embeddings and Graph Neural Networks applications as a powerful tool for organizing structured data and making sense of unstructured data, which can be applied to a variety of real-world problems.

Keywords:

Knowledge Graph, Graph Neural Network, DeepWalk, Node2Vec, Structural Deep Network Embedding

JEL Classification: C45

1 Introduction

Knowledge Graphs represent a collection of nodes and edges, so visually it looks as below:

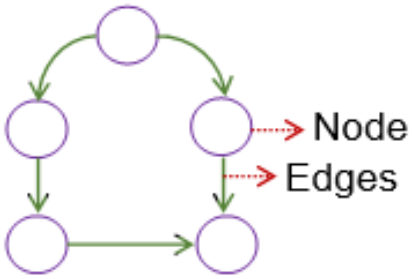

Figure 1: A simple graph

Source: Author

The edges represent the relationship between the nodes. Based on the need, we can have different types of edges representing nodes, so for example, for Social Network perhaps some edges represent a friendship relationship, so the nodes represent people and some others edges represent “enemies”. The difference between a regular graph and a knowledge graph mostly comes down to the domain that they are being used and for what they are being used for. Therefore, graph going to be usually created for searches within graphs, if trying to reroute network packets from one row to another we can probably just call that a graph for the knowledge graph but if we are doing something like a social network and we would appraise the nodes as people and edges just as relationships, we can also consider that a knowledge graph. As a result, when it comes to knowledge graphs the important thing that people care about is the concept of knowledge triplets, so triplets have 3 components, they have: the head, the tail and some relationship that connects them. *(Gaudelet, et al., 2021)*

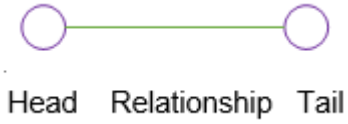

Figure 2: Knowledge triplets

Source: Author

The reason for using knowledge triplets consists in an easier way of describing a knowledge graph and therefore, instead of having it in a visual state we can have this knowledge triplets that represent the connections and store the information. Some useful applications for knowledge graphs and knowledge triplets are known as a knowledge completion:

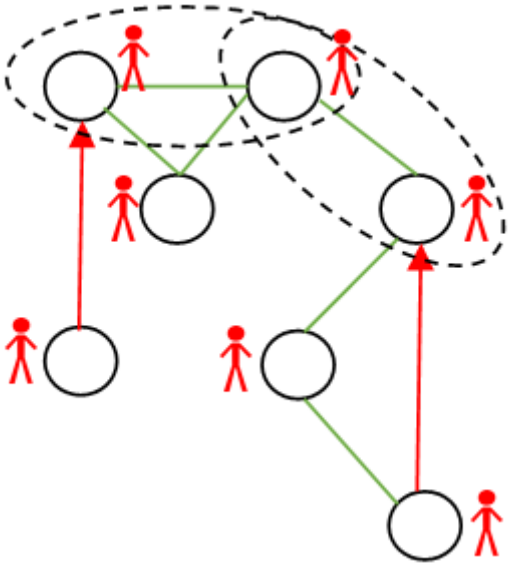

Figure 3: Knowledge completion

Source: Author

The idea of knowledge completion is to use the relationships that already exist and try and predict new relationships, new knowledge triplets that exists within the graph. *(Gaudelet, et al., 2021)*

Additionally, Graph Neural Network (GNN) attempts to leverage the hierarchical structure of graphs in knowledge graphs to learn something about the graph when we don't use case for this, something where we learn a vector representation of each of the nodes and different nodes in the graph are being represented by this vector representation. *(Gaudelet, et al., 2021)*

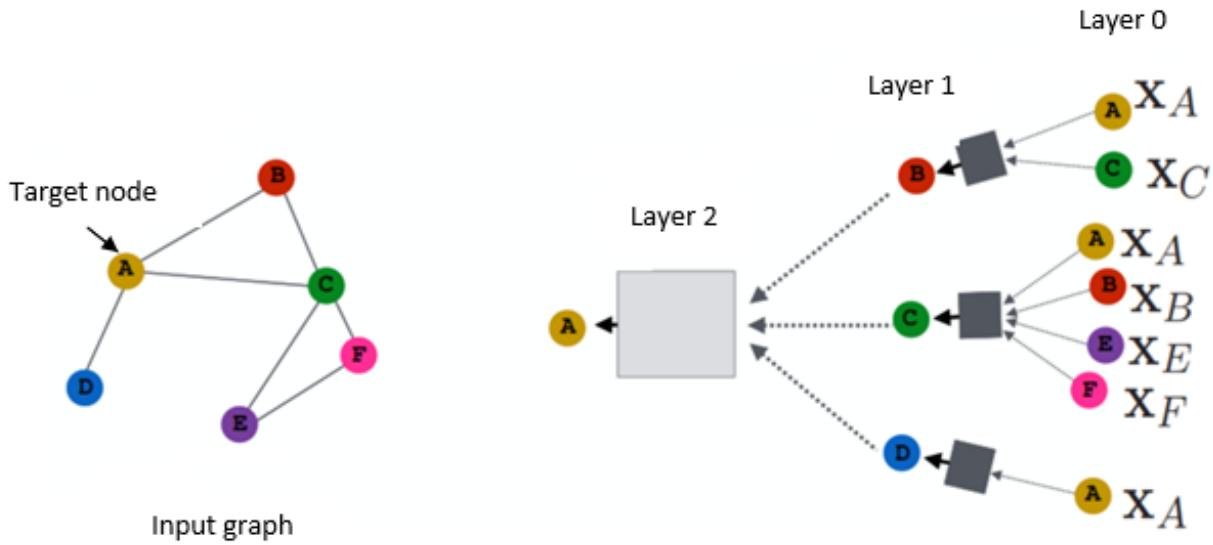

Figure 4: Graph Neural Network

Source: Rex, et al., 2018

2 Some Types of Graph Embeddings

Some example types of graph embeddings, one of the sorts of simplest to understand is DeepWalk. It presents the concept of Random Walk for embedding creation for the first time. A Random Walk is a method of transforming a graph into a series of nodes that can then be used to train a Word2Vec model. Essentially, the model constructs a random path of nodes connected for each node in the network. Once we have these random paths of nodes, we must train a Word2Vec (skip-gram) model to extract the node embeddings. *(Perozzi, Al-Rfou, & Skiena, 2014)*

Another type of graph embeddings is Node2Vec which works at the node level and essentially tries to earn a good representation for these vectors. Node2Vec is an algorithm for generating vector representations of nodes on a graph. The Node2Vec framework learns low-dimensional representations of nodes in a graph by randomly traversing the graph, starting at the target node. This is useful for many machine learning applications. In addition to reducing engineering effort, the representations produced by the algorithm provide greater predictive power. Node2vec follows the intuition that random walks on a graph can be thought of as sentences in a corpus. Each node in the graph is treated as a single word, and the random walk is treated as a sentence. By feeding these "sentences" into a skip-gram or by using a continuous set of words, the model paths found by random walks can be considered as sentences, and traditional data mining methods for documents can be used. The algorithm generalizes previous work based on rigid notions of network neighbourhoods and argues that additional flexibility in learning neighbourhoods is the key to learning richer node representations in graphs. This algorithm is an extension of the Word2Vec Gensim algorithm and is considered one of the best classifiers for nodes in a graph. *(Grover & Leskovec, 2016)*

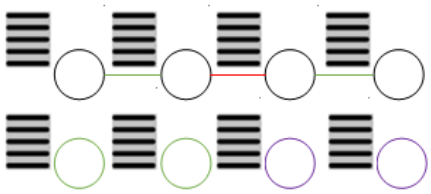

Figure 5: Node2vec

Source: Author

An example of Node2Vec application for social networks: a person which has 3 friends, potentially they also have a similar mutual friend as well, and so one way that we might want to represent these people is to have a vectoral representation and since they are relatively similar in terms of their social network like group of friends, that vector representation would be similar to each other. So, the way that Node2Vec works is that it generates random walks. In the Figure 6 is represented this walk from the start to the end.

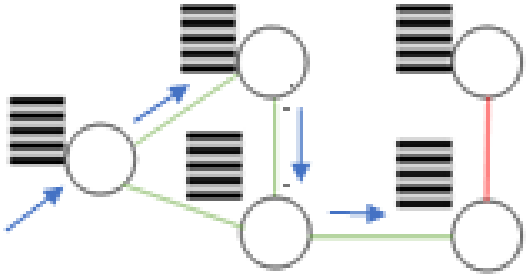

Figure 6: Random walk from the start to the end of a graph

Source: Author

Structural Deep Network Embedding does not use random walks, unlike previous embedding approaches. Instead, it makes an attempt to learn from two different metrics:

- Two nodes are defined comparable if they share an edge;
- Two nodes are comparable if they have a lot of neighbouring/adjacent nodes in common.

The ultimate objective is to capture structures that are quite non-linear. This is accomplished by preserving the first order (supervised) and second order (unsupervised) network proximities with deep autoencoders (semi-supervised). *(Wang, Cui, & Zhu, 2016)*

3 Application: Recommended systems

Graphs evolve in the context of user interactions with products on e-commerce platforms. As a result, many companies use graph neural networks to build recommender systems. Usually, graphs are used to model user interaction with products, teach embeddings based on a properly selected negative sample, and by ranking the results, personalized product offers are selected and shown to specific users in real time. One of the first services with a such mechanism was Uber Eats: the GraphSage neural network selects food and restaurant recommendations. Although the graphs are relatively small in the case of food recommendations due to geographical restrictions, some companies use neural networks with billions of connections. For example, the Alibaba has launched graph embeddings and graph neural networks for billions of users and products. As a result of the Aligraph pipeline, we can build a graph with 400 million nodes in just five minutes, because it supports efficient, distributed graph storage, optimized fetch operators, and a bunch of native graph neural networks. This pipeline is now used for recommendations and personalized searches across the company's products. *(Zhu, et al., 2019)*

Pinterest has proposed a PinSage model that efficiently matches neighbor nodes with personalized PageRank and updates vertex embeddings by aggregating information from neighbors. The next PinnerSage model can already work with multi-embeddings to accommodate different user tastes. These are just a couple of notable examples in the realm of recommender systems, for example Amazon's research on knowledge graphs and graph neural networks, or Fabula AI's use of graph neural networks to detect fake news. But even without this, it is clear that graph neural networks show promising results with a significant signal from user interactions. *(Rex, et al., 2018)*

Graph Neural Networks can be classified according to different attributes and below are listed some types of tasks that we can solve with GNN:

- Node Level
    - Clustering – graph visualization or retrieval systems
- Edge Level
    - Link prediction/Knowledge Completion – friend recommendations
- Graph Level
    - Graph classification – drug identification

Pharmaceutical corporations are aggressively seeking new ways to develop pharmaceuticals, fiercely competing with one another and investing billions of dollars in research. Graphs can be used to represent interactions at many levels in biology. Bonds between nodes, for example, reflect interatomic forces in a molecule or interactions between amino acid bases in a protein at the molecular level. Graphs can illustrate interactions between proteins and RNA or metabolic products on a bigger scale. Graphs can be used for target discovery, molecular property prediction, high-throughput screening, drug design, protein engineering, and drug repurposing, depending on the level of abstraction. The findings of MIT researchers, published in Cell in 2020, were perhaps the most promising result of using graph neural networks in this domain. They used a Chemprop deep learning model to predict the antibiotic capabilities of the compounds, which included inhibition of Escherichia coli reproduction. Chemprop was trained on just 2,500 molecules from an FDA-approved library before being applied to a bigger dataset, which included a Drug Repurposing Hub containing the Halicin molecule, which was renamed after the AI HAL 9000 from 2001". Because its structure is so unlike to that of known antibiotics, Halicin has only been examined in relation to the treatment of diabetes until now. Halicin, on the other hand, has been found to be a broad spectrum antibiotic in vitro and in vivo. The importance of Halicin's qualities revealed using graph neural networks was underscored by a thorough comparison with strong neural network models. The Chemprop design is notable for other reasons, in addition to its practical value: unlike many graph neural networks, it has 5 layers and 1600 hidden dimensions, which is significantly larger than the standard parameters of graph neural networks for such applications. *(Gaudelet, et al., 2021)*

Graph neural networks may solve a variety of natural science problems by representing interactions between particles or molecules as graphs and predicting the attributes of new materials and substances. Facebook and CMU, for example, are working together on the Open Catalyst initiative to find innovative methods to store renewable energy from the sun and wind. One proposed answer is to transfer this energy into other fuels, such as hydrogen, through chemical processes. However, new catalysts for high-intensity chemical reactions must be developed, and current approaches such as DFT are too costly. The project's creators have compiled the most comprehensive list of graph neural network catalysts, DFT attenuations, and base levels.

Although items in the actual world are highly linked, graph neural networks can successfully analyse pictures of the objects. Scene graphs, for example, are a set of items in a picture with their relationships that may be used to interpret the content of an image. Images are found, their information is interpreted and comprehended, subtitles are added, visual inquiries are answered, and images are generated using scene graphs. Model performance can be considerably improved by using the graphics. *(Ashual & Wolf, 2019)*

4 Development of knowledge graph and graph neural network in Albania

The structure embeddings and graph neural networks models introduced in 2016 were evaluated in a variety of real-world applications and demonstrated excellent efficiency, including the most well-known example of the PinSAGE model, which is utilized in Pinterest's recommendation system. Since then, research on the field has grown at an exponential rate, with more and more applications in domains where prior methodologies failed to account for links between items in models. It's worth noting that graph neural networks have given machine learning automation and the hunt for new efficient neural network architectures a fresh incentive to develop. *(Rex, et al., 2018)*

Unfortunately, Albania in most cases lags behind modern research in the field of artificial intelligence. The number of articles at international conferences and in leading journals is an order of magnitude less than those of scientists from the US, Europe, Russia and China, and financial support for research in new areas is met with resistance in the ossified environment of academics who are stuck in approaches from the last century. As a result, at leading conferences, the names of Albanian researches are mainly affiliated with Western universities.

5 Conclusion

Graph applications abound, ranging from knowledge graphs to social networks. Graph neural networks has given way to transformers with their large architectures that describe all possible dependencies in unstructured data, but generate models that cost a lot to train and are available only to mega-corporations. Graph neural networks have not only successfully taken their place as the standard for building machine learning on structural data, but have also proven to be an effective tool for building structural attention in related industries, including high performance in multi-example learning and metric learning tasks. As a result of graph neural networks application, we will get new discoveries in materials science, pharmacology and medicine. Perhaps there will be new, more efficient models for big data that have the properties of transferring knowledge between different graph data. Models will overcome applicability problems for graphs whose structure is opposite to the similarity of features, but in general, this area of machine learning has become an independent science.

Table 1: Summary of Graph Neural Networks

	Advantages	Disadvantages
Graph Neural Networks	. A powerful architecture for modeling structural data; . High performance in generating models for unstructured data - can describe all possible dependencies in unstructured data.	. It is inefficient to update the hidden node states for a fixed location iteratively; . Most popular neural networks utilize different parameters in different layers to extract hierarchical features, while GNN uses the same parameters per iteration; . There are also certain informative aspects at the edges that a typical GNN cannot model properly *(for example, edges in a knowledge graph, for example, have relationship kinds, and the propagation of a message along different edges must differ based on their types. Furthermore, locating concealed edge states is a critical challenge.);* . High cost.

6 References

Ashual, O., & Wolf, L. (2019). Specifying object attributes and relations in interactive scene generation. *IEEE/CVF International Conference on Computer Vision*, (p. 45614569).

Gaudelet, T., Day, B., Jamasb, A. R., Soman, J., Regep, C., Liu, G., . . . Roblin, D. (2021). Utilising Graph Machine Learning within Drug Discovery and. *Briefings in Bioinformatics*.

Grover, A., & Leskovec, J. (2016). Node2vec: scalable feature learning for networks. *KDD*, (pp. 855-864).

Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). DeepWalk: Online Learning of Social Representations. *International Conference on Knowledge Discovery and Data Mining.*

Rex, Y., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., & Leskovec, J. (2018). Graph convolutional neural networks for web-scale recommender systems. *24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 974-983). ACM.

Wang, D., Cui, P., & Zhu, W. (2016). Structural Deep Network Embedding. *22nd ACM SIGKDD international conference on knowledge discovery and data mining*, (pp. 1225-1234). San Francisco.

Zhu, R., Zhao, K., Yang, H., Lin, W., Zhou, C., Ai, B., . . . Zhou, J. (2019). Aligraph: a comprehensive graph neural network platform.