University of Camerino

School of Sciences and Technology

Master of Science in
Computer Science (LM-18)

University of Applied Sciences and Arts Northwestern Switzerland

School of Business

Master of Science in
Business Information Systems

HAI-BEMS: A Hybrid Artificial Intelligent Approach for Building Energy Management Systems

Candidate:

Piermichele Rosati

Matricula 124176

FHNW Supervisor:

Dr. Emanuele Laurenzi

Unicam Supervisor:

Prof. Michela Quadrini

Empa Supervisor:

Dr. James Allan

A thesis presented to the School of Sciences and Technology of the University of Camerino and School of Business of the University of Applied Sciences and Arts Northwestern Switzerland in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

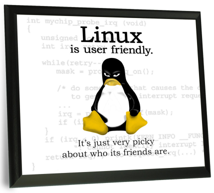
A.Y. 2023/2024

Abstract

The Thesis Abstract is written here (and usually kept to just this page)...

Statement of Authenticity

I, Piermichele Rosati, hereby confirm that this report was performed autonomously using only the sources, aids and assistance stated in the report, and that quotes are readily identifiable as such.

Signed: 

Date: 12th June 2024

Contents

Chapter 1

Introduction

1.1 Background

1.2 Problem Statement

1.3 Thesis Statement

1.4 Research Questions

Scarselli et al., 2009

Chapter 2

Literature Review

2.1 Knowledge Graphs

TODO: change intro

TODO: reread every section and change (modifying, adding something)

Knowledge Graphs (KGs) have came up as a key technology in the field of data management and Artifical Intelligence (AI), enabling sophisticated data integration, retrieval and analysis, and analysis. This literature review provides an in-depth examination of KGs, their theoretical foundations, practical applications and recent advances.

Theoretical Foundations of Knowledge Graphs

Definition and Structure

KGs are directed graph-based data structures that represent real-world entities and their interrelations, providing a way to model complex domains and their underlying semantics. A KG consists of nodes (entities) and edges (relationships), forming a network of interconnected information. This structure allows KGs to capture rich contextual information and provide a semantic framework for data (Hogan et al., 2021).

A KG refers to a semantic network graph which is consisted of diverse entities, concepts, and relationships in the real world. It is used to formally describe various things and their associations in the real world. KGs are generally represented in triples $KG = \{E, R, F\}$.

- E represents the entity set $\{e_1, e_2, ..., e_E\}$, and the entity e is the most basic element in the KG, referring to the items that exist objectively and can be distinguished from each other.
- R represents the relation set $\{r_1, r_2, ..., r_R\}$, and the relation r is an edge in the KG, representing a specific connection between different entities.
- F represents the fact set $\{f_1, f_2, ..., f_F\}$, and each f is defined as a triple $(h, r, t) \in f$, in which h denotes the head entity, r stands for the relationship, and t indicates the tail entity.

Ontologies and Semantic Web Technologies

An ontology is a formal representation of knowledge in a domain, specifying the concepts, relationships, and constraints that exist within that domain. The term "ontology" can be used to the shared understanding of soma domain of interest (Uschold and Gruninger, 1996). Ontologies play a critical role in defining the schema and semantics of KGs. They specify the types of entities, relationships, and constraints, thereby providing a formalized structure for the data. The Semantic Web technologies, particularly the Resource Description Framework (RDF) and the Web Ontology Language (OWL), are fundamental to the development and functioning of KGs (G. (Grigoris) Antoniou and Frank. Van Harmelen, 2008).

- RDF is a standard model for data interchange on the web. It uses triples (subject-predicate-object) to represent information, providing a flexible and extensible framework for creating and managing KGs (W3C CITATION).
- OWL is used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. It enables more complex and expressive representations compared to RDF Schema (RDFS) (Deborah and van Harmelen Frank, 2004).

Query Languages

SPARQL Protocol and RDF Query Language (SPARQL) is the standard query language for retrieving and manipulating data stored in RDF format. It allows users to write complex queries to extract specific information from a KG, making it a powerful tool for data analysis and knowledge discovery (Pérez et al., 2009).

Cypher is another query language for graph databases, such as Neo4j, that allows users to interact with graph data using a pattern-matching syntax. Cypher queries are used to traverse the graph, retrieve specific patterns, and perform operations on the data (Francis et al., 2018).

Applications of Knowledge Graphs

General Applications

KGs have been adopted across various domains due to their ability to integrate heterogeneous data sources, provide semantic context, and enable advanced querying and reasoning.

- In healthcare, KGs are used to integrate patient records, clinical trials, research data, and medical ontologies, enabling personalized medicine and decision support systems. They help in identifying relationships between diseases, treatments, and patient outcomes (Kapanipathi et al., 2020).
- Financial institutions leverage KGs to connect data from various sources, such as market data, regulatory information, and customer transactions. This integration facilitates risk management, fraud detection, and compliance monitoring (Tchechmedjiev et al., 2019).
- In e-commerce, KGs enhance product recommendation systems by linking customer preferences, purchase history, and product information. They enable more personalized and relevant recommendations, improving customer satisfaction and sales (Zhang et al., 2021).

Enterprise Knowledge Management

Within enterprises, KGs are used to manage and utilize internal knowledge effectively. They integrate data from different departments, such as human resources, finance, and operations, providing a unified view of the organization's information. This integration supports decision-making, collaboration, and innovation (Pujara et al., 2013).

Search and Information Retrieval

KGs significantly enhance search engines by providing semantic search capabilities. They enable the understanding of user queries in context, allowing for more accurate and relevant search results. Google's Knowledge Graph is a prominent example, enhancing search results with information about entities and their relationships (Singhal et al., 2012).

Recent Advancements in Knowledge Graphs

Integration with Machine Learning

Recent research has focused on integrating KGs with Machine Learning (ML) and Deep Learning (DL) techniques to enhance their capabilities and applications. These integrations have led to significant advancements in various areas, including Natural Language Processing (NLP), recommendation systems, and predictive analytics.

- **Knowledge Graph Embeddings (KGEs)**: KGE techniques represent entities and relationships in a continuous vector space, enabling the use of machine learning algorithms for tasks such as link prediction, entity classification, and clustering. Popular methods include TransE, TransH, and TransR, each providing different ways to model relationships in the embedding space (Q. Wang et al., 2017).
- **Graph Neural Networks (GNNs)**: GNNs are DL models designed to operate on graph-structured data. They leverage the relational nature of graphs to perform tasks such as node classification, link prediction, and graph classification. GNNs have been successfully applied to enhance the capabilities of KGs in various domains (Wu et al., 2021).

  GNNs are described in Sec. 2.2.

Natural Language Processing and Question Answering

KGs have been instrumental in advancing NLP applications, particularly in question answering systems. By providing structured and semantically rich information, KGs enable systems to understand and generate human language more effectively.

- **Question Answering Systems**: KGs support question answering systems by enabling them to retrieve and reason over structured data. These systems can answer complex queries by traversing the graph and applying logical inferences based on the relationships between entities (Yasunaga et al., 2021).
- **Semantic Search and Text Analysis**: KGs enhance text analysis and semantic search by providing contextual information about entities mentioned in the text. This contextual understanding improves the accuracy of information retrieval and the relevance of search results (Fernández et al., 2011).

Challenges and Future Directions

Scalability and Performance

As KGs grow in size and complexity, scalability and performance become critical challenges. Efficient storage, querying, and updating of large KGs require advanced techniques and architectures. Research in distributed computing, graph databases, and parallel processing is ongoing to address these issues (CITATION).

Data Quality and Integration

Ensuring the accuracy and consistency of data in KGs is essential for their reliability. Data quality issues, such as inconsistencies, duplications, and inaccuracies, can significantly impact the performance of applications relying on KGs. Developing methods for automatic data cleaning, validation, and integration is an active area of research (Paulheim, 2017).

Privacy and Security

The integration of sensitive data into KGs raises concerns about privacy and security. Protecting personal and confidential information while allowing for meaningful data analysis is a significant challenge. Research is focusing on developing techniques for secure data sharing, access control, and anonymization within KGs (Bonatti et al., 2017).

Interoperability and Standardization

Interoperability and standardization are crucial for the widespread adoption of KGs. Ensuring that different KGs can work together seamlessly and that their data can be easily integrated requires the development of common standards and protocols. Efforts such as the Linked Open Data (LOD) initiative and W3C standards aim to address these challenges (Bizer et al., 2023).

Conclusion

Knowledge Graphs represent a transformative technology for data integration, retrieval, and analysis, offering significant benefits across various domains. Their ability to provide semantic context and capture complex relationships makes them invaluable for applications in healthcare, finance, e-commerce, and enterprise knowledge management. Recent advancements in machine learning, particularly in the integration with GNNs, have further enhanced the capabilities of KGs, opening new avenues for research and application. However, challenges related to scalability, data quality, privacy, and interoperability remain and must be addressed to fully realize the potential of KGs. Continued research and development in these areas will be crucial for the future evolution and adoption of KGs.

2.2 Graph Neural Networks

TODO: change, reread

Graph Neural Networks have become a cornerstone in the realm of deep learning, particularly for tasks involving non-Euclidean data structures. These models have significantly impacted various domains, such as social network analysis, bioinformatics, recommendation systems, and NLP. This literature review explores the development, key architectures, methodologies, applications, challenges, and future directions of GNNs.

Historical Context and Evolution

Early Work and Foundations

The early works on neural networks for structured data laid the groundwork for GNN. The concept of Recursive Neural Networks (RecNNs) for processing tree-like structures and graph-based methods in machine learning were the precursors to modern GNNs. However, these early models were limited by their computational inefficiencies and inability to scale.

Spectral Approaches

The breakthrough in spectral approaches marked a significant evolution in GNNs. Bruna et al., 2013 introduced a method leveraging spectral graph theory to define convolution operations on graphs. This approach was refined by Defferrard et al., 2016, leading to more efficient models. Kipf and Welling, 2017 simplified this concept further, making it more accessible and practical, thus popularizing the Graph Convolutional Network (GCN).

Key Architectures and Methodologies

Graph Convolutional Networks (GCNs)

GCNs represent one of the most influential architectures in the GNN landscape. They generalize the convolution operation to graph data, enabling the aggregation of feature information from a node's neighbors.

Mathematical Formulation:

A GCN layer can be represented as:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}\right)$$

where:

- $\tilde{A} = A + I$ is the adjacency matrix with added self-loops.
- $\tilde{D}$ is the degree matrix of $\tilde{A}$.
- $H^{(l)}$ and $H^{(l+1)}$ are the input and output feature matrices for layer l.
- $W^{(l)}$ is the trainable weight matrix.
- σ is an activation function like ReLU.

Advantages:

- Simplicity and effectiveness for semi-supervised learning tasks.
- Captures local neighborhood structures well.

Limitations:

- Limited expressiveness due to fixed aggregation scheme.
- Struggles with capturing long-range dependencies.

Graph Attention Networks (GATs)

Veličković et al., 2017 introduced GATs, which incorporate attention mechanisms to dynamically weigh the importance of neighboring nodes.

Attention Mechanism:

$$e_{ij} = \text{LeakyReLU}\left(a^T [Wh_i \| Wh_j]\right)$$

where e_{ij} is the attention score, a is the learnable attention vector, and $\|$ denotes concatenation.

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

The node features are updated as:

$$h_i' = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} W h_j\right)$$

Advantages:

- Handles heterogeneous graphs by assigning different importances to neighbors.
- Enhanced interpretability through attention weights.

Limitations:

- Computationally expensive due to the attention mechanism.
- Can become inefficient for very large graphs.

GraphSAGE

GraphSAGE (Hamilton et al., 2017) introduced an inductive approach that can generalize to unseen nodes by sampling and aggregating features from a node's local neighborhood.

Sampling and Aggregation:

GraphSAGE samples a fixed-size set of neighbors and uses aggregation functions such as mean, LSTM, or pooling to update node embeddings:

$$h_i' = \sigma\left(W \cdot \text{AGG}\left(\{h_j, \forall j \in \mathcal{N}(i)\}\right)\right)$$

Advantages:

- Scalable to large graphs.
- Supports inductive learning.

Limitations:

- Information loss due to fixed-size sampling.
- Requires carefully designed aggregation functions.

Message Passing Neural Networks (MPNNs)

MPNNs (Gilmer et al., 2017) formalized the message-passing framework for GNNs. In each layer, nodes exchange messages with their neighbors and update their states.

General Framework:

1. Message Function: $m_{ij}^{(l)} = M(h_i^{(l)}, h_j^{(l)}, e_{ij})$
2. Update Function: $h_i^{(l+1)} = U\left(h_i^{(l)}, \sum_{j\in\mathcal{N}(i)} m_{ij}^{(l)}\right)$

where h_i and h_j are node features, and e_{ij} are edge features.

Advantages:

- General and flexible framework.
- Can model complex dependencies and interactions.

Limitations:

- High computational cost for dense graphs.
- Complexity in designing effective message and update functions.

Training Techniques and Challenges

Mini-Batch Training

To manage large graphs, mini-batch training techniques are employed. Subgraphs or neighborhoods are sampled in each training iteration to reduce memory consumption and improve efficiency.

Graph Partitioning

Graph partitioning techniques like METIS and Louvain divide large graphs into smaller subgraphs that can be processed independently. This helps in parallelizing computations and managing memory usage.

Challenges:

- Maintaining the integrity of graph structure during partitioning.
- Ensuring balanced computational load across partitions.

Optimization Algorithms

Specialized optimizers, such as Adam and RMSprop, are used to stabilize training. Regularization techniques like dropout and weight decay help prevent overfitting.

Applications of GNNs

Social Network Analysis

GNNs are extensively used in social network analysis for tasks such as community detection, link prediction, and influence maximization. They model complex interactions and dependencies among users, providing insights into social dynamics.

Examples:

- Community Detection: Using GNNs to identify overlapping communities in social networks (Chen et al., 2017).
- Link Prediction: Predicting future connections by learning node embeddings (Zeng et al., 2019).

Biological Networks

In bioinformatics, GNNs are used to analyze molecular structures, predict protein functions, and understand biological processes. They model intricate interactions between biological entities, aiding in drug discovery and genomics.

Examples:

- Protein-Protein Interaction: Predicting interactions between proteins by modeling their structural properties (Fout et al., 2017).
- Drug Discovery: Identifying potential drug compounds by analyzing molecular graphs (Jin et al., 2018).

Recommendation Systems

GNNs enhance recommendation systems by modeling user-item interactions. They improve the accuracy and relevance of recommendations by capturing complex relationships.

Examples:

- Collaborative Filtering: Enhancing collaborative filtering with GNNs to model user-item interactions (X. Wang et al., 2019).
- Content-Based Recommendations: Using GNNs to model item features and user preferences for personalized recommendations (Ying et al., 2018).

Natural Language Processing

GNNs are applied in NLPs for tasks like semantic parsing, machine translation, and text classification. By representing sentences or documents as graphs, GNNs capture relationships between words or entities.

Examples:

- Semantic Parsing: Modeling the syntactic structure of sentences for accurate semantic parsing (Zeng et al., 2019).
- Relation Extraction: Extracting relationships between entities in text using GNNs to model dependency trees (Sahu et al., 2019).

Recent Advances and Future Directions

Scalability

Scalability remains a critical challenge for GNNs. Recent advancements focus on models and techniques that handle large-scale graphs efficiently. Methods like GraphSAINT (Zeng et al., 2019) and Cluster-GCN (Chiang et al., 2019) emphasize sampling and partitioning strategies to enable scalable training.

Future Directions:

- Distributed GNNs: Developing distributed frameworks for training GNNs on large-scale graphs.
- Efficient Sampling Techniques: Improving sampling methods to balance efficiency and information retention.

Expressiveness

Enhancing the expressiveness of GNNs involves designing architectures that capture more complex patterns and dependencies. Higher-order GNNs and graph transformers are being explored to address this need.

Future Directions:

- Higher-Order GNNs: Extending GNN architectures to capture higher-order interactions between nodes. Graph Transformers: Leveraging transformer models for graph data to enhance representational power.

Interpretability

Understanding the decision-making process of GNNs is crucial for their adoption in critical applications. Techniques like attention visualization, gradient-based methods, and node importance scores are being developed to interpret GNN predictions.

Future Directions:

- Explainable GNNs: Designing GNN models with built-in interpretability features.
- Interpretable Training Methods: Developing training techniques that enhance model transparency and explainability.

Challenges and Limitations

Despite the significant advancements and applications, GNNs face several challenges and limitations:

- Computational Complexity: GNNs can be computationally intensive, especially for large and dense graphs, limiting their practical applicability.
- Scalability Issues: Handling extremely large-scale graphs remains challenging, requiring efficient algorithms and distributed computing frameworks.
- Over-smoothing: Deep GNNs can suffer from over-smoothing, where node representations become indistinguishable after several layers, reducing model performance.
- Lack of Interpretability: The black-box nature of GNNs poses challenges in understanding their decision-making process, which is crucial for sensitive applications.
- Limited Expressiveness: Some GNN architectures struggle to capture long-range dependencies and complex interactions, necessitating further research into more expressive models.

Conclusion

Graph Neural Network have revolutionized the field of Deep Learning by providing powerful tools for modeling graph-structured data. They have demonstrated remarkable success across various domains, including social network analysis, bioinformatics, recommendation systems, and NLP. Despite their advantages, GNNs face challenges related to scalability, interpretability, and computational complexity. Ongoing research aims to address these challenges, exploring new architectures, efficient training techniques, and methods to enhance expressiveness and interpretability. The future of GNNs looks promising, with potential breakthroughs that could further expand their applications and impact.

2.3 Building Energy Management Systems

Chapter 3

Conclusion

Bibliography

Bizer, C., Heath, T., & Berners-Lee, T. (2023). Linked data - the story so far. In *Linking the world's information: Essays on tim berners-lee's invention of the world wide web* (1st ed., pp. 115–143). Association for Computing Machinery. https://doi.org/10.1145/3591366.3591378

Bonatti, P., Kirrane, S., Polleres, A., & Wenning, R. (2017). Transparent personal data processing: The road ahead. In S. Tonetta, E. Schoitsch & F. Bitsch (Eds.), *Computer safety, reliability, and security* (pp. 337–349). Springer International Publishing.

Bruna, J., Zaremba, W., Szlam, A., & LeCun, Y. (2013). Spectral networks and locally connected networks on graphs. http://arxiv.org/abs/1312.6203

Chen, J., Zhu, J., & Song, L. (2017). Stochastic training of graph convolutional networks with variance reduction. http://arxiv.org/abs/1710.10568

Chiang, W. L., Li, Y., Liu, X., Bengio, S., Si, S., & Hsieh, C. J. (2019). Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 257–266. https://doi.org/10.1145/3292500.3330925

Deborah, L. M., & van Harmelen Frank. (2004). Owl web ontology language overview. http://www.w3.org/TR/2003/PR-owl-features-20031215/

Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. https://github.com/mdeff/cnn_graph

Fernández, M., Cantador, I., López, V., Vallet, D., Castells, P., & Motta, E. (2011). Semantically enhanced information retrieval: An ontology-based approach. *Journal of Web Semantics*, *9*, 434–452. https://doi.org/10.1016/j.websem.2010.11.003

Fout, A., Byrd, J., Shariat, B., & Ben-Hur, A. (2017). Protein interface prediction using graph convolutional networks.

Francis, N., Green, A., Guagliardo, P., Libkin, L., Lindaaker, T., Marsault, V., Plantikow, S., Rydberg, M., Selmer, P., & Taylor, A. (2018). Cypher: An evolving query language for property graphs. *Proceedings of the ACM SIGMOD International*

Conference on Management of Data, 1433–1445. https://doi.org/10.1145/3183713.3190657

G. (Grigoris) Antoniou and Frank. Van Harmelen. (2008). *A semantic web primer*. MIT Press.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017). Neural message passing for quantum chemistry.

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs.

Hogan, A., Blomqvist, E., Cochez, M., D'Amato, C., Melo, G. D., Gutierrez, C., Kirrane, S., Gayo, J. E. L., Navigli, R., Neumaier, S., Ngomo, A. C. N., Polleres, A., Rashid, S. M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., & Zimmermann, A. (2021). Knowledge graphs. *ACM Computing Surveys*, *54*. https://doi.org/10.1145/3447772

Jin, W., Barzilay, R., & Jaakkola, T. (2018). Junction tree variational autoencoder for molecular graph generation.

Kapanipathi, P., Abdelaziz, I., Ravishankar, S., Roukos, S., Gray, A., Astudillo, R., Chang, M., Cornelio, C., Dana, S., Fokoue, A., Garg, D., Gliozzo, A., Gurajada, S., Karanam, H., Khan, N., Khandelwal, D., Lee, Y.-S., Li, Y., Luus, F., ... Yu, M. (2020). Leveraging abstract meaning representation for knowledge base question answering. http://arxiv.org/abs/2012.01707

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. http://arxiv.org/abs/1609.02907

Paulheim, H. (2017). Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, *8*, 489–508. https://doi.org/10.3233/SW-160218

Pérez, J., Arenas, M., & Gutierrez, C. (2009). Semantics and complexity of sparql. *ACM Transactions on Database Systems*, *34*. https://doi.org/10.1145/1567274.1567278

Pujara, J., Miao, H., Getoor, L., & Cohen, W. (2013). Knowledge graph identification. *The Semantic Web–ISWC 2013: 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I 12*, 542–557.

Sahu, S. K., Christopoulou, F., Miwa, M., & Ananiadou, S. (2019). Inter-sentence relation extraction with document-level graph convolutional neural network. http://arxiv.org/abs/1906.04684

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The graph neural network model. *IEEE Transactions on Neural Networks*, *20*, 61–80. https://doi.org/10.1109/TNN.2008.2005605

Singhal, A., et al. (2012). Introducing the knowledge graph: Things, not strings. *Official google blog*, *5*(16), 3.

Tchechmedjiev, A., Fafalios, P., Boland, K., Gasquet, M., Zloch, M., Zapilko, B., Dietze, S., Todorov, K., & Zloch, M. (2019). Claimskg: A knowledge graph of fact-checked claims, 309–324. https://doi.org/10.1007/978-3-030-30796-7_20

Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods and applications.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2017). Graph attention networks. http://arxiv.org/abs/1710.10903

Wang, Q., Mao, Z., Wang, B., & Guo, L. (2017). Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, *29*, 2724–2743. https://doi.org/10.1109/TKDE.2017.2754499

Wang, X., He, X., Wang, M., Feng, F., & Chua, T. S. (2019). Neural graph collaborative filtering. *SIGIR 2019 - Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 165–174. https://doi.org/10.1145/3331184.3331267

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*, 4–24. https://doi.org/10.1109/TNNLS.2020.2978386

Yasunaga, M., Ren, H., Bosselut, A., Liang, P., & Leskovec, J. (2021). Qa-gnn: Reasoning with language models and knowledge graphs for question answering. http://arxiv.org/abs/2104.06378

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., & Leskovec, J. (2018). Graph convolutional neural networks for web-scale recommender systems. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 974–983. https://doi.org/10.1145/3219819.3219890

Zeng, H., Zhou, H., Srivastava, A., Kannan, R., & Prasanna, V. (2019). Graphsaint: Graph sampling based inductive learning method. http://arxiv.org/abs/1907.04931

Zhang, W., Deng, S., Chen, M., Wang, L., Chen, Q., Xiong, F., Liu, X., & Chen, H. (2021). Knowledge graph embedding in e-commerce applications: Attentive reasoning, explanations, and transferable rules. *ACM International Conference Proceeding Series*, 71–79. https://doi.org/10.1145/3502223.3502232

List of Figures

List of Tables

The acknowledgements and the people to thank go here, don't forget to include your project advisor…