AGHE - Approach for Generating Enhanced Heterogeneous Embeddings from Heterogeneous Graphs

Silvio F. Angonese[1], **Renata Galante**[1]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{sfangonese, galante}@inf.ufrgs.br

Abstract. *Embeddings represent a viable solution to address the challenge of data and information generation in Heterogeneous Graphs. This paper presents the approach for generating and processing heterogeneous embeddings (AGHE), which are built from various data types such as text, images, and subgraphs embedded in nodes. The AGHE comprises several steps, from graph creation to generating embeddings using metapaths and aggregating information from neighboring nodes. The experiments conducted investigated the performance of Recommender System tasks applied to the generated embeddings: node-local text-based, neighbor-aggregated text-based, metapath-based, and text and metapath composition. Results underscore the effectiveness in representing data heterogeneity in Deep Learning systems based on Heterogeneous Graph.*

1. Introduction

Nowadays, we have a vast expanse of unexplored and underutilized data, representing a disconnected and unseen world amidst the abundance of information. Heterogeneous Graph (HG) is an example of a powerful data structure able to hold valuable insights about the domain modeled, thus exploring HG, it is possible to navigate this uncharted territory, uncovering valuable insights and discovering hidden knowledge that can transform the understanding and utilization of data. Turning a digital desert into a fertile and shareable land of data and information. Representation Learning (RL) using Deep Learning (DL) models can be used to complement, or even replace, traditional approaches to data generation with neural network approaches. An example lies in downstream applications like Recommender System (RecSys), where DL can solve the intricate relationships within the data itself, providing high-quality data to achieve superior recommendations [Wang et al. 2015].

Most graphs seem not to concern themselves with complete information about their components, they typically focus only on the relationships between their nodes. HG has the potential to become an important dataset embedded within a much larger collection of data and information because HG has different types of nodes that can contain not only plain text data. By expanding to other data types, such as images and subgraphs, node embeddings can be generated from this varied information using DL techniques to extract information from images and plain text, thereby turning them into embeddings saved within the nodes. This mechanism can leverage downstream applications to enhance RL using graph embedding.

The work [Hamilton et al. 2017c, Ying et al. 2018a] introduces the generation of embeddings based on the neighboring nodes, increasing their expressiveness and recommendation performance. Meanwhile, the works of [Zhang et al. 2019, Wu et al. 2022] evaluate the heterogeneity of data types embeddings, opening up a new perspective. [Dong et al. 2017, Wang et al. 2023] propose a new approach for generating embeddings from metapaths, capturing semantics based on node relationships. Vision GNN [Han et al. 2022] and Superpixel Image Classification [Avelar et al. 2020] are examples of the few works where the graph is an image and the nodes are parts of it, demonstrating the feasibility of having images as nodes. None of the related work addresses the topic of using heterogeneous graphs with also heterogeneous data types, mainly such as images and subgraphs, with the potential to extract information from images through DL techniques, especially Autoencoders, saving them as heterogeneous embeddings.

This paper aims to propose an approach capable of generating heterogeneous embeddings through the processing of texts, images, and subgraphs represented in the nodes of HG, thus enhancing the performance of downstream applications like RecSys. The approach is named AGHE - Approach for Generating Enhanced Heterogeneous Embeddings from Heterogeneous Graphs, which is separated into five steps: the first one is the creation of the HG with texts, images, and subgraphs associated with the nodes; the second one is the generation of embeddings using specialized Autoencoders; the third one is the generation of embeddings from metapaths and neighboring nodes; the fourth one is the creation of recommendation data based on the generated embeddings; and the fifth and final step is the reconstruction of the recommended graph and its provision as a dataset. The experiments were conducted on an HG with 463 nodes, 497 edges, and 13 metapaths, applying the main RecSys tasks to the different generated embeddings, demonstrating the performance evolution from a baseline without embeddings to achieve the best performance with the heterogeneous composition of Features and Metapath embedding. These results underscore the effectiveness of this approach in representing data heterogeneity in DL systems based on HG. Therefore, the main contribution of this approach is to generate embeddings from many data types, providing a graph as a rich dataset for downstream applications to enhance their performance.

The remainder of this paper is organized as follows: Section 2 presents the related work. Section 3 conceptualizes the background techniques applied in this paper. Section 4 presents the proposed approach. Section 5 conducts some experiments while Section 6 exposes the conclusions and future works.

2. Related Work

The Homogeneous Graph can be traced back to generate data embedding from node features based on random walk approach [Hamilton et al. 2017c, Ying et al. 2018a] improving the node expressivity. Based on these studies we build our aim, which is to propose an approach using HG with heterogeneous data type embeddings. More relevant and close to our aims is the recent work [Zhang et al. 2019] which includes the definition of Heterogeneous Graph Neural Network (HetGNN) with the processing of embedding. This paper contributes by highlighting that heterogeneity lies not only in the node type but also in the data type of the embeddings, hence our work allows (e.g. in a Family graph) the node type to be a Pet and the data embedding could be the image of the family pet. The survey [Wu et al. 2022] shows Graph Neural Networks (GNN) have been widely used in downstream applications essentially because graph structure and GNN have superiority in graph representation learning, citing GraphSAGE [Hamilton et al. 2017a] as an important work regarding generating node embedding from node feature information.

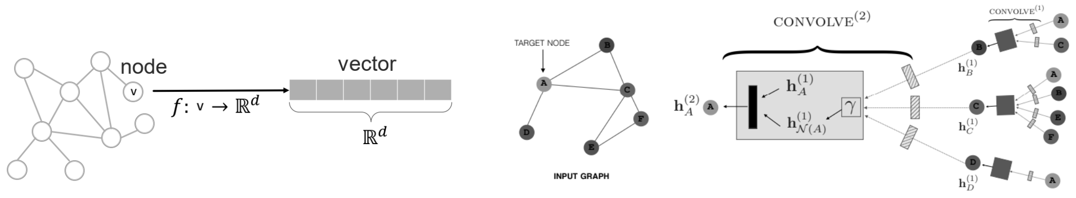
MetaPath2Vec [Dong et al. 2017, Wang et al. 2023] is another crucial aspect of this research due to its ability to capture the structure of HG, guiding random walks to generate sequences of heterogeneous nodes with rich semantics. Hence, metapath plays an important role in our work capturing vital information by leveraging the relationships among heterogeneous nodes, transforming it into a form of node embedding. Vision GNN paper [Han et al. 2022] and Superpixel Image Classification [Avelar et al. 2020] are other sources of inspiration that illustrate image representation in the form of a graph. In this context, each node corresponds to a distinct part of the same image, implying that every node encapsulates an image. Adopting a similar conceptualization, we can extend this idea to employ an image for representing node content. Therefore, both the previously mentioned study and our work share a commonality in utilizing images as nodes.

HG can be integrated with some applications. However, one usually needs to carefully consider two factors: the first is how to construct HG for a specific application, and the second is what information or domain knowledge should be incorporated into a HG to ultimately benefit the application [Wang et al. 2023]. In the RecSys, the interaction between the user and items can be naturally modeled as a HG with two types of nodes. Therefore, RecSys is a typical scenario that widely uses heterogeneous information. Moreover, other types of information, such as social relationships and personal data, can also be easily introduced into HG [Shi et al. 2015], thus applying heterogeneous graph embedding to recommendation applications is an important research field [Wang et al. 2023]. Hence, this work uses recommendations as an assessment to validate the value of performance by employing various types of node embeddings from the HG.

3. Conceptualization

This section aims to present techniques in DL that can be combined to formulate an approach that facilitates the generation and extraction of relevant information in scenarios where data representation has a significant impact on the application.

Heterogeneous Graph and Embeddings. HG nodes and edges can be of different types, e.g., the graph encoding the relationship between Person and their Cars and Pets [Fu et al. 2020]. The challenge of the heterogeneous graph representation learning is to figure out the information of nodes from it and their neighborhoods, which makes the aggregated embedding more powerful [Zhang et al. 2019, Wang et al. 2019, Jin et al. 2021]. The central problem in DL on graphs is finding a way to incorporate information about graph structure into DL models. From this perspective, the challenge is that there is no straightforward way to encode this high-dimensional, non-Euclidean information about graph structure into a feature vector [Hamilton et al. 2017c]. Heterogeneous graph embedding aims to learn a function $f : \mathbb{V} \to \mathbb{R}^d$ that embeds the nodes v into a low-dimensional Euclidean space with $d \ll |V|$ shown by Fig. 1(a) [Wang et al. 2023]. Thus, graph embedding is the transformation of property graphs to a vector or a set of vector spaces. Embedding should capture the graph topology, node features, node-to-node relationship, and other relevant information about graphs, subgraphs, and nodes. The similarity of embedding between nodes indicates their similarity in the network, i.e., both nodes are close to each other, connected or not by an edge, potentially used for any kind of prediction [Hamilton et al. 2017b, Rozemberczki et al. 2020].



(a) Embedding is saved into a dense vector space.

(b) Aggregated embeddings.

Figura 1. Embeddings representation and aggregation.

The basic idea behind node embedding approaches is to use dimensionality reduction techniques to figure out the high-dimensional information about the neighborhood of nodes into a dense vector embedding [Perozzi et al. 2014, Hamilton et al. 2017b]. Neural networks like GraphSAGE [Ying et al. 2018a, Zhang et al. 2019] can generate individual node embedding by passing, transforming, and aggregating [Zhiyuan Liu 2020, Gilmer et al. 2017], node feature information across the graph [Ying et al. 2018b]. Fig. 1(b) shows a small example of an input graph and two neural network layers that compute the embedding, $h_A^{(2)}$ of node A using the previous layer representation $h_A^{(1)}$ from node A and its neighborhoods $N(A)$ nodes B, C, D.

Thus, is able to generalize the embedding approach as models that learn graph embedding with methods inspired by the Skip-Gram model. DeepWalk learns embedding via the prediction of the local neighborhood of nodes, sampled from random walks on the graph [Perozzi et al. 2014].

Metapath2vec. It is a heterogeneous graph embedding model that formalizes metapath based on random walk to construct the heterogeneous neighborhood of a node and then leverages a heterogeneous Skip-Gram model to perform node embedding. Maximizing the probability of preserving both the structures and semantics of a given heterogeneous network, being able to learn desirable node representations in heterogeneous networks [Dong et al. 2017]. Fig. 2 shows an example of the application of Metapath2vec architecture where the purpose is to generate paths that are able to capture both the semantics and structural correlations between different types of nodes. Under the metapath scheme CPE the walker is biased towards Person node P given its previous step on a Car node C, and its next step is a Pet node E, following the semantics of this metapath. Metapath2vec encourages all types of nodes to appear in metapath definition [Sun and Han 2012].

Autoencoders. They are useful for incorporating structural graph information from nodes, providing specific implementations. Autoencoder is a kind of neural network architecture that imposes a bottleneck on the network that forces a compressed knowledge representation of the original input. If the input resources were independent of each other, the compression and subsequent reconstruction would be a very difficult task. Thus, AE are neural networks that aim to copy their input to their output, compressing the input in a latent space representation, called Encoder $h = f(x)$, and after that, rebuilding the output through this representation, called Decoder $r = g(h)$ [Bank et al. 2021, Kipf and Welling 2016, Wu et al. 2019]. Learning a representation via

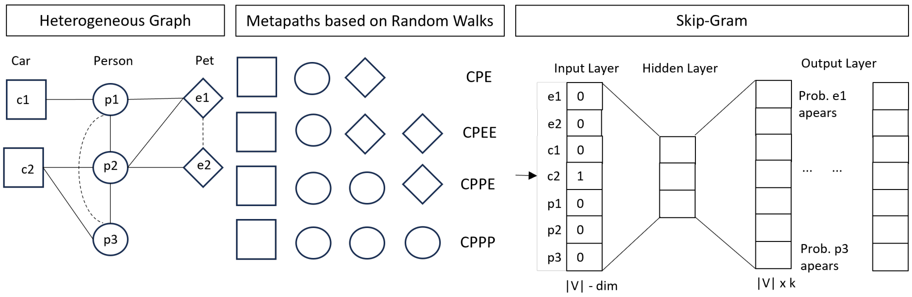


Figura 2. Use case of Metapath2vec architecture.

Autoencoder can be used for various applications, due to the fact that different types of Autoencoder may be modified or combined to form new models for various applications such as generative models, classification, clustering, anomaly detection, recommendation, dimensionality reduction, and capture information [Bank et al. 2020].

4. AGHE - Approach for Generating Enhanced Heterogeneous Embeddings

The primary objective of this approach is to generate embeddings from many data types, providing a graph as a rich dataset for downstream applications to enhance their results. The challenge at hand is how to generate data embeddings to enhance node semantics. Fig. 3 shows the pipeline designed to process embeddings of heterogeneous data types from HG, resulting in an enriched graph with features and heterogeneous embeddings. The pipeline shows the big picture regarding the complete process, where the input is the entire heterogeneous graph $HG = (V, E, OH_V)$ where V is a set of nodes, E is a set of edges, O is a set of objects embedded in node $v \in V$ that describe and representing it, thus OH_V is heterogeneous data type embeddings and $OH \in \{text, image, subgraph\}$. The tasks are to design some Autoencoders $\mathcal{AE}_\Theta$ with Θ parameters able to process each node heterogeneous data features OH_V generating as output just text features and respective embeddings O_V attached on each node v. At the end of the pipeline, a Decoder $\mathcal{D}_\Theta$ where Θ parameter is a dense vector graph representation, is able to rebuild the graph G with news and standardized data type features and within different types of embedding. HG has different types of nodes, and each node has its data features, typically based on the text data type representation. Expanding the node data features to other data types, such as images, and subgraphs embedded in the node, the RL could leverage applications based on graph embeddings as input datasets.

The first challenge associated with the Input step is to model some business cases as a graph. The second one associated with the Processing Data step lies in obtaining node features from images and subgraphs. Thus, our approach addresses this scenario by using images and subgraphs as a data source to generate node data embeddings. Hence, node embedding can originate from local information, subgraph information, or be coming from metapaths and random walk information associated with the node. Given this challenge, these steps are proposed for generating and processing heterogeneous embeddings.

4.1. Steps of the AGHE Approach

Fig. 4 illustrates the main steps and capabilities employed in creating a heterogeneous graph and generating vectorial numeric embeddings through text and image processing.

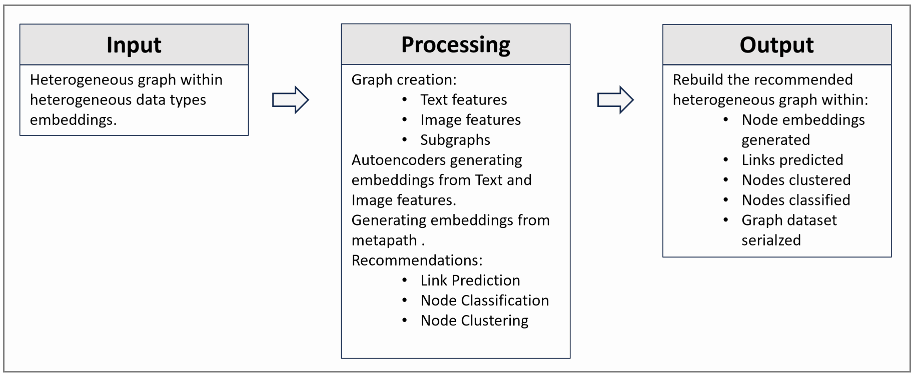


Figura 3. Pipeline for generating and processing features and embeddings.

A heterogeneous graph is created by integrating text, image, and subgraph data types, employing vectorial numeric representations such as embeddings generated through specialized Autoencoders for processing a set of words and images. Additionally, Metapath2vec produces vectorial numeric representations through metapath walks, which are utilized to generate embeddings, thereby offering a rich dataset for downstream applications like RecSys. This enhances the data graph, delivering a novel heterogeneous graph with embedded data and recommendations. Our approach provides valuable data about nodes, which proves useful for downstream applications. Each step shown in Fig. 4 will be detailed in the sequence, addressing the questions “What is done?” and “What is delivered”?

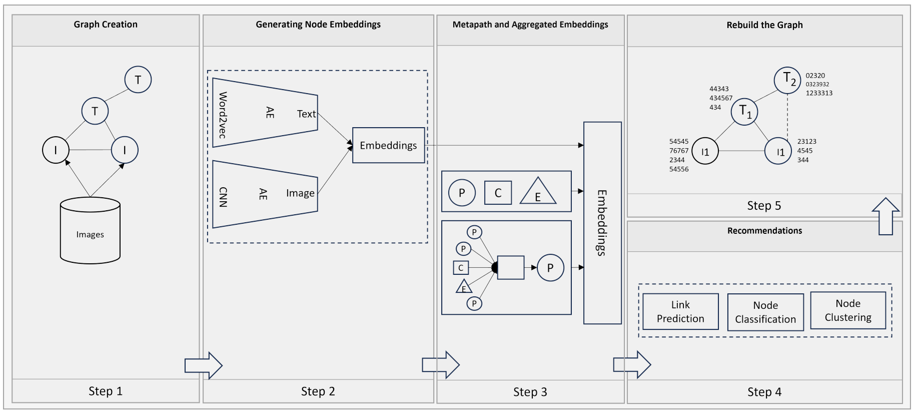


Figura 4. Main steps of the approach AGHE.

Step 1 - Graph Creation. There are two main implementations, the first one regards adding nodes and defining their types, characterizing a heterogeneous graph. Subsequently, should attach the node features with a set of short node descriptions with simple words. In this context, edge features are not considered as part of this research. The second one maps relationships between nodes through adding edges to the heterogeneous graph, providing navigation between the nodes. Another important aspect related to the creation of the heterogeneous graph is the association of each node of the type image with its image, which involves uploading the node image from an image database provided by the application that understands the graph. The entire process can be replaced by ingesting the graph components from the graph database, regardless of whether they exist. As a result, the HG is created and can be used in subsequent steps. It is a critical step because all the other steps and results depend on it.

Step 2 - Generating Text Node Embeddings. It is responsible for generating text node embeddings from images, transforming the entire set of node text features into numeric vectorial embeddings, and saving them into each node. Where the Image Autoencoder takes, as input, the image from nodes whose types are images. The Image Autoencoder implements a Convolutional Neural Network (CNN) to extract image classification and certain characteristics from the processed image, generating text features that will later be saved into each node. Thus, the entire graph has text node features, coming from the original text features or the node images. This step is carried out by the Text Autoencoder, which employs the Word2Vec algorithm to process the complete set of text feature nodes generating respective node embeddings. The objective is to incorporate the semantics of text features into continuous numeric vector space and subsequently learn a compact representation of these vectors, with the aim of dimensionality reduction while preserving the semantic information of words. As a result, the heterogeneous graph contains text features across all nodes, represented as numeric vectorial features or embeddings in the entire set of graph nodes.

Step 3 - Metapath and Aggregated Node Embeddings. It aims to traverse the nodes in the defined metapaths, capturing complex semantic patterns in the relationships between different types of nodes using the MetaPath2Vec algorithm. This technique tries to learn a compact latent representation of the graph nodes while preserving the semantic structure and meaningful relationships between them. Based on the HG used in the experiments, there are three different types of nodes: Car, Person, and Pet, which generate a set of metapaths $\{(Car, Person), (Pet, Person), (Car, Person, Person), (Pet, Person, Pet), (Person, Person, Car), (Person, Person, Pet), (Person, Person, Person)\}$. After the processing of metapaths, it produces a new numeric vectorial embedding for each node. This representation captures semantics from nodes based on each metapath and their direct or indirect neighbors, depending on the value of the random walk size parameter. Using the random walk approach, aggregated node embeddings are generated by merging local embeddings with embeddings from neighboring nodes.

Step 4 - Graph Enhancing with RecSys Tasks. Enhances the data graph by aggregating new data into the nodes, including predicted links, classified node, and clustered node, based on the embeddings generated in the previous steps. This step delivers a heterogeneous graph enhanced with new predicted edges and more semantic information attached to its nodes.

Step 5 - Rebuild the Recommended Graph. It is the final step, aiming to deliver the recommended graph with aggregating features and embeddings from different data types, predicted links, node classified, and clustered. This step provides for downstream applications a graph enriched by more informative nodes. Consequently, the resulting graph is ready to be utilized in applications that require a rich data source. Additionally, this step produces a JSON file within the final heterogeneous graph, serialized and published as a public dataset, as shown in Fig. 5.

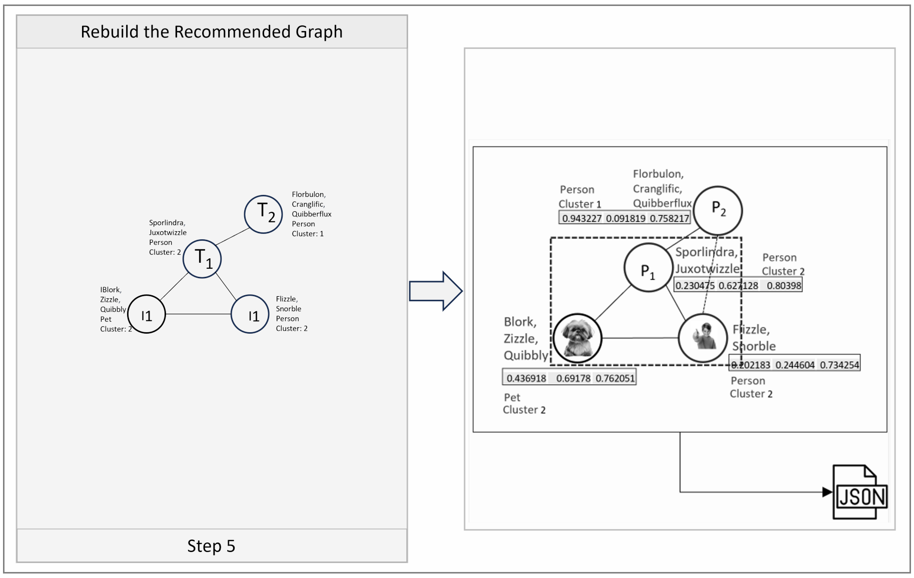


Figura 5. The final step of the approach AGHE.

5. Experiments

This section intends to execute some experiments guided by the methodology based on the approach proposed for generating enhanced heterogeneous features and embeddings. The experiments aim to validate of the assumption that enriching the heterogeneous graph within heterogeneous embeddings, generated from processing the available data within the graph, could enhance the performance of downstream applications.

Methodology. Fig. 6 shows the main pipeline as part of the methodology used to execute the entire set of experiments. Essentially, it involves the generation of features and embeddings from nodes, and the validation of RecSys performance based on each type of node embedding.

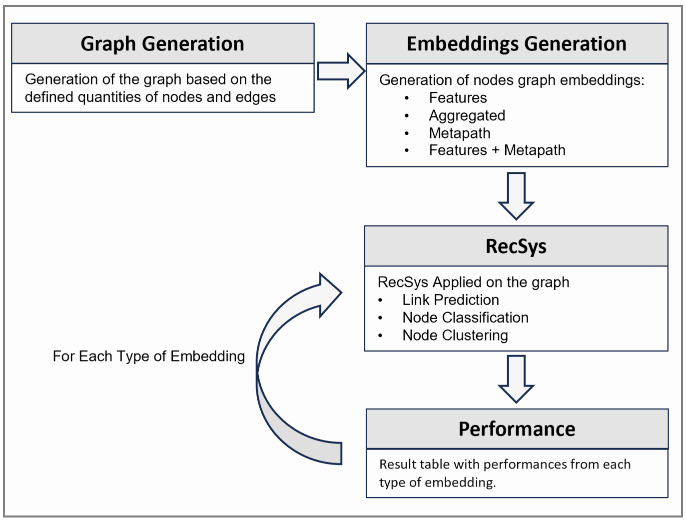


Figura 6. Steps of experiments methodology.

Heterogeneous Graph Model. Fig. 7 shows an example of a HG within features of different data types, which represents the model used in the experiments. The graph includes many node types such as Person, Car, and Pet. Each node has its own features, which can be in plain text, images, or subgraphs like the family of Mary embedded in its node. The real graph used in the experiments has exactly the same model shown in Fig.7 only with a larger number of graph elements, i.e., 463 nodes, 497 edges, and 13 metapaths. Steps 2 and 3 generated five heterogeneous embeddings to support the experiments:

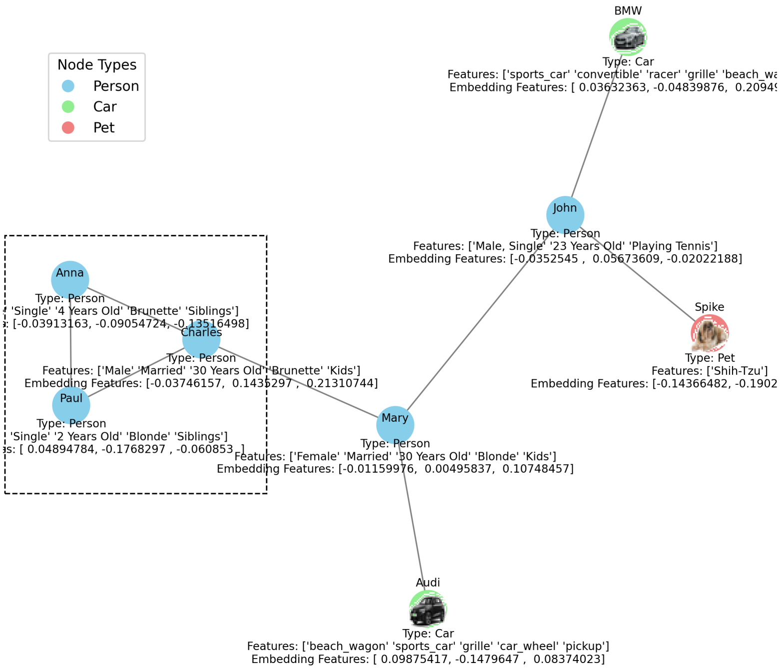


Figura 7. Heterogeneous graph model used in the experiments.

The column "Embeddings" in Table. 1 means:

- *No Embeddings.* Represents the performance of RecSys calculated from the same heterogeneous graph but without any features and embeddings. Thus, it serves as a baseline to compare the performance with other types of embedding;
- *Features.* Using the original text features of nodes, as well as the class and characteristics of images generated from the Image Autoencoder. A set of text features and their respective embeddings are generated and saved into each node;
- *Aggregated.* Each node aggregates its text features with its neighbors through the random walk approach generating its corresponding embeddings;
- *Metapaths.* Based on the metapaths defined in Step 3, the metapath embedding is generated by traversing metapaths to capture the relationship and semantic information of nodes and saving it into the central node.
- *Features_Metapaths.* Based on the features and metapaths embedding was created a specific embedding that captures the local node semantic by its text features, and the relationship between the nodes through metapath.

Results and Analysis of the Experiments. The assumption that enriching the heterogeneous graph within heterogeneous embeddings, generated from processing the available data within the graph, could enhance the performance of downstream applications, such as RecSys, was validated. Table 1 illustrates the evolution of this performance starting with Features embeddings and progressing to the best performance achieved by the combination of Features and Metapaths node embeddings, as indicated by the Accuracy and F-1 Score metrics. Link Prediction was calculated using Cosine Similarity with a threshold of 80% and using a specific Person "Erika" as the node target. Link prediction from "No Embeddings" was calculated using the Jaccard algorithm based on the intersection and union set operation $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$. Link prediction count using Features embedding is so high, it can indicate data homogeneity with lack of distinctive features. Although it varies based on certain factors, predictions from Metapaths and Features_Metapaths may be deemed more reliable. Node clustering already reveals similar cluster distributions independent of the embedding used, except Aggregated embedding. "No Embeddings" Clusters were calculated using the Louvain algorithm based on the nodes community where nodes without community identified have no cluster assigned.

Tabela 1. Performance of Different Types of Graph Embeddings

Nodes: 463, Edges: 497, Metapaths: 13								
	Prediction	**Classification**				**Clusters**		
Embeddings	**Links**	**Acc**	**F-1**	**Correct**	**Incorrect**	**C0**	**C1**	**C2**
No Embeddings	7	44.71%	61.79%	207	256	36	32	32
Features	459	50.11%	66.76%	232	231	256	20	187
Aggregated	114	64.36%	78.32%	298	165	24	346	93
Metapaths	7	65.44%	79.11%	303	160	213	100	150
Features_Metapaths	7	**71.06%**	**83.08%**	329	134	213	100	150

6. Conclusion

This paper presents an Approach for Generating Heterogeneous Embeddings (AGHE) from HG, enhancing the graph as a dataset and improving downstream applications, thus demonstrating the high RL achieved. This enhances the performance of RecSys, which is used as a system reference in this paper. Developing effective and efficient graph analytics from information embedding, can greatly help to better understand complex graphs, and provide innovative solutions for data models. Based on the obtained results, we believe that the conducted study can open doors for its use in different downstream applications as demonstrated in this work. Some specific evaluations can be achieved: the choice of appropriate embeddings plays a crucial role in the performance of downstream tasks. The results indicate that a one-size-fits-all embedding approach is not necessarily the best for all tasks and datasets.

It is important to explore a variety of embedding generation techniques and consider the unique characteristics of the data and tasks at hand. Combining information from different sources, such as node features and structural relationships defined by metapaths, can lead to more comprehensive and informative node representations in the graph. This highlights the importance of exploring hybrid approaches that combine multiple types of embeddings to improve the performance of RecSys tasks and other downstream applications. However, it is important to note that combining feature and metapath embeddings may increase the dimensionality of the data, which can lead to computational and generalization challenges, especially in large and complex HG. Hence, it is important to strike a balance between model complexity and performance in terms of efficiency and generalization capability. Additionally, careful validation of the results is crucial, including evaluation of separate test datasets and comparison with benchmarks and state-of-the-art approaches to ensure the reliability and robustness of the findings, whether it is feasible.

Future works include evaluating recommendation performance using both a tabulated dataset and the same dataset modeled as a HG with heterogeneous embeddings; creating a heterogeneous embedding combining Features Aggregated and Metapaths; deeply analyzing the architecture and methods of algorithm; applying the proposed approach to a popular dataset as a performance baseline; evaluating the application of AGHE to different downstream applications; and aggregating edge data features into AGHE.

Referências

Avelar, P. H. C., Tavares, A. R., da Silveira, T. L. T., Jung, C. R., and Lamb, L. C. (2020). Superpixel image classification with graph attention networks. In *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 203–209.

Bank, D., Koenigstein, N., and Giryes, R. (2020). Autoencoders. *CoRR*, abs/2003.05991.

Bank, D., Koenigstein, N., and Giryes, R. (2021). Autoencoders.

Dong, Y., Chawla, N. V., and Swami, A. (2017). Metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 135–144, New York, NY, USA. Association for Computing Machinery.

Fu, X., Zhang, J., Meng, Z., and King, I. (2020). Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of The Web Conference 2020*, WWW '20, page 2331–2341, New York, NY, USA. Association for Computing Machinery.

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

Hamilton, W. L., Ying, R., and Leskovec, J. (2017a). Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 1025–1035, Red Hook, NY, USA. Curran Associates Inc.

Hamilton, W. L., Ying, R., and Leskovec, J. (2017b). Inductive representation learning on large graphs. *CoRR*, abs/1706.02216.

Hamilton, W. L., Ying, R., and Leskovec, J. (2017c). Representation learning on graphs: Methods and applications. *CoRR*, abs/1709.05584.

Han, K., Wang, Y., Guo, J., Tang, Y., and Wu, E. (2022). Vision gnn: An image is worth graph of nodes. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 8291–8303. Curran Associates, Inc.

Jin, D., Huo, C., Liang, C., and Yang, L. (2021). Heterogeneous graph neural network via attribute completion. In *Proceedings of the Web Conference 2021*, WWW '21, page 391–400, New York, NY, USA. Association for Computing Machinery.

Kipf, T. N. and Welling, M. (2016). Variational graph auto-encoders.

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). DeepWalk. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM.

Rozemberczki, B., Davies, R., Sarkar, R., and Sutton, C. (2020). Gemsec: Graph embedding with self clustering. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '19, page 65–72, New York, NY, USA. Association for Computing Machinery.

Shi, C., Zhang, Z., Luo, P., Yu, P. S., Yue, Y., and Wu, B. (2015). Semantic path based personalized recommendation on weighted heterogeneous information networks. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, page 453–462, New York, NY, USA. Association for Computing Machinery.

Sun, Y. and Han, J. (2012). *Mining Heterogeneous Information Networks: Principles and Methodologies*. Morgan amp; Claypool Publishers.

Wang, H., Wang, N., and Yeung, D.-Y. (2015). Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 1235–1244, New York, NY, USA. Association for Computing Machinery.

Wang, X., Bo, D., Shi, C., Fan, S., Ye, Y., and Yu, P. S. (2023). A survey on heterogeneous graph embedding: Methods, techniques, applications and sources. *IEEE Transactions on Big Data*, 9(2):415–436.

Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., and Yu, P. S. (2019). Heterogeneous graph attention network. In *The World Wide Web Conference*, WWW '19, page 2022–2032, New York, NY, USA. Association for Computing Machinery.

Wu, S., Sun, F., Zhang, W., Xie, X., and Cui, B. (2022). Graph neural networks in recommender systems: A survey. *ACM Comput. Surv.*, 55(5).

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. (2019). A comprehensive survey on graph neural networks. *CoRR*, abs/1901.00596.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. (2018a). Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '18, page 974–983, New York, NY, USA. Association for Computing Machinery.

Ying, R., You, J., Morris, C., Ren, X., Hamilton, W. L., and Leskovec, J. (2018b). Hierarchical graph representation learning with differentiable pooling. *CoRR*, abs/1806.08804.

Zhang, C., Song, D., Huang, C., Swami, A., and Chawla, N. V. (2019). Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '19, page 793–803, New York, NY, USA. Association for Computing Machinery.

Zhiyuan Liu, J. Z. (2020). *Introduction to Graph Neural Networks*. Springer Cham.