Enhancing NLP through GNN-Driven Knowledge Graph Rewiring and Document Classification

Alex Romanova
Independent Researcher
McLean, VA, USA
sparkling.dataocean@gmail.com

***Abstract*—This study explores the application of Graph Neural Networks (GNNs) in Natural Language Processing (NLP), with a specific focus on Knowledge Graph Rewiring and Document Classification. Leveraging the distinct capabilities of GNNs, we aim to advance text analysis by revealing hidden semantic connections and improving recommendation systems. Our methodology introduces a novel approach for constructing and analyzing semantic graphs, employing GNN-driven techniques to uncover complex patterns and relationships within text data, often missed by traditional methods.**

We conduct a comparative analysis of GNN models to detect and classify intricate relationships in knowledge graphs derived from biographies of modern art artists. This research underscores GNNs' potential to not only enhance the accuracy and depth of classification tasks but also to provide a deeper understanding of text construction and interpretation. We critically examine the effectiveness of GNNs in managing noise and identifying outliers, highlighting the need for continued advancements in model refinement.

Our findings demonstrate GNNs' ability to significantly improve data analysis through knowledge graph rewiring and document classification. Emerging as potent tools for delivering nuanced, context-aware insights, GNNs represent a major progression in NLP and beyond. By pioneering in knowledge representation and revealing deep semantic connections, this research paves the way for the broader application of GNNs in fields requiring detailed text analysis and sophisticated knowledge graph interpretation.

I. Introduction

2012 was a breakthrough year for deep learning and knowledge graphs. In that year Convolutional Neural Network (CNN) image classification gained prominence with the introduction of AlexNet [1] and concurrently Google introduced knowledge graphs. This breakthrough marked the superiority of CNN techniques over previous machine learning approaches across diverse domains [2]. Knowledge graphs revolutionized data integration and management by enhancing products with intelligent and magical capabilities [3].

For several years, deep learning and knowledge graphs progressed in parallel paths. CNN deep learning excelled at processing grid-structured data but faced challenges when dealing with graph-structured data. Graph techniques effectively represented and reasoned about graph structured data but lacked the powerful capabilities of deep learning. In the late 2010s, the emergence of Graph Neural Networks (GNN) bridged this gap and combined the strengths of deep learning and graphs. GNN became a powerful tool for processing graph-structured data through deep learning techniques [4].

Graph Neural Networks employ deep learning to effectively process and interpret graph-structured data, excelling at unraveling complex entity interactions and the sophisticated structures and dynamics prevalent in graphs. These models stand out for their broad utility in graph analysis tasks, such as GNN Node Classification, GNN Link Prediction, and GNN Graph Classification. In node classification, GNNs assign labels to nodes by considering both local and overarching graph contexts, aiming to pinpoint a node's category through its direct and indirect connections. For link prediction, the models estimate the probability of a connection between node pairs, leveraging node attributes and the graph's topology to reveal unseen or potential relationships. Graph classification further extends GNNs' applicability by categorizing entire graphs or subgraphs based on structural attributes and the features of nodes and edges, thus enabling the differentiation of graph types through holistic analysis. The versatility of Graph Neural Networks (GNNs) extends their impact across diverse sectors, from financial markets to healthcare, supply chain, energy management, and entertainment. By analyzing complex data networks, GNNs play a crucial role in unveiling insights and optimizing processes across various industries, highlighting their wide-ranging applicability and transformative potential.

This study investigates two GNN techniques for analyzing text data: GNN Link Prediction and GNN Graph Classification. GNN Link Prediction aims to identify hidden connections within knowledge graphs, which can reveal new insights by uncovering relationships not immediately apparent. GNN Graph Classification seeks to provide semantic insights at a detailed level, allowing for a deeper understanding of the text's content and structure. By integrating these GNN approaches with semantic graph and Natural Language Processing (NLP), this research aims to demonstrate how GNNs can be applied to text data to extract deeper semantic meaning and discover hidden patterns.

To build a semantic graph, we first identify co-located word pairs in the text to serve as nodes, linking them through shared words to form edges. This approach not only traces direct links but also maps wider semantic networks, effectively illustrating how concepts are interwoven within the text. The constructed semantic graph then serves as a detailed map of textual relationships, enabling in-depth analysis that uncovers hidden patterns and connections, thereby enriching our comprehension of the text's complexity.

The method for constructing semantic graphs from co-located word pairs, was introduced in our previous study [5]. As the input data at that study we used wikipedia articles about biographies of modern art artists. On top of semantic graph in that study we used GNN Link Prediction to rewire knowledge graphs and reveal hidden relationships.

Expanding upon our earlier work, this study integrates GNN Graph Classification to deepen our exploration of textual relationships, thereby enhancing our understanding of the subtle nuances within text. We maintain our focus on biographies of modern art artists, employing the proven technique of constructing semantic graphs from co-located word pairs. Through this enriched analytical lens, we seek to uncover the nuanced connections in text, offering fresh perspectives on the intricate relationships between words and their meanings.

In our earlier study [6], we investigated the connections between modern art artists using their biographies and art movements. This work's insights will help us make detailed comparisons of semantic similarities using GNN Graph Classification in our current research.

This study leverages GNN Link Prediction and GNN Graph Classification for analyzing text documents through semantic graphs. The utilization of these methods aims to exploit their strengths in unveiling complex semantic connections and categorizing text data intricately. This approach promises a deeper, more nuanced analysis of textual relationships, enhancing our ability to interpret and understand the underlying semantic frameworks within documents.

GNN Link Prediction enhances the traditional binary approach to graph connections by introducing a nuanced spectrum of connection strengths, thus facilitating a more detailed and effective rewiring of knowledge graphs. In this study we are using Link Prediction to leverage aggregated artist output vectors and cosine similarities between vectors for precise link prediction. This advanced methodology allows for an in-depth exploration of complex relationships within text data, providing a comprehensive tool for mapping and understanding the intricate connections that define knowledge graphs.

In this study, we explore the potential of GNN Graph Classification in document classification, leveraging its proficiency in categorizing small graphs. Our approach involves creating small, labeled graphs centered on nodes with high betweenness centrality for input into GNN Graph Classification models. This centrality metric is crucial for identifying nodes that act as key connectors within the network, ensuring our models focus on the most significant and influential connections within the text. By adopting this methodology, we aim to uncover complex patterns and relationships in textual data, providing insights that traditional analysis methods might overlook.

One of the challenges in GNN Graph Classification models lies in their sensitivity, which is vital for distinguishing class differences but complex when it comes to identifying outliers. Nonetheless, this characteristic turned out to be advantageous in our research for applying GNN Graph Classification methods to time series data, facilitating anomaly detection and providing deeper insights into data patterns [7], [8]. The models' sensitivity to graph details greatly enhances their ability to detect minor differences, highlighting GNNs' potential to improve data analysis by uncovering complex relationships within datasets.

This research aims to revolutionize text analysis by integrating dynamic, context-sensitive systems through the use of GNNs for knowledge graph adaptation and document classification. Our objective is to make a substantial contribution to the NLP field by presenting a new method that not only improves classification accuracy and depth but also the way we construct and understand texts.

This study addresses the challenge of harnessing GNNs for text analysis, merging deep learning with graph techniques to reveal new insights and methods in document classification and knowledge graph exploration. By exploring this intersection, we highlight GNNs' significant potential to enhance Natural Language Processing (NLP), aiming to refine how we classify, construct, and interpret text.

II. Related Work

The emergence of AlexNet models and Knowledge Graphs in 2012 marked a revolution in deep learning and entity relationship understanding, setting a precedent in machine learning across various domains. Google's Knowledge Graphs furthered data integration, improving product intelligence and user experience. This evolution led to the advent of GNNs by the late 2010s, serving as a critical junction between deep learning and knowledge graphs. GNNs excel in analyzing complex data by capturing node relationships within graphs, enhancing prediction and decision-making processes. This has revolutionized how predictions and decisions are made based on graph-structured data [9]–[11].

Real-world data are dynamic and continuously evolving, presenting significant challenges in creating accurate and comprehensive knowledge graphs. The task of automatically constructing complete, dynamic knowledge graphs is particularly daunting. Link prediction emerges as a crucial solution to address these challenges, offering a method to enhance and refine knowledge graphs by predicting missing connections within them [12].

Link prediction is a fundamental problem that attempts to estimate a likelihood of existence of a link between two nodes, which makes it easier to understand associations between two specific nodes and how the entire network evolves [13]. The problem of link prediction over complex networks can be categorized into two classes. One is to reveal the missing links. The other is to predict the links that may exist in the future as the network evolves.

Various types of link predictions has been widely applied to a variety of fields. In social networks link predictions support potential collaborations and help to find assistants. In biology and medicine link predictions provide the ability to foresee hidden associations like protein–protein interactions. [14].

In recent years, link predictions are extensively used in social networks, citation networks, biological networks, recommender systems, security and so on and link prediction models attract more and more studies.

Before GNN became an emerging research area link prediction techniques were based either on graph topology or on node features [15]. There has been a surge of algorithms that make link prediction through representation learning that learns low dimensional embeddings such as DeepWalk [16], node2vec [17], etc. Over the years many link prediction methods have been developed [12].

As the Graph Neural Networks have been an emerging research area in recent years, significant advances and various architectures were proposed and developed [13]. For this study we will use GraphSAGE link prediction model [18], an inductive learning algorithm for GNNs which instead of applying the whole adjacency matrix information among all nodes, learns aggregator functions that can induce the embedding of a new node given its features and neighborhood information without retraining of the entire model [12].

Link prediction involves estimating the likelihood of a link's existence between two nodes and the evolution of the entire network [14]. Link prediction techniques have found applications in various fields. In social networks, link predictions support the identification of potential collaborations and help find suitable connections. In biology and medicine, link predictions enable the anticipation of hidden associations, such as protein–protein interactions [13].

GNN Graph Classification is widely used in chemistry and medicine, where small graphs are used to capture relationships between entities such as molecules, proteins, and biomolecules. It is also commonly applied in biology for analyzing brain networks and genomic data. For example, molecular structures can be represented as graphs with atoms as nodes and chemical bonds as edges, and protein-protein interaction networks can be represented as graphs in medicine [19]. Researchers in medicine and biology are gaining insights into complex biological systems and develop new therapies or treatments [19]–[21].

Our previous research [7], [8] explored the use of GNNs in analyzing time series data, particularly focusing on EEG and climate records. This work demonstrated the GNN models' heightened sensitivity to graph topology, which proved instrumental in identifying anomalies and providing deep insights into complex data patterns. By repurposing GNN Graph Classification for time series analysis—a technique traditionally applied in biology and chemistry—we revealed new potential for GNNs in data interpretation, highlighting their adaptability and the innovative approach of constructing graphs within and across datasets.

As our research focused on the application of GNN Graph Classification models to NLP tasks, we found a lack of previous studies in this specific area. The literature review revealed no existing works exploring the use of GNN Graph Classification models for NLP tasks. This highlights the novelty of our research and the unique contribution we make in this field.

In studies concerning art and artists, specialized networks such as ContextNet have been adopted to extract nuanced artistic contexts from artworks, leveraging both multitask learning techniques and knowledge graphs to understand subtle artistic connections. These innovations have enhanced standard neural networks, improving tasks such as art classification, retrieval, and visualization on knowledge graphs [22]. Concurrently, the development of knowledge graphs, notably ArtGraph, which consolidates information from sources like WikiArt and DBpedia, has notably enriched the retrieval process and prediction models of art features by fusing deep learning models with rich contextual data [23].

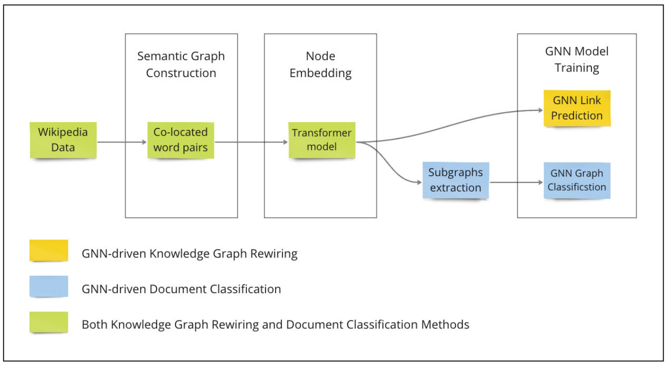
Semantic Graph Construction
Node Embedding
GNN Model Training
Wikipedia Data
Co-located word pairs
Transformer model
GNN Link Prediction
Subgraphs extraction
GNN Graph Classification
🟨 GNN-driven Knowledge Graph Rewiring
🟦 GNN-driven Document Classification
🟩 Both Knowledge Graph Rewiring and Document Classification Methods

Fig. 1. Architecture pipeline of GNN-driven Knowledge Graph Rewiring and GNN-driven Document Classification methods.

III. Methods

Our study leverages Wikipedia articles to delve into the connections among modern art artists, employing an integrated framework that melds GNN link prediction with Graph Classification models. This harmonized approach is visually summarized in Figure 1, which delineates the methodologies adopted in our analysis.

Our method starts by building semantic graphs from artist articles, using a transformer model for text embedding to set the stage for analysis. After embedding, we train the GNN Link Prediction model to predict new connections in the artist network. Meanwhile, for the Graph Classification model, we carefully select labeled subgraphs from the main semantic graph. This preparation focuses the model on specific, useful data, improving its ability to recognize and classify intricate artist relationships and patterns. Figure 1 illustrates the shared and distinct aspects of the methodologies applied, providing a clear visual representation of our analytical architecture.

The methodologies can be summarized as follows:

- For semantic graph construction we use a method based on co-located word pairs.
- To translate text to vectors we use the same node embedding transformer model.
- To get small graphs for GNN Graph Classifcation, we extract subgraphs centered on nodes with high betweenness centrality.
- For the Knowledge Graph Rewiring scenario, we use the DGL GNN Link Prediction model.

- For Document Classification scenario, we use the Pytorch Geometric Graph Classification model.

Detail information is described in posters of our technical blog [24], [25]

A. Build Semantic Knowledge Graph on Co-located Word Pairs

To transform text data to semantic graph with nodes as co-located word pairs we will do the following:

- Tokenize Wikipedia text and exclude stop words.
- Get nodes as co-located word pairs and edges between nodes.
- Get edges between nodes.
- Build semantic graph.

To generate edges we will find pair to pair neighbors following text sequences within articles and joint pairs that have common words.

```
if pair1=[leftWord1, rightWord1],
   pair2=[leftWord2, rightWord2]
   and rightWord1=leftWord2,
then there is edge12={pair1, pair2}
```

Graph edges built based of these rules will cover word to word sequences and word to word chains within articles.

B. Node Embedding

For both Link Prediction and Graph Classification scenarios to translate text to vectors we will use the 'all-MiniLM-L6-v2' transformer model from Hugging Face. This is a sentence-transformers model that maps text to a 384 dimensional dense vector space.

C. Extract Semantic Subgraphs

As input data for GNN Graph Classification model we create a collection of subgraphs derived from documents, focusing on neighbors and neighbors of neighbors around nodes with high betweenness. Betweenness centrality measures a node's importance as a bridge along the shortest paths between pairs of nodes within a graph. This metric highlights nodes that significantly influence the flow of information across the network, ensuring our model's input data captures pivotal structural and contextual elements in the text.

Description of GNN Graph Classification method and code can be checked in our technical blog [25]

D. Training the GNN Link Prediction Model

For this study we will use GraphSAGE link prediction model [18]. This algorithm is based on learning aggregator functions that can induce the embedding of a new node given its features and neighborhood information without retraining of the entire model. The concatenated vector will be passed through a GNN layer to update the node embedding.

As GNN Link Prediction model we used a model from Deep Graph Library (DGL) [26]. The model is built on two GraphSAGE layers and computes node representations by averaging neighbor information. For data preparation and model training we used the code provided by DGL tutorial. In our code we only had to transform input graph data to DGL data format.

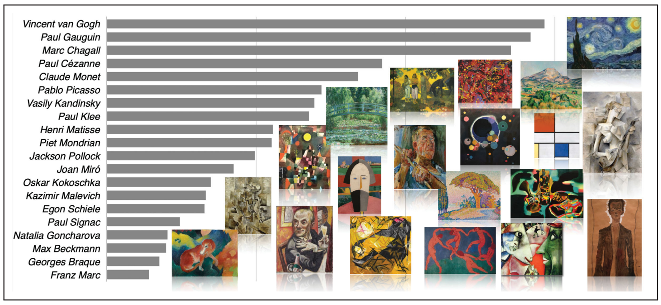

Fig. 2. Modern art artists Numbers of words in Wikipedia articles about modern art artists

The results of GNN Link Prediction model are re-embedded nodes that can be used for further data mining such as node classification, k-means clustering, link prediction and so on. To find unknown connections between modern art artists, we will use a non-traditional approach. We will aggregate re-embedded nodes by artists and estimate link predictions by cosine similarities between aggregated vectors.

E. Train the GNN Graph Classification Model

As GNN Graph Classification model we will use a GCN-Conv (Graph Convolutional Network Convolution) activation model. The model code is taken from tutorial of the PyTorch Geometric Library (PyG) [27]. The GCNConv graph classification model is a type of graph convolutional network that uses convolution operations to aggregate information from neighboring nodes in a graph. It takes as input graph data (edges, node features, and the graph-level labels) and applies graph convolutional operations to extract meaningful features from the graph structure.

The Python code for the GCNConv model is provided by the PyG library. The code for converting data to the PyG data format, model training and interpretation techniques are available in our technical blog [25].

IV. Experiments: Rewiring Knowledge Graphs by GNN Link Prediction

A. Data Source

For experiments conducted in this study, we utilized Wikipedia articles about biographies of modern art artists. These articles focus on a list of 20 modern art artists represented in Table I and in in Fig. 2.

To compare sizes of Wikipedia articles we tokenized text data and calculated counts of words. Based on text size distribution (Table I), the most well known artist in this list is Vincent van Gogh and the most unknown artist is Franz Marc. The size of Wikipedia article about Franz Marc is less than 10 percent of the size of Wikipedia article about Vincent van Gogh.

The list of artists, their art styles and corresponding Wikipedia articles sizes are presented in Fig. 2.

TABLE I. NUMBERS OF WORDS IN WIKIPEDIA ARTICLES ABOUT MODERN ART ARTISTS

Artist	Number of Words
Vincent van Gogh	13677
Paul Gauguin	13249
Marc Chagall	12627
Paul Cézanne	8609
Claude Monet	7852
Pablo Picasso	6713
Vasily Kandinsky	6491
Paul Klee	6314
Henri Matisse	5188
Piet Mondrian	5148
Jackson Pollock	4626
Joan Miró	3959
Oskar Kokoschka	3247
Kazimir Malevich	3097
Egon Schiele	3048
Paul Signac	2290
Natalia Goncharova	1897
Max Beckmann	1850
Georges Braque	1639
Franz Marc	1324

B. Building Initial Knowledge Graph on Co-located Word Pairs

The GNN Link Prediction scenario is based on a knowledge graph that is built on co-located word pairs as nodes and word chains within and across the articles as edges. As we illustrated in Table I, artists have Wikipedia articles of very different sizes and if we use full Wikipedia text data, well-known artists, i.e. artists with longest articles will get more word pairs and much more connections than unknown artists.

To balance artist to artist relationship distribution we selected subsets of articles with similar word pair counts. As all selected Wikipedia articles about artists start with high level artist biography descriptions, from each article we selected the first 800 words.

To generate initial knowledge graph we used the following steps:

- Tokenized Wikipedia text and excluded stop words.
- Selected the first 800 words from Wikipedia articles.
- Generated nodes as co-located word pairs.
- Calculated edges as pair to pair neighbors following text sequences within articles.
- Calculated edges as joint pairs that have common words. These edges will represent word chains within articles and connect different articles through word chains across them.
- Built an initial knowledge graph.

Coding techniques for building initial knowledge graph for this scenario are described in our technical blog [24]

C. Node Embedding

We'll employ the Hugging Face transformer model to translate text into vector embeddings. In our scenario, these transformer-generated vectors serve as rich node features, significantly improving the model's ability to predict links by capturing the nuanced semantic relationships within the text data. This approach leverages the depth of contextual understanding inherent in transformer models, offering a powerful method to enrich graph-based analyses with detailed linguistic insights.

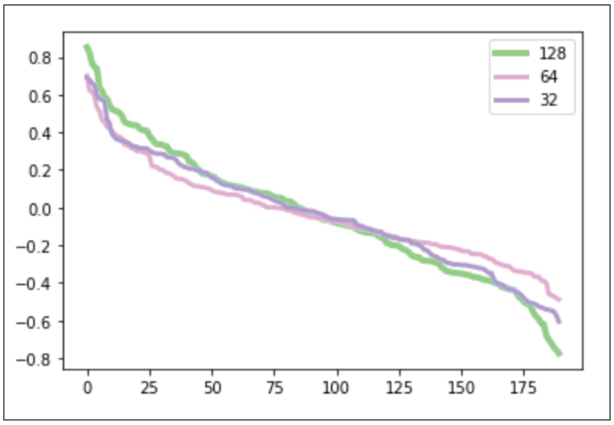

Fig. 3. Cosine similarity distributions for GraphSAGE link prediction model outputs of sizes 128, 64 and 32

D. Training GNN Link Prediction Models

As a GNN Link Prediction model we used a GraphSAGE model from Deep Graph Library (DGL). The model code was provided by DGL tutorial [26] and we only had to transform nodes and edges data from our data format to DGL data format.

We used the model with the following parameters:

- 14933 nodes.
- 231699 edges.
- PyTorch tensor of size [14933, 384] for embedded nodes.

To assess our model's performance, we utilized the Area Under Curve (AUC) as the accuracy metric. Experimenting with the GraphSAGE model, we evaluated output vector sizes of 32, 64, and 128, finding their accuracy metrics to be comparably effective, as detailed in Table II.

TABLE II. AUC ACCURACY METRICS FOR GNN LINK PREDICTION GRAPH-SAGE MODEL

Output Vector Size	AUC
32	96.6 percents
64	96.8 percents
128	96.3 percents

Our goal was to identify the optimal vector size through the analysis of cosine similarity distributions for knowledge graph rewiring. The output size of 128 was chosen based on its ability to yield a more consistent cosine similarity distribution, indicating its effectiveness in our context.

```
model =
GraphSAGE(train_g.ndata['feat']
.shape[1], 128)
```

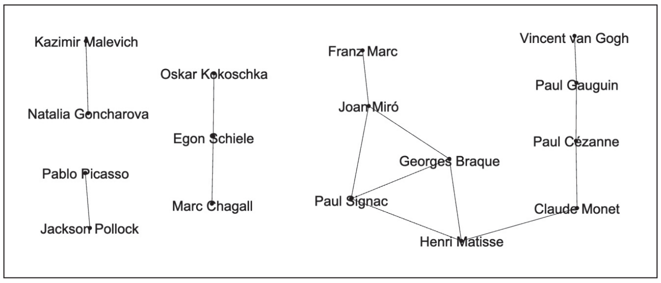

Fig. 4. Rewired Knowledge Graph

E. Rewiring Knowledge Graph

The GNN Link Prediction model doesn't directly produce 'predicted links'. Instead, it generates re-embedded node vectors. These vectors can then be used in subsequent analysis to predict potential edges in the graph. Notably, the model transitions from representing graph relationships in binary terms (connected/not connected) to a continuum, allowing for more nuanced understanding of the connections.

The results of the knowledge graph rewiring scenario are 14933 re-embedded nodes and to detect relationships between artists we calculated average node vectors by artists and estimated link predictions by cosine similarities between them.

As we mentioned above, we experimented with GraphSAGE model output vector sizes of 32, 64 and 128 and compared distributions of cosine similarities between artist pairs. The number of cosine similarity pairs for 20 artists is 190 and the Figure 3 illustrates cosine similarity distributions for model outputs of sizes 128, 64 and 32. For knowledge graph rewiring we selected the model results with output size 128 that reflect a smooth cosine similarity distribution.

In the Table III you can see pairs of artists with highest scores of cosine similarities and in the Table IV - pairs of artists with cosine similarity lowest scores. On Figure 4 you can see graph visualization for pairs of artists with cosine similarity scores more than 0.5.

TABLE III. Artist Pairs with Highest Cosine Similarities

Artist1	Artist2	score
Paul Signac	Henri Matisse	0.8525
Egon Schiele	Marc Chagall	0.8237
Paul Cézanne	Paul Gauguin	0.7679
Kazimir Malevich	Natalia Goncharova	0.7473
Georges Braque	Henri Matisse	0.7392
Georges Braque	Joan Miró	0.6372
Pablo Picasso	Jackson Pollock	0.6224
Georges Braque	Paul Signac	0.5851
Paul Signac	Joan Miró	0.5788
Vincent van Gogh	Paul Gauguin	0.5459
Henri Matisse	Claude Monet	0.5225
Paul Cézanne	Claude Monet	0.5160
Egon Schiele	Oskar Kokoschka	0.5118
Franz Marc	Joan Miró	0.5030

TABLE IV. Artist Pairs with Lowest Cosine Similarities

Artist1	Artist2	score
Egon Schiele	Henri Matisse	-0.7749
Marc Chagall	Henri Matisse	-0.7578
Georges Braque	Egon Schiele	-0.7433
Kazimir Malevich	Claude Monet	-0.7217
Egon Schiele	Paul Signac	-0.7033
Marc Chagall	Paul Signac	-0.6804
Georges Braque	Marc Chagall	-0.6205
Paul Cézanne	Vasily Kandinsky	-0.6167
Paul Klee	Joan Miró	-0.5926
Natalia Goncharova	Claude Monet	-0.5804
Vasily Kandinsky	Claude Monet	-0.5622

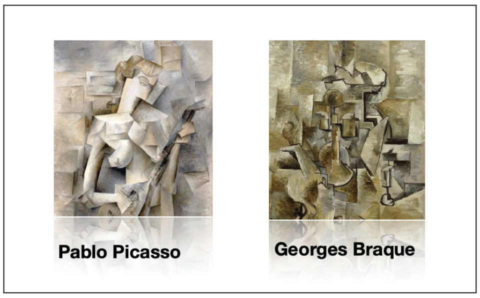

Fig. 5. Artist pairs with high semantic relationships: Pablo Picasso and George Braque were pioneers of cubism art movement.

F. Interpretation of GNN Link Prediction Model Results

The power of relationships between artists observed in our old study [6] aligns with the findings of our study [5], where we investigated semantic similarities and dissimilarities among artist biographies by GNN Link Prediction models applied to Wikipedia articles.

In this study, we explore the significance of node pairs with high cosine similarities, or high weight edges, in graph mining for tasks like node classification and community detection, focusing on their role in analyzing semantic relationships between artists through Wikipedia articles. For instance, Pablo Picasso and Georges Braque, known for pioneering Cubism, exhibit high semantic similarity, as do Paul Gauguin and Vincent van Gogh despite their different art styles. Interestingly, unexpected connections, such as between Egon Schiele and Marc Chagall, present new avenues for modern art research.

We leverage these insights for knowledge graph rewiring,

Fig. 6. Highly disconnected artists taken from not overlapping modern art movements: Futurism - Natalia Goncharova, Impressionism - Claude Monet and De Stijl - Piet Mondrian.

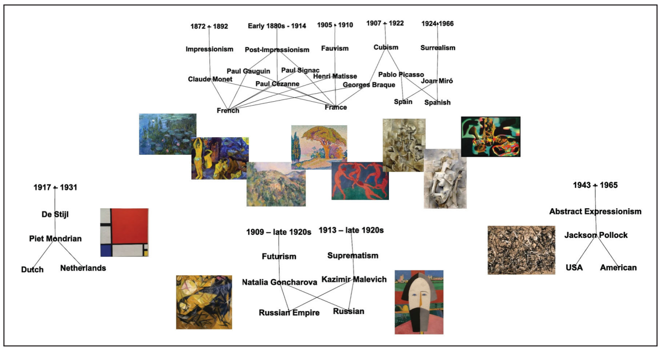

Fig. 7. Modern art artists relationships by their biographies and modern art movements [6]

enhancing recommender systems by suggesting artists or artworks based on semantic similarities. For example, fans of Picasso might be interested in Braque's work, illustrating the practical applications of identifying such high-weight links.

Conversely, node pairs with low cosine similarities, or negative weight edges, traditionally underused, offer a novel perspective. They can effectively indicate that nodes belong to different communities, aiding in community detection validation. Graphs incorporating these dissimilar pairs cover broader spaces, showcasing the potential to diversify recommendations in art, such as suggesting works from different modern art movements to someone familiar with Claude Monet, thereby enriching user experience and exploration in digital art platforms.

V. Experiments: GNN Graph Classification

A. Data Source

As a data source for GNN Graph Classification section, we used the same data as in Rewiring Knowledge Graph section, data from Wikipedia articles on 20 modern art artists represented in Table I and illustrated in Figure 2.

In our GNN Graph Classification model, we deliberately select pairs of artists, both similar and dissimilar, to examine how closely the model's predictions align with our expectations of similarity and dissimilarity. For example, Pablo Picasso and Georges Braque, both pivotal in the Cubism movement, serve as our "similar" pair, anticipating that the model will yield analogous results for them due to their shared artistic lineage and influence. On the other hand, we contrast this by considering a "dissimilar" pair, such as Claude Monet and Kazimir Malevich, who are distinct not only in their artistic styles but also in their historical art movements. This selection aims to test the model's effectiveness in distinguishing between the nuanced connections of artist pairs with close semantic relationships and those with minimal similarities, thereby assessing its capability in reflecting true artistic affiliations and divergences as highlighted (see Figure 7) from study [6].

For a more detailed exploration of the relationships between modern art artists discovered through knowledge graph techniques, you can refer to our technical blog [25]. In that blog post, we provide comprehensive insights into the methods, analysis, and findings of our research.

B. Input Data Preparation for GNN Graph Classification

For the GNN Graph Classification phase, to transform text into semantic graphs we used the co-located word pairs methodology, the same as we used in the Knowledge Graph Rewiring phase.

GNN Graph Classification models require input data as sets of smaller, labeled graphs with defined node features and edge relationships. To prepare the input data for the GNN Graph Classification model, we generated labeled semantic subgraphs from each document of interest. These subgraphs were constructed by selecting neighbors and neighbors of neighbors around specific "central" nodes. The "central" nodes were determined by identifying the top 500 nodes with the highest betweenness centrality within each document.

By focusing on these central nodes and their neighboring nodes, we aimed to capture the relevant information and relationships within the document. This approach allowed us to create labeled subgraphs that served as the input data for the GNN Graph Classification model, enabling us to classify and analyze the documents effectively.

For text to vector translation we used 'all- MiniLM-L6-v2' transformer model from Hugging Face, the same method as it was used in GNN-drive Knowledge Graph Rewiring.

C. Training the GNN Graph Classification Model

In this study, we utilized the GCNConv graph classification model from the PyG library. The Python code for the GCNConv model is available through the PyG library [27], while the code for preparing data and encoding it into the PyTorch Geometric data format can be found on our technical blog [25].

Reflecting on our experience, the models achieved very high accuracy metrics in just a few epochs. Accuracy metrics for Monet and Malevich classifications are presented in Table V and for Picasso and Braque classification in Table VI.

TABLE V. Accuracy Metrics for Classification of Wikipedia Articles about Claude Monet and Kazimir Malevich

Epoch	Train Accuracy	Test Accuracy
1	1.0000	1.0000
2	1.0000	0.9923
3	1.0000	1.0000
4	1.0000	1.0000
5	1.0000	1.0000
6	1.0000	1.0000
7	1.0000	1.0000
8	1.0000	1.0000
9	1.0000	1.0000

This efficiency echoes our earlier findings in GNN graph classification for time series analysis [7], [8]: the GCNConv model's heightened sensitivity distinctly outperformed the GNN Link Prediction model, deftly discerning nuanced variations in artist narratives.

TABLE VI. ACCURACY METRICS FOR CLASSIFICATION OF WIKIPEDIA ARTICLES ABOUT PABLO PICASSO AND GEORGES BRAQUE

Epoch	Train Accuracy	Test Accuracy
1	0.6655	0.7308
2	0.9561	0.9615
3	0.9655	0.9538
4	0.9701	0.9538
5	0.9862	0.9615
6	0.9931	0.9769
7	0.9977	0.9769
8	0.9943	0.9615
9	0.9989	0.9769

Given the distinct differences between Monet and Malevich as artists, we anticipated achieving high accuracy metrics but in the classification of Wikipedia articles about Pablo Picasso and Georges Braque, we were not anticipating the significant differentiation between these two documents: these artists had very strong relationships in biography and art movements. Also GNN Link Prediction models classified these artists as highly similar.

This observation highlights the high sensitivity of the GNN Graph Classification model and emphasizes the ability of the GNN Graph Classification model to capture nuanced differences and provide a more refined classification approach compared to the GNN link prediction models.

D. Interpretation of GNN Graph Classification Model Results

Our GNN Graph Classification model quickly reached 100% accuracy in just a few epochs. At first, this made us question the results, but a deeper investigation revealed that this outstanding performance stemmed from the model's acute sensitivity to variations in graph topology. This heightened sensitivity, while offering precision in identifying minor data distinctions, also flagged potential challenges, such as the model's vulnerability to random noise. This discovery highlighted the importance of careful noise management in subsequent investigations.

The model's results were revealing. Although artists like Picasso and Braque worked closely together, the model found big differences in their Wikipedia articles. In contrast, the GNN Link Prediction models viewed them as similar. For artists like Monet and Malevich, who were from different backgrounds, the model correctly found them dissimilar.

One challenge with this model is spotting outliers. As expected, in the scenario of classifying Wikipedia articles about the biographies of Claude Monet and Kazimir Malevich, the trained model did not detect any outliers.

In the case of Pablo Picasso and Georges Braque, their Wikipedia articles exhibited just a few similarities. Despite their shared biographies and involvement in the same art movements, the model saw their articles as very different. Out of 1000 subgraphs, we found only 8 outliers, shown in Table VII.

In summary, our GNN Graph Classification model effectively classifies text by identifying complex patterns, thanks to its high sensitivity. Yet, this sensitivity requires careful use to avoid noise and ensure accurate, dependable analysis.

TABLE VII. OUTLIERS IN GNN GRAPH CLASSIFICATION MODEL FOR CLASSIFICATION OF WIKIPEDIA ARTICLES ABOUT PABLO PICASSO AND GEORGES BRAQUE

Probability	Central Node
0.5088	always things
0.5437	develop explore
0.8668	question standard
0.9516	etchings aquatints
0.9999	producing guitar
1.0000	reducing everything
1.0000	regarded statesmen
1.0000	models virtuality

VI. CONCLUSION

In our study, we found that using Graph Neural Networks (GNNs) greatly improves how we represent knowledge and analyze text. By combining GNNs for changing knowledge graphs and classifying documents, we've been able to make our comparisons of texts more detailed. This approach has helped us better understand the relationships within texts and get deeper insights from GNN models. Our results show that these methods work well together, making it easier to understand complex text data and knowledge graphs.

In our prior research, we introduced a novel approach using GNNs for Knowledge Graph Rewiring to uncover hidden relationships within graphs, specifically applied to the analysis of modern art artists' biographies through Wikipedia. This method allowed us to detect both closely and loosely connected node pairs, highlighting the breadth of knowledge graphs for community detection. In the current study, we've taken this methodology further by comparing it with the more detailed process of GNN Graph Classification, focusing on achieving a deeper understanding of semantic connections. While we've previously demonstrated how these rewired knowledge graphs could enhance recommender systems, our primary aim here is to delve deeper into the method's potential, pushing the boundaries of how we analyze and interpret complex data structures.

In our GNN Graph Classification research, we ventured beyond the conventional domains of chemistry and biology, applying GNN Graph Classification models to the realm of text analysis. This approach, centered on constructing semantic graphs from text documents, has opened new frontiers for data exploration. While the model's sensitivity to graph topology is a double-edged sword—indicating potential overfitting yet also displaying an acute awareness of network structures—it highlights the critical need for refined noise management and regularization strategies in future developments.

The nuanced analysis of artist biographies underscores the model's capability to discern complex relational patterns, offering a rich ground for future exploration into feature importance and model interpretability. However, the challenge of outlier detection within extensive datasets remains a significant limitation, signaling the necessity for advanced methodologies or model enhancements to address this issue.

In conclusion, our exploration of GNN for Knowledge Graph Rewiring and Document Classification has expanded the scope of GNN applications, demonstrating their effectiveness in uncovering hidden meanings in text and improving recommendation systems. As we further develop these models, it will be crucial to enhance their resistance to noise and their ability to detect outliers. This research not only deepens our comprehension of GNN's capabilities but also lays the groundwork for future breakthroughs in text analysis and knowledge graph rewiring.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[3] N. Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, and J. Taylor, "Industry-scale knowledge graphs: Lessons and challenges," *ACM Queue*, 2019, https://queue.acm.org/detail.cfm?id=3332266.

[4] M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, "Geometric deep learning: Grids, groups, graphs, geodesics, and gauges," *arXiv*, 2021.

[5] A. Romanova, "Rewiring knowledge graphs by graph neural network link predictions," *International Conference on Agents and Artificial Intelligence (ICAART)*, 2023.

[6] ——, "Building knowledge graph in spark without sparql," in *Database and Expert Systems Applications. DEXA 2020, MLKgraphs2020. Communications in Computer and Information Science*. Springer, 2020.

[7] ——, "Gnn graph classification method to discover climate change patterns," 2023.

[8] ——, "Enhancing time series analysis with gnn graph classification models," in *Studies in Computational Intelligence*, ser. SCI, vol. 1141, 2024.

[9] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *arXiv*, 2019.

[10] H. Wu, C. Song, Y. Ge, and T. Ge, "Link prediction on complex networks: An experimental survey," *Data Science and Engineering*, 2022.

[11] M. Wang, L. Qiu, and X. Wang, "A survey on knowledge graph embeddings for link prediction," *Symmetry*, 2021.

[12] X. Wang and A. Vinel, "Benchmarking graph neural networks on link prediction," *arXiv*, 2021, https://arxiv.org/pdf/2102.12557.pdf.

[13] H. Wu, C. Song, Y. Ge, and T. Ge, "Link prediction on complex networks: An experimental survey," *Data Science and Engineering*, 2022.

[14] T. Zhou, "Progresses and challenges in link prediction," *iScience*, 2021.

[15] T. Zhou, L. Lu, and Y.-C. Zhang, "Predicting missing links via local information," 2009, https://doi.org/10.1140/EPJB/E2009-00335-8.

[16] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," 2014, https://arxiv.org/abs/1403.6652.

[17] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," 2016, https://arxiv.org/abs/1607.00653.

[18] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," 2017, https://arxiv.org/abs/1706.02216.

[19] J. Adamczyk, "Application of graph neural networks and graph descriptors for graph classification," 2022.

[20] W. Hu1, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec, "Strategies for pre-training graph neural networks," 2020.

[21] H. He, O. Queen, T. Koker, C. Cuevas, T. Tsiligkaridis, and M. Zitnik, "Domain adaptation for time series under feature and label shifts," 2023.

[22] N. Garcia, B. Renoust, and Y. Nakashima, "Contextnet: Representation and exploration for painting classification and retrieval in context," 2020.

[23] G. Castellano, G. Sansaro, and G. Vessio, "Integrating contextual knowledge to visual features for fine art classification," 2021.

[24] sparklingdataocean.com. (2022) Find semantic similarities by gnn link predictions. [Online]. Available: http://sparklingdataocean.com/2022/11/09/knowledgeGraph4NlpGnn/

[25] ——. (2023) Exploring document comparison with gnn graph classification. [Online]. Available: http://sparklingdataocean.com/2023/07/07/knowledgeGraph4NlpGnn2/

[26] Deep Graph Library (DGL), "Link prediction using graph neural networks," 2018, available online: https://www.dgl.ai.

[27] PyTorch Geometric Library (PyG), "Graph classification with graph neural networks," 2023, available online: https://pytorch-geometric.readthedocs.io.