

Disentangled Representation Learning with Large Language Models for Text-Attributed Graphs

Yijian Qin¹ Xin Wang¹ Ziwei Zhang¹ Wenwu Zhu¹

Abstract

Text-attributed graphs (TAGs) are prevalent on the web and research over TAGs such as citation networks, e-commerce networks and social networks has attracted considerable attention in the web community. Recently, large language models (LLMs) have demonstrated exceptional capabilities across a wide range of tasks. However, the existing works focus on harnessing the potential of LLMs solely rely on prompts to convey graph structure information to LLMs, thus suffering from insufficient understanding of the complex structural relationships within TAGs. To address this problem, in this paper we present the Disentangled Graph-Text Learner (DGTL) model, which is able to enhance the reasoning and predicting capabilities of LLMs for TAGs. Our proposed DGTL model incorporates graph structure information through tailored disentangled graph neural network layers, enabling LLMs to capture the intricate relationships hidden in text-attributed graphs from multiple structural factors. Furthermore, DGTL operates with frozen pre-trained LLMs, reducing computational costs and allowing much more flexibility in combining with different LLM models. Experimental evaluations demonstrate the effectiveness of the proposed DGTL model on achieving superior or comparable performance over state-of-the-art baselines. Additionally, we also demonstrate that our DGTL model can offer natural language explanations for predictions, thereby significantly enhancing model interpretability.

1. Introduction

Text-attributed graphs (TAGs) are employed to represent a group of structured data where textual entities are connected by graph relations. TAGs are ubiquitous on the web, such as citation networks, e-commerce networks, social media, recommendation systems, web pages etc. As such, representation learning on TAGs has become an important research problem recently in the web community, where the TAGs are explored to capture rich semantic relationships and dependencies among connected textual elements, providing valuable contexts for better understanding and reasoning in various downstream tasks. Classical text-attributed graph representation approaches normally utilize graph neural networks (GNNs) to capture structural information, transforming textual attributes into shallow or hand-crafted representations such as bag-of-words or skip-gram features, which will then be used for prediction tasks in TAG (Kipf & Welling, 2017). Some works also use natural language processing models to enhance GNN classifiers by augmenting the node features and capture the rich semantic information (Zhang et al., 2020; Yang & Cui, 2021).

Recent advancements in machine learning and artificial intelligence have witnessed the emergence of large language models (LLMs) that exhibit unprecedented capabilities in various tasks (Zhao et al., 2023). These models have demonstrated remarkable proficiency in natural language processing related tasks including language generation, machine translation and sentiment analysis, as well as other fields such as recommendation system (Wu et al., 2023), social network analysis (Gao et al., 2023), code analysis (Chen et al., 2021), bioinformatics (Thirunavukarasu et al., 2023) and many more.

Therefore, the advent of LLMs has promoted the direct exploration of LLMs to solve prediction problems in TAG without GNN classifiers. These existing approaches solely rely on prompts to convey graph structure information to LLMs, suffering from insufficient understanding of the complex structural relationships within TAGs. For example, Graph-LLM (Chen et al., 2021) uses neighbor summary to generate prompts with structural information, while Instruct-GLM (Ye et al., 2023) directly describes all the neighbors in the prompt. Nevertheless, only utilizing prompts to pass

¹Department of Computer Science and Technology, Tsinghua University. Correspondence to: Wenwu Zhu <wwzhu@tsinghua.edu.cn>, Xin Wang <xin_wang@tsinghua.edu.cn>.

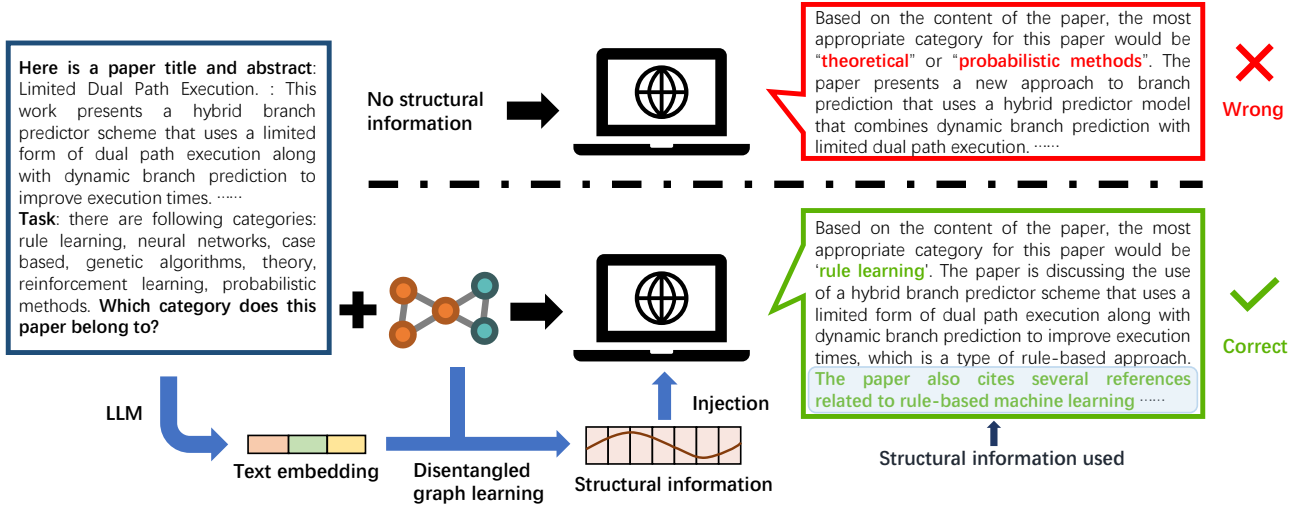


Figure 1. An illustration of using LLMs to solve paper classification task on TAGs. Top: existing methods fail to give correct answer because of lacking structural information on the TAG. Bottom: our method can predict the correct label by utilizing the structural information, i.e., the reference information in the citation graph.

information about graph structure over LLMs hinders the model’s ability to capture and utilize the intricate relationships and dependencies encoded in the graph structures of TAGs, resulting in limited capability of exploiting the potential power of LLMs in TAG tasks.

To tackle this problem, in this paper we go beyond the vanilla prompt-based methods and effectively integrate graph structure information into LLMs, in order to enable the holistic utilization of LLMs’ exceptional powers in TAG tasks. However, achieving this goal poses the following technical challenges. First, TAGs usually contain rich yet entangled structural information, i.e., not all the information is relevant or helpful for downstream tasks. LLMs need to effectively filter out and extract the pertinent information while disregarding the irrelevant details carried in the graph structure. Second, adapting LLMs to a specific TAG task is challenging, given that LLMs typically require extensive training and learning of task-specific knowledge. The process of fine-tuning LLMs for particular tasks involves striking the balance between maintaining the pre-trained knowledge and acquiring new knowledge specific to the target tasks.

To solve these challenges, we propose the Disentangled Graph Text Learner (DGTL) model in this paper, which can boost LLMs in deeply exploiting structural information to enhance their reasoning and predicting capabilities for TAG tasks. The proposed DGTL model first encodes raw text information carried in TAGs using a pre-trained LLM. Then, a group of tailored disentangled GNN layers is developed to capture graph neighborhood information in TAGs from multiple structural factors. By injecting the learned features with disentangled factors into the LLM pre-

dictor, we enable the model to comprehend complex graph structure information in TAGs. Moreover, DGTL allows the pre-trained LLMs to remain frozen, thereby reducing computation costs and mitigating the risk of catastrophic forgetting in LLMs. This flexibility enables DGTL to be compatible with different LLM models, ensuring its practical applicability. Overall, DGTL is able to serve as a general framework for combining text-attributed graph learning and natural language modeling to improve the explainability and predictive performance of LLMs for TAG tasks.

To demonstrate the effectiveness of our proposed method, we compare it with state-of-the-art approaches on various TAG benchmarks. Our method achieves superior or comparable results with baseline methods. Additionally, we demonstrate that DGTL can offer human-understandable explanations for the model predictions in natural language.

In summary, we make the following contributions:

- We propose Disentangled Graph Text Learner (DGTL), a novel model which deeply exploits graph structure information to enhance the reasoning and predicting capabilities of LLMs for TAG tasks. DGTL also serves as a general framework for integrating structural analysis abilities of GNNs with the powerful language modeling capabilities of LLMs.
- We propose tailored disentangled GNN layers to capture graph neighborhood information from multiple structural factors, enabling the LLMs to comprehend complex graph structure information in TAGs.
- Our proposed DGTL enables pre-trained LLMs to remain frozen, benefiting from reduced computation costs as well

as mitigating the risk of catastrophic forgetting.

- We conduct extensive experiments on various TAG benchmarks and compare DGTL with state-of-the-art baselines. The results demonstrate that DGTL is able to achieve superior or comparable performance, as well as provide users with human understandable natural language explanations for the model’s predictions.

2. Related Works

2.1. Text-attributed Graphs

Text-attributed graphs (TAGs) have gained significant attention in the field of graph machine learning in recent years (Zhao et al., 2022; Huang et al., 2023; Duan et al., 2023). A TAG is a type of graphs where each node is associated with a text attribute. This representation captures the rich semantic relationships and dependencies among textual elements, making TAGs valuable for understanding and reasoning tasks. Commonly used TAG benchmark datasets include Cora, CiteSeer, PubMed (Sen et al., 2008), and OGBN-arXiv (Hu et al., 2020), where nodes represent papers and edges represent reference relationships.

Message-passing GNNs (Kipf & Welling, 2017; Veličković et al., 2018; Xu et al., 2019; Hamilton et al., 2017) have been proposed as an effective framework for graph machine learning following the neighborhood aggregation scheme. At each layer, nodes learn representations by aggregating their neighbors’ representations (Gilmer et al., 2017). GNNs have also made significant progress in TAG tasks by considering both node attributes and graph structures. Classical GNN pipelines typically handle text attributes by converting them into shallow or hand-crafted features such as bag-of-words (Zhang et al., 2010) or skip-gram (Mikolov et al., 2013) representations. However, with the advancements in natural language processing, there has been a shift towards utilizing language models to generate more comprehensive node features based on the text attribute. This approach allows for a deeper understanding and representation of learning on TAGs.

Despite the progress made by LLMs and GNNs in capturing textual or structural information for representation learning on TAGs, there are still large rooms for improvement. The integration of these models can lead to enhanced performance and more effective utilization of the rich information contained within TAGs.

2.2. LLMs for Graph Tasks

Recent researches have also delved into the exploration of leveraging LLMs directly for addressing graph-related tasks (Zhang et al., 2023). The fundamental concept behind this approach is to convert graph data, including both

structural components and features, as well as graph tasks, into natural language representations. By treating graph problems as conventional NLP problems, researchers have unlocked the potential of utilizing LLMs for graph-related tasks. In the subsequent sections, we present a comprehensive overview of these recent advancements in the field.

Pioneer researches begin with explorations on synthetic graph tasks. NLGraph (Wang et al., 2023) reorganizes graphs as natural language description and conducts a systematic evaluation of LLMs on eight graph reasoning tasks in natural language, including connectivity, shortest path, maximum flow, simulating GNNs, etc. Meanwhile, GPT4Graph (Guo et al., 2023) also conducts extensive experiments by converting graphs into specific code formats. They evaluate the graph understanding capabilities of LLMs across ten distinct tasks, including structure understanding tasks and semantic understanding tasks. LLMtoGraph (Liu & Wu, 2023) also tests GPT-3.5 and GPT-4 for various graph tasks by natural language description and makes some interesting observations.

More recently, several works carry out explorations on TAGs with LLMs. Graph-LLM (Chen et al., 2023) systematically investigates two strategies on TAGs: LLMs-as-Enhancers and LLMs-as-Predictors. The former strategy uses LLM to enhance the representations of text attributes of nodes before passing them to GNNs, while the latter one directly employs LLM as TAG task predictors with natural language prompts. They also explore using ego-graph description and neighbor summary to incorporate structural information by prompt. InstructGLM (Ye et al., 2023) expands the vocabulary by creating new tokens for every node in the TAG, which enables tuning LLMs to handle various TAG tasks in a generative manner. Nevertheless, the existing methods use prompt to convey neighborhood information for downstream LLMs, which faces several challenges, such as the issue of excessive neighboring information leading to lengthy prompt texts.

In our method, we employ a disentangled graph learning approach to compress the neighboring information on the TAG into a small number of tokens. This enables LLMs to learn and utilize the rich knowledge contained within these compressed tokens for downstream inference tasks. Besides, our method follows a delta-tuning scheme where the LLM is kept frozen and only a small number of parameters are tuned (Ding et al., 2023), making our method easy to integrate with off-the-shelf LLMs.

3. Problem Formulation and Preliminaries

3.1. Text-attributed Graphs

A text-attributed graph can be formulated as $\mathcal{G} = (\mathcal{V}, \mathbf{A}, \mathbf{y})$, where \mathcal{V} denotes the set of nodes, \mathbf{A} denotes the adjacent

matrix, and \mathbf{y} denotes the labels of nodes. In TAGs, each node is associated with a text description, e.g., the abstract of papers for citation graphs. Before the message-passing procedure in the GNNs, we need to process the raw texts into real-valued features, i.e., text embeddings.

In this paper, we focus on node classification, one of the most typical tasks on TAGs. We adopt the semi-supervised settings, where the text information of all the node set \mathcal{V} and \mathbf{A} is given at the training procedure, as well as a part of the node labels $\{\mathbf{y}_u | u \in \mathcal{V}_{tr}\}$, where \mathcal{V}_{tr} is the training node set. The task aims at predicting the labels $\{\mathbf{y}_u | u \in \mathcal{V}_{te}\}$ of the testing node set \mathcal{V}_{te} .

3.2. Graph Neural Network

GNNs are state-of-the-art models for graph machine learning, which typically follow a message passing scheme where nodes aggregate information from their neighbors in each layer formulated as:

$$\mathbf{m}_i^{(l)} = \text{Agg}(\mathbf{h}_j^{(l)} | j \in \mathcal{N}_i), \quad (1)$$

$$\mathbf{h}_i^{(l+1)} = \text{Update}(\mathbf{m}_i^{(l)}), \quad (2)$$

where $\mathbf{h}_i^{(l)}$ is the representation of node i at the l -th layer, \mathcal{N}_i denotes the neighbors of node i derived from the adjacent matrix, $\text{Agg}(\cdot)$ is the aggregation function, $\text{Update}(\cdot)$ is an updating function between two node representations.

3.3. Large Language Models

Large language models (LLMs) have revolutionized natural language processing tasks by demonstrating remarkable capabilities in understanding and generating human-like text. One of the key architectural advancements that underpins the success of LLMs is the transformer model, which has become the de facto standard for modeling sequential data and computer vision.

The transformer architecture (Vaswani et al., 2017) is based on the principle of self-attention. The attention mechanism enables the model to weigh the importance of different parts of the input sequence when making predictions. The attention mechanism calculates attention scores between each pair of positions in the input sequence, allowing the model to focus on relevant information while disregarding irrelevant or redundant elements. This attention mechanism is crucial for capturing long-range dependencies and contextual relationships in text. The self-attention in current decoder-based LLMs can be formulated as follows:

$$\text{Attention}(\mathbf{H}) = \text{softmax}\left(\frac{f_q(\mathbf{H})f_k(\mathbf{H})}{\sqrt{d}}\right)f_v(\mathbf{H}). \quad (3)$$

Here, \mathbf{H} is the hidden features, $f_q(\cdot)$, $f_k(\cdot)$, and $f_v(\cdot)$ are the learnable projection functions to calculate the query, key, and values.

In a transformer, the attention mechanism operates through multiple self-attention layers. Each layer consists of multiple attention heads, which learn different representations and capture diverse aspects of the input sequence. By employing self-attention across multiple layers, transformers can effectively model both local and global dependencies in the input sequence.

Furthermore, LLMs are typically pre-trained on massive amounts of text data using unsupervised learning objectives. The most commonly used task is language modeling, which is formulated as:

$$\mathcal{L} = -\log p(s_i | s_{1:i-1}), \quad (4)$$

where $s_{1:i}$ is a natural language sequence. This pre-training phase enables the models to acquire a rich understanding of language patterns and structures, and commonsense knowledge. The pre-trained LLMs can then be fine-tuned on specific downstream tasks with task-specific supervised learning, allowing them to adapt their knowledge to specific tasks and domains.

4. Method

4.1. Overall Framework

We aim to leverage LLMs to provide predictions for TAG tasks while enabling interpretability. The main challenge lies in efficiently equipping the LLM with neighborhood knowledge of the target nodes in the TAG. To address this, our framework incorporates disentangled graph learning, which allows us to learn rich semantic information from the neighborhood and inject it into the downstream LLM. An overall framework of our method is shown in Figure 2. Next, we elaborate our proposed method.

- In **Step 1**, we generate text embeddings by computing the average of the features at the last layer in the upstream LLM. This process captures the overall context and semantics of the text associated with each node in the TAG, which form the foundation for subsequent steps.
- In **Step 2**, we employ disentangled graph learning to learn embeddings that incorporate diverse neighborhood information. The disentangled graph learning approach allows us to capture varied information from the neighborhood, facilitating a comprehensive understanding of the TAG.
- In **Step 3**, we inject the learned features with neighborhood information into the downstream LLM. This integration of the LLM and the disentangled GNN facilitates more accurate and informed predictions for the downstream TAG tasks.

The combination of disentangled graph learning and information injection in the downstream LLM forms the core

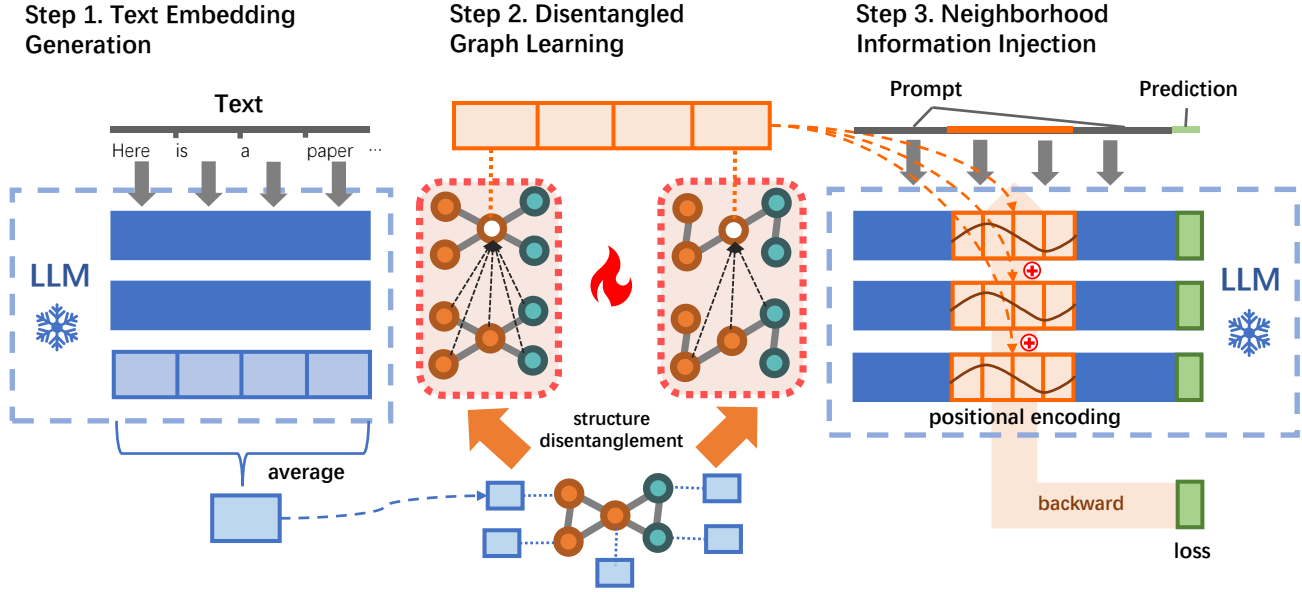


Figure 2. An overview of our proposed DGTL method. **Step 1:** Generating text embedding by taking average of the features at the last layer in the upstream LLM. **Step 2:** Using our proposed disentangled graph learning to learn embeddings with diverse structural information. **Step 3:** Injecting the features with neighborhood information into the downstream LLM.

of our framework. This approach allows us to harness the strengths of both graph representation learning and language modeling, enabling effective utilization of structural information and enhancing the interpretability of our predictions.

Moreover, we adopt a scheme where the pre-trained LLMs remain frozen in our approach. We only focus on tuning the disentangled GNN to generate structural information for LLMs. This scheme benefits us from several advantages. Firstly, we keep a very low computation cost on fine-tuning since the disentangled GNN only cover a small rate of the parameters. Even if the LLM is updated to a new version, we can quickly adapt to the new one with minimal tuning procedure. Secondly, our approach maximizes the utilization of LLMs’ pre-existing knowledge and mitigates the risk of catastrophic forgetting.

4.2. Text Embedding Generation

The first step in our method is to generate an effective text embedding that encapsulates the semantics and contextual information of the input text. Traditionally, previous works(Chen et al., 2023) have primarily focused on utilizing the embedding of the End-Of-Sentence (EOS) token at the last layer as the text embedding. However, we propose that taking the average of the hidden states in the last layer provides a more comprehensive representation of the entire input text.

Considering the average of the hidden states enables us to capture the collective information and contextual under-

Algorithm 1 The training process of DGTL

Require: Text-attributed graph $\mathcal{G} = (\mathbf{T}, \mathbf{A})$, LLM L_{up}, L_{down}
Ensure: Parameters of the disentangled GNN θ

- 1: Get node embedding $\mathbf{H}^{(0)}$ of \mathbf{T} by L_{up}
- 2: Initialize θ randomly
- 3: **while** not converge **do**
- 4: Sample a mini-batch from the training set
- 5: Compute $\mathbf{H}_i^{(1)} = \text{GNN}_{i,1}(\mathbf{H}^{(0)}, \mathbf{A})$
- 6: Compute disentangled graph structures \mathbf{A}_i by Eq. (5)
- 7: Compute $\mathbf{H}_i^{(2)} = \text{GNN}_{i,2}(\mathbf{H}_i^{(1)}, \mathbf{A}_i)$
- 8: **for** node u in the mini-batch **do**
- 9: Construct prompt and response template of u
- 10: Inject $\mathbf{h}_u^{(2)}$ into L_{down} by Eq. (6)(7)(8) during generation
- 11: Generate prediction and calculate the loss function
- 12: **end for**
- 13: Update θ by gradient descent
- 14: **end while**

standing of the text across all positions in the input sequence. This approach allows us to incorporate a broader range of linguistic cues and dependencies into the text embedding, resulting in a more robust and informative representation.

4.3. Disentangled Graph Learning

Our next step is learning hidden features that capture relevant neighborhood information for the target task. By in-

Table 1. Dataset Statistics.

Dataset	Nodes	Edges	Classes	Domain
Cora	2,708	5,429	7	Academic
PubMed	19,717	44,338	3	Academic
Books-History	41,551	358,574	12	E-commerce

jecting these features into the downstream LLM, we enable the LLM to effectively utilize the complex neighborhood information of the nodes in the TAG to assist in predicting the downstream task.

To achieve this, we employ disentangled graph learning, which allows us to capture and represent the diverse neighborhood information present in the TAG. Specifically, we adopt multiple parallel 2-layer GNNs to learn the features, and disentangle them by assigning diverse graph structures. We use a parameter-efficient and differentiable manner to generate the weights of edges for the graph structures, which can be formulated as follows:

$$\mathbf{A}_{i,(u,v)} = \delta \mathbf{A}_{(u,v)} + (1 - \delta) \mathbf{A}_{(u,v)} \cdot \sigma((\mathbf{S}_i^u \mathbf{h}_u)^\top (\mathbf{S}_i^v \mathbf{h}_v)), \quad (5)$$

where $\mathbf{A}_{i,(u,v)}$ is the weight of edge between node u and v in the adjacent matrix of the i -th GNN, $\mathbf{A}_{(u,v)}$ indicates if there is an edge between node u and v in the original graph, \mathbf{S}_i^u and \mathbf{S}_i^v are learnable parameters to generate edge weights of the i -th GNN, δ is the hyper-parameter, and $\sigma(\cdot)$ denotes the sigmoid function. Consequently, we generate diverse graph structure by assign different edge weights in a continuous space. We can also conveniently use gradient based methods to optimize the parameters.

Our disentangled GNN architecture incorporates diverse graph structures to ensure the learning of varied information from the neighborhood. These graph structures highlight different aspects of the TAG’s topology and semantics, enabling the following LLM to leverage specific types of neighborhood information during downstream task predicting.

4.4. Neighborhood Information Injection

To take full advantages of the LLMs’ ability to understand and model complex patterns and semantics in TAGs, we inject the learned disentangled embedding directly into the downstream LLM.

Specifically, we reserved a set of token positions for placing these disentangled embeddings in the prompt input to the downstream LLM. Even if these embeddings are not in the form of natural language, the form of our input will still allow the LLM to think that they are aligned with the natural semantic space that humans can understand. In this way, we can optimize our disentangled GNNs by gradient descent

methods during fine-tuning.

However, the gradient back-propagation through the entire LLM neural architecture can make the optimization process extremely difficult during the fine-tuning procedure. Therefore, we take a step further by performing the embedding injection in all layers of the downstream LLM. Specifically, we add the disentangled embedding to the text embedding at the reserved positions in key and query projection functions as follows:

$$f_{\{q,k\}}(\mathbf{x}_i) = \mathbf{W}_{\{q,k\}}(\mathbf{x}_i + \mathbf{p}_{\{q,k\}}^{(i)} + \mathbf{h}_u^{(i)}), \quad (6)$$

where $\mathbf{h}_u^{(i)}$ is the disentangled embedding of node u placed in position i , $\mathbf{W}_{\{q,k\}}$ is the projection matrix, \mathbf{x}_i is the corresponding feature from the last layer in the LLM, and $\mathbf{p}_{\{q,k\}}^{(i)}$ indicates the absolute or relative position embedding of position i . As an alternative, we use the following function for rotary position embedding:

$$f_{\{q,k\}}(\mathbf{x}_i) = \mathbf{R}_i \mathbf{W}_{\{q,k\}}(\mathbf{x}_i + \mathbf{h}_u^{(i)}), \quad (7)$$

where \mathbf{R}_i is the rotary matrix of the position embedding. Due to the varying position encoding of the injected features, our disentangled GNNs are encouraged to learn diverse knowledge, further enriching the semantic knowledge learned by different parts of GNNs. In addition, we also use the disentangled embedding to in value calculation by a similar way:

$$f_v(\mathbf{x}_i) = \mathbf{W}_v(\mathbf{x}_i + \mathbf{h}_u^{(i)}). \quad (8)$$

As such, our method incorporates neighborhood information from GNNs into each layer of the large language model LLM, enabling the LLM to benefit from a comprehensive understanding of the graph structure throughout the entire model. This injection of information in all layers facilitates a direct gradient flow to the GNNs, resulting in more accurate and informative gradient updates. This integration of language modeling and graph representation learning allows our model to leverage the contextual information captured by the LLM and the structural patterns learned by the GNNs, leading to effective learning and improved performance.

4.5. Fine-tuning Procedure

We summarize the fine-tuning procedure in Algorithm. 1. After generating node embedding by LLMs for the input graph at the beginning of the algorithm, the algorithm iteratively updates the parameters until convergence. Each iteration involves sampling a mini-batch from the training set and performing forward propagation through the disentangled GNN layers to capture both the graph structure and the text attributes. We follow the classic auto-regressive scheme to fine-tune. For each node in the mini-batch, we construct its prompt (refer to Section 5.1.2 for more details) as the input. Since we are only concerned about the

Table 2. The illustration of prompts and typical responses of different types of datasets. The blue parts indicate the texts populated based on the content of the dataset, the red parts indicate the embeddings obtained by fine-tuning based on neighborhood information. and the green parts indicate the prediction given by the LLM.

Dataset		Content
Citation Dataset	Prompt	Here is a paper title and abstract: (paper content). Some information about the references cited in this paper: (neighbor content). Task: there are following categories: (list of categories). Which category does this paper belong to? Output the most 1 possible category of this paper.
	Response Template	Based on the content of the paper, the most appropriate category for this paper would be (category prediction). . . .
E-commerce Dataset	Prompt	Here is a book description and title: (book content). Some information about the books frequently purchased together: (neighbor content). Task: there are following categories: (list of categories). Which category does this book belong to? Output the most 1 possible category of this book.
	Response Template	Based on the information of the book, the most appropriate category for this book would be (category prediction). . . .

LLM’s prediction for node categories, we design a response template (some examples are shown in Table 2) as part of the prompt. Consequently, the LLM can directly predict the category at the first token of the generation procedure. Then we can conveniently calculate the loss function with respect to the category prediction, which can be formulated as follows:

$$\mathcal{L} = -\log p(s_{\text{lab}}|s_{\text{pro}} + s_{\text{res}}), \quad (9)$$

where s_{lab} indicates the label token, s_{pro} and s_{res} indicate the sequences of the prompt and response template. We find that only calculating the loss function of the first token can work well in practice. As such, we can fine-tune the model in a more targeted manner, enabling it to learn the most useful information for downstream classification tasks. Back-propagation is performed after each node classification, and the parameters θ are updated through a gradient descent method when all nodes in the mini-batch are predicted. Through this iterative process, DGTL learns a text generation model that leverages the rich information present in text-attributed graphs.

5. Experiments

5.1. Experimental Setup

5.1.1. DATASETS

We conducted experiments on benchmark TAGs to assess the effectiveness of our framework. These TAGs encompass both citation networks and e-commerce networks, providing a diverse range of domains for evaluation.

For the citation networks, we utilize widely recognized datasets including Cora and PubMed (Sen et al., 2008). The Cora dataset focuses on computer science research papers. PubMed, on the other hand, is specific to biomedical literature, making it particularly relevant for medical-related

tasks. These datasets consist of academic papers as nodes and their citation relationships as edges. Each paper has its title and abstract as text information. The task on these citation networks is to determine the category of the papers.

In addition, we also incorporated e-commerce graphs into our evaluation. Specifically, we employ Book-History dataset (sktsherlock, 2023), which contains a rich variety of historical books. In the dataset, nodes represent books, while the edges indicate the connected books are frequently purchased together. Each book also has its description as the text information. The task on the e-commerce network is to determine the category of the books.

The statistics of the datasets are shown in Table 1. For all datasets, we follow the classical semi-supervised learning setting, i.e., we random select 20 nodes of each category as the training set, and randomly select 1,000 nodes from the rest nodes as the test set.

5.1.2. IMPLEMENTATIONS

In our experiments, we employed Llama-2-13B-chat (Touvron et al., 2023b) as the backbone LLM. Llama2 is a representative LLM known for its impressive language modeling capabilities and extensive pre-training on a vast corpus of text data. By utilizing Llama2 as our backbone LLM, we aimed to validate the effectiveness of our approach when collaborated with LLMs.

To tailor our framework to different datasets and tasks, we designed specific prompts for different types of datasets. The prompts served as initial instructions or cues provided to the LLM to guide our fine-tuning for the target task. The details of the designed prompts are summarized in Table 2. The table outlines the specific prompt structure, including any additional instructions or formatting applied, for each dataset used in our experiments. By customizing

Table 3. Performance comparison with GNN predictor baselines.

Dataset	Cora			PubMed		
GNN backbone	GCN	GAT	MLP	GCN	GAT	MLP
TF-IDF	81.99 \pm 0.63	82.30 \pm 0.65	67.18 \pm 1.01	78.86 \pm 2.00	77.65 \pm 0.91	71.07 \pm 0.78
Word2Vec	74.01 \pm 1.24	72.32 \pm 0.17	55.34 \pm 1.31	70.10 \pm 1.80	69.30 \pm 0.66	63.48 \pm 0.54
Deberta-base	48.49 \pm 1.86	51.02 \pm 1.22	30.40 \pm 0.57	62.08 \pm 0.06	62.63 \pm 0.27	53.50 \pm 0.43
Fine-tuned Deberta-base	59.23 \pm 1.16	57.38 \pm 2.01	30.98 \pm 0.68	62.12 \pm 0.07	61.57 \pm 0.07	53.65 \pm 0.26
Llama-7B	66.80 \pm 2.20	59.74 \pm 1.53	52.88 \pm 1.96	73.53 \pm 0.06	67.52 \pm 0.07	66.07 \pm 0.56
Sentence-BERT	82.20 \pm 0.49	82.77 \pm 0.59	74.26 \pm 1.44	81.01 \pm 1.32	79.08 \pm 0.07	76.66 \pm 0.50
e5-large	82.56 \pm 0.73	81.62 \pm 1.09	74.26 \pm 0.93	82.63 \pm 1.13	79.67 \pm 0.80	80.38 \pm 1.94
GLEM-GNN	48.49 \pm 1.86	51.02 \pm 1.22	-	62.08 \pm 0.06	62.63 \pm 0.27	-
GLEM-LM	59.23 \pm 1.16	57.38 \pm 2.01	-	62.12 \pm 0.07	61.57 \pm 0.07	-
DGTL	81.10 \pm 0.20	80.40 \pm 0.42	-	87.10 \pm 1.31	85.63 \pm 1.54	-

Table 4. Performance comparison with LLM predictor baselines and the ablation model.

Method	Cora	PubMed	Books-History
0-hop	60.80	62.30	17.70
Neighbor Summary	77.30	67.70	17.90
DGTL	81.10 \pm 0.20	87.10 \pm 1.54	55.70 \pm 0.89
DGTL w/o disen	80.67 \pm 0.47	85.00 \pm 1.47	55.40 \pm 1.04

the prompts, we ensured that the LLM could effectively adapt its pre-trained knowledge to the specific requirements and characteristics of each dataset.

For other hyper-parameters, we set the number of disentangled channels is 16 for citation datasets, and 8 for the e-commerce datasets. The number of hidden dimensions of each disentangled GNN channel is 32. The hyper-parameter to control the disentangled structure δ is 0.8. For optimization, we use the Adam optimizer with a learning rate of 0.001.

5.1.3. BASELINES

We compare our model with baselines from the following two categories.

- **GNN Predictors.** Following Chen et al. (Chen et al., 2023), we consider different language models to enhance the node features of the TAG, including Deberta (He et al., 2020), Llama (Touvron et al., 2023a), Sentence-BERT (Reimers & Gurevych, 2019), and e5 (Wang et al., 2022). Shallow embeddings TF-IDF and Word2vec are also included. We include GLEM (Zhao et al., 2022) as a baseline as well. For all these feature enhancer methods, three GNN backbones, including GCN (Kipf & Welling, 2017), GAT (Velićković et al., 2018), and MLP are adopted. For these baselines, we directly use the classification accuracy as the evaluation metric.
- **LLM predictors.** We also consider using LLM (Llama2-

13B-chat) as the predictors as our baselines, including using prompts without neighborhood information (0-hop prompt) and neighborhood summarization (Chen et al., 2023). For these methods, we use exact match scores (Rajpurkar et al., 2016), i.e., the percentage of predictions that match any one of the ground truth, as the metric. truth answers exactly.

5.2. Experimental Results

The result comparison with the GNN predictor baselines is shown in Table 3. From the table, we can find that GNN backbones are generally better than MLP backbone, indicating that structural information is essential to solve TAG tasks. Our method achieves comparable performance to SOTA baselines on Cora and outperforms all baselines on PubMed. Besides, our method offers a distinct advantage in terms of interpretability. Unlike the baselines, our method can provide natural language explanations for the predictions, enhancing the transparency and comprehensibility of the model’s decision-making process. This interpretability aspect is particularly valuable in scenarios where understanding the reasoning behind the predictions is crucial.

The result comparison with the LLM predictor baselines is shown in Table 4. Although neighbor summary prompt achieves better performance than 0-hop prompt by considering neighborhood information, our method outperforms these baselines by a large margin, indicating that our method enables LLMs to effectively learn and utilize task-relevant knowledge and benefits the downstream task prediction.

5.3. Interpretability with Structural Information

We showcase some examples to illustrate the interpretation of our method. The results as well as the comparison with two LLM predictor baselines are shown in Table 5.

The results indicate that our proposed method is more capable of predicting the correct label on the target node while

Table 5. The interpretation comparison with LLM predictor methods on two cases. Case 1: a paper classification task on Cora dataset. Case 2: a book classification task on Book-History dataset. The blue parts are the labels. The red parts are the wrong predictions, while the green parts are the correct predictions and the corresponding structurally relevant interpretations.

Text	0-hop	Neighbor Summary	DGTL
(label: rule learning) Here is a paper title and abstract: Limited Dual Path Execution. : This work presents a hybrid branch predictor scheme that uses a limited form of dual path execution along with dynamic branch prediction to improve execution times. . . .	Based on the content of the paper, the most appropriate category for this paper would be "theoretical" or "probabilistic methods". The paper presents a new approach to branch prediction that uses a hybrid predictor model that combines dynamic branch prediction with limited dual path execution. . . .	Based on the content of the paper, the most appropriate category for this paper would be 'theory'. The paper presents a new approach to branch prediction and speculative execution, and proposes a hybrid predictor model that combines dynamic branch prediction and dual path execution. . . .	Based on the content of the paper, the most appropriate category for this paper would be 'rule learning'. The paper is discussing the use of a hybrid branch predictor scheme that uses a limited form of dual path execution along with dynamic branch prediction to improve execution times, which is a type of rule-based approach. The paper also cites several references related to rule-based machine learning. . . .
(label:military) Here is a book description and title: Description: Between Frederick Buechners profound autobiography Sacred Journey and Russell Bakers warm and humorous Growing Up, Matt Matthews One Thousand Miles charts a compelling journey all its own through the past and the present. . . .	Based on the information provided, the book "One Thousand Miles" by Matt Matthews belongs to the category of "M memoir/Biography" or "History". The book is described as a "compelling journey through the past and present" that explores themes of identity, family, and the human experience, and draws on a wide range of sources including autobiography, history, and culture. . . .	Based on the book description, the most appropriate category for "Matt Matthews One Thousand Miles" would be "Americas" or "North American Historical Study." The book is described as a story of a son's pilgrimage into the heart of his father's story, set against the backdrop of World War II and the generation that endured it. . . .	Based on the information of the book, the most appropriate category for this book would be 'Military History'. The book is about a son's journey to understand his father's experience during WWII, which is a historical event. There are also other books that are frequently purchased together, which are related to military history, such as 'The Second World War' and 'The Cold War'. Therefore, the most appropriate category for this book would be 'Military History'.

the other methods fail. Moreover, our method also generates explanations related to neighborhood information after giving the prediction. For example, in the first case, our method predicts the paper belongs to "rule learning", and a supporting evidence for this prediction is "the paper also cites several references related to rule-based machine learning." This supporting evidence is exactly derived from the neighborhood information on the TAG, which demonstrates that our method can effectively capture the semantic information on the graph to make more accurate predictions. In case 2, our method predicts that the book belongs to military history. In addition to making predictions based on the content of the book, it also utilize the neighborhood information in the TAG to assist in the prediction process, as it states "there are also other books that are frequently purchased together, which are related to military history, such as 'The Second World War' and 'The Cold War'".

The shown cases demonstrate that our method generates interpretations for classification predictions that are accurate and structurally relevant. By contrast, the baselines produce incorrect predictions without using structural information for interpretation. Our method effectively integrates the graph's structural information in LLM generation, providing meaningful insights and justifications for the classification results. The experimental results highlight the importance

of incorporating structural information in achieving accurate and interpretable predictions in TAG tasks. More importantly, through our approach, humans can harness intuitive interpretations based on graph structures when using LLMs to tackle TAG problems. This greatly enhances the utilization of knowledge embedded in TAG and unleashes the potential of LLMs on graphs.

5.4. Ablation Study

In this section, we evaluate the effectiveness of the disentanglement component of our method through an ablation study. The results are also shown in Table 4. The results demonstrate that the disentanglement is greatly beneficial and crucial for our method to learn better structural information and enables the downstream LLM to give more accurate predictions accordingly.

6. Conclusion

In conclusion, this paper addresses the challenge of effectively integrating structural information into large language models (LLMs) for text-attributed graph (TAG) tasks. We propose the Disentangled Graph Text Learner (DGTL) model, which leverages tailored disentangled graph neural network layers to capture complex graph structure infor-

mation in TAGs. Our method enhances the reasoning and prediction capabilities of LLMs while providing natural language explanations for model predictions, which is crucial for interpretability in TAG tasks. Through extensive experiments on various TAG benchmarks, we demonstrate that DGTL achieves competitive performance compared to state-of-the-art baselines while offering human-understandable explanations for model predictions. Our work contributes to advancing the field of TAG analysis by harnessing the power of LLMs and improving their interpretability for real-world applications.

References

- Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.
- Chen, Z., Mao, H., Li, H., Jin, W., Wen, H., Wei, X., Wang, S., Yin, D., Fan, W., Liu, H., et al. Exploring the potential of large language models (llms) in learning on graphs. *arXiv preprint arXiv:2307.03393*, 2023.
- Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C.-M., Chen, W., et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023.
- Duan, K., Liu, Q., Chua, T.-S., Yan, S., Ooi, W. T., Xie, Q., and He, J. Simteg: A frustratingly simple approach improves textual graph learning. *arXiv preprint arXiv:2308.02565*, 2023.
- Gao, C., Lan, X., Lu, Z., Mao, J., Piao, J., Wang, H., Jin, D., and Li, Y. S3: Social-network simulation system with large language model-empowered agents. *arXiv preprint arXiv:2307.14984*, 2023.
- Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272, 2017.
- Guo, J., Du, L., and Liu, H. Gpt4graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking. *arXiv preprint arXiv:2305.15066*, 2023.
- Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, 2017.
- He, P., Liu, X., Gao, J., and Chen, W. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.
- Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in neural information processing systems*, 2020.
- Huang, X., Han, K., Bao, D., Tao, Q., Zhang, Z., Yang, Y., and Zhu, Q. Prompt-based node feature extractor for few-shot learning on text-attributed graphs. *arXiv preprint arXiv:2309.02848*, 2023.
- Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.
- Liu, C. and Wu, B. Evaluating large language models on graphs: Performance insights and comparative analysis. *arXiv preprint arXiv:2308.11224*, 2023.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.
- Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, 2016.
- Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.
- Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., and Eliassi-Rad, T. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.
- sktsherlock. Tag-benchmark. Online, Oct 2023. URL <https://github.com/sktsherlock/TAG-Benchmark>.
- Thirunavukarasu, A. J., Ting, D. S. J., Elangovan, K., Gutierrez, L., Tan, T. F., and Ting, D. S. W. Large language models in medicine. *Nature medicine*, 29(8):1930–1940, 2023.
- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.
- Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P.,

- Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.
- Wang, H., Feng, S., He, T., Tan, Z., Han, X., and Tsvetkov, Y. Can language models solve graph problems in natural language? *arXiv preprint arXiv:2305.10037*, 2023.
- Wang, L., Yang, N., Huang, X., Jiao, B., Yang, L., Jiang, D., Majumder, R., and Wei, F. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.
- Wu, L., Zheng, Z., Qiu, Z., Wang, H., Gu, H., Shen, T., Qin, C., Zhu, C., Zhu, H., Liu, Q., et al. A survey on large language models for recommendation. *arXiv preprint arXiv:2305.19860*, 2023.
- Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *7th International Conference on Learning Representations*, 2019.
- Yang, Y. and Cui, X. Bert-enhanced text graph neural network for classification. *Entropy*, 23(11):1536, 2021.
- Ye, R., Zhang, C., Wang, R., Xu, S., and Zhang, Y. Natural language is all a graph needs. *arXiv preprint arXiv:2308.07134*, 2023.
- Zhang, J., Zhang, H., Xia, C., and Sun, L. Graph-bert: Only attention is needed for learning graph representations. *arXiv preprint arXiv:2001.05140*, 2020.
- Zhang, Y., Jin, R., and Zhou, Z.-H. Understanding bag-of-words model: a statistical framework. *International journal of machine learning and cybernetics*, 1:43–52, 2010.
- Zhang, Z., Li, H., Zhang, Z., Qin, Y., Wang, X., and Zhu, W. Large graph models: A perspective. *arXiv preprint arXiv:2308.14522*, 2023.
- Zhao, J., Qu, M., Li, C., Yan, H., Liu, Q., Li, R., Xie, X., and Tang, J. Learning on large-scale text-attributed graphs via variational inference. In *The Eleventh International Conference on Learning Representations*, 2022.
- Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.