Understanding Survey Paper Taxonomy about Large Language Models via Graph Representation Learning

Jun Zhuang
Department of Computer Science
Boise State University
junzhuang@boisestate.edu

Casey Kennington
Department of Computer Science
Boise State University
caseykennington@boisestate.edu

Abstract

As new research on Large Language Models (LLMs) continues, it is difficult to keep up with new research and models. To help researchers synthesize the new research many have written survey papers, but even those have become numerous. In this paper, we develop a method to automatically assign survey papers to a taxonomy. We collect the metadata of 144 LLM survey papers and explore three paradigms to classify papers within the taxonomy. Our work indicates that leveraging graph structure information on co-category graphs can significantly outperform the language models in two paradigms; pre-trained language models' fine-tuning and zero-shot/few-shot classifications using LLMs. We find that our model surpasses an average human recognition level and that fine-tuning LLMs using weak labels generated by a smaller model, such as the GCN in this study, can be more effective than using ground-truth labels, revealing the potential of weak-to-strong generalization in the taxonomy classification task.

1 Introduction

Collective attention in Natural Language Processing (NLP)—and the wider public—has turned to Large Language Models (LLMs) (Lyu et al., 2022; Xie et al., 2023; Jin et al., 2024). It has become so difficult to keep up with the proliferation of new models that many researchers have written survey papers to help synthesize research progress. Survey papers are often crucial for newcomers to gain an in-depth understanding of the research evolution. However, volumes of survey papers have become unruly for researchers—especially newcomers—to sift through. As illustrated in Figure 1, the number of survey papers has been increasing significantly. This leads to our research question: Is it possible to automatically reduce the barriers for newcomers in a way that can keep up with the constant influx of new information? This question underscores one of the challenges in scholarly document processing (Chandrasekaran et al., 2020).

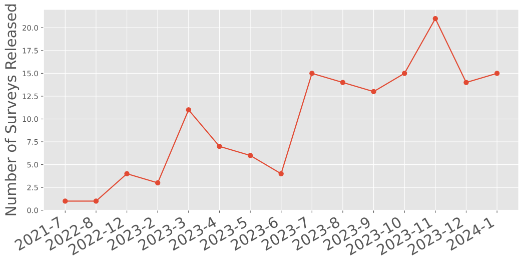


Figure 1: Trends of survey papers about large language models since 2021. Numbers reflect the year and month (e.g., 2023-3 is March 2023).

In this work, we address the above question by developing a method that can automatically assign survey papers to a taxonomy. Such a taxonomy will help researchers see new trends in the field and focus on specific survey papers that are relevant to their research. Classifying papers into a taxonomy may seem an ordinary task, but it is actually quite challenging for the following reasons:

1. Our dataset contains 144 papers. While this number for the survey papers is uncommonly large, the number of instances in the dataset is still relatively small.
2. We propose a new taxonomy for the collected survey papers, where the distribution of each category is not uniform, which leads to a substantial class imbalance issue.
3. Authors usually use similar terminologies to describe LLMs in the title and abstract. Such textual similarity introduces significant difficulties in taxonomy classifications.

To answer our research questions, we investigate three types of attributed graphs: text graphs, co-author graphs, and co-category graphs. Extensive experiments indicate that leveraging graph structure information of co-category graphs can help better classify the survey papers to the corresponding categories in the proposed taxonomy.

Proceedings of the Fourth Workshop on Scholarly Document Processing (SDP 2024), pages 58–69
August 16, 2024 ©2024 Association for Computational Linguistics

Moreover, we validate that graph representation learning (GRL) can outperform language models in two paradigms, fine-tuning pre-trained language models and zero-shot/few-shot classifications using LLMs. Inspired by a recent study, which indicates that leveraging weak labels, which are generated by smaller (weaker) models, may help enhance the performance of larger (stronger) models (Burns et al., 2023), we further examine whether using weak labels, which are generated by GNNs in this study, in the fine-tuning paradigm can help the pre-trained language models. The experiments demonstrate that fine-tuning using weak labels can exceed that using ground-truth labels. For the latter paradigm, we use the results of human recognition as the baseline. The analysis demonstrates that GRL achieves higher accuracy and F1 scores, and even surpasses the average human recognition level by a substantial margin. Overall, our primary contributions can be summarized as follows:

- We collected and analyzed 144 survey papers about LLMs and their metadata.[1]
- We propose a new taxonomy for categorizing the survey papers, which will be helpful for the research community, particularly newcomers and multidisciplinary research teams.
- Extensive experiments demonstrate that graph representation learning on co-category graph structure can effectively classify the papers and substantially outperform the language models and average human recognition level on a relatively small and class-imbalanced dataset with high textual similarity.
- Our results also reveal the potential of fine-tuning pre-trained models using weak labels.

2 Related Work

Taxonomy Classification Conventional taxonomy classification is a subset of Automatic Taxonomy Generation (ATG), which aims to generate or augment taxonomy for processing scholarly documents (Krishnapuram and Kummamuru, 2003; Smirnova and Mayr, 2023; Lo et al., 2023; Kinney et al., 2023; Lee et al., 2024). The main challenge in ATG is to cluster the unlabeled topics into different hierarchical structures. Thus, most existing methods in ATG are clustering-based methods. Zamir and Etzioni (1998) design a mechanism, Grouper, that dynamically groups and labels the search results. Vaithyanathan and Dom (1999) propose a model to generate hierarchical clusters. Lawrie et al. (2001) discover the relationship among words to generate concept hierarchies. Within these methods, a subset, called co-clustering, clusters keywords and documents simultaneously (Frigui and Nasraoui, 2002; Kummamuru et al., 2003). Different from ATG, we classify survey papers into corresponding categories in the proposed taxonomy on small and class-imbalanced datasets with highly similar terminologies.

Graph Representation Learning Graph representation learning (GRL) is a powerful approach for learning the representation in graph-structure data (Zhou et al., 2020), whereas most recent works achieve this goal using Graph Neural Networks (GNNs) (Defferrard et al., 2016; Veličković et al., 2018; Xu et al., 2018; Liu et al., 2022, 2021). Bruna et al. (2013) first introduce a significant advancement in convolution operations applied to graph data using both spatial method and spectral methods. To improve efficiency, Kipf and Welling (2016) simplify graph convolutions to a first-order polynomial while yielding superior performance in semi-supervised learning. Hamilton et al. (2017) propose an inductive-learning approach that aggregates node features from corresponding fixed-size local neighbors. Regarding ATG via GNNs, several works attempt to generate or expand taxonomies using new proposed graph models (Shen et al., 2020; Xu et al., 2022). Compared to model-based approaches, we first improve the performance of taxonomy classification from data-driven perspectives.

3 Methodology

In this section, we first introduce the procedure of data collection and then explore the metadata. We further explain the process of constructing three types of attributed graphs and how we learn graph representation via graph neural networks.

3.1 Data Collection and Exploration

We scraped the metadata of survey papers about LLMs from arXiv and further manually supplemented the metadata from Google Scholar and the ACL anthology. The papers range from July 2021 to January 2024. Given these survey papers, we designed a taxonomy and assigned each paper to a corresponding category within the taxonomy. Our motivation is that a reasonable taxonomy can provide a clear hierarchy of concepts for readers to bet-

[1] We release our datasets and source code at https://github.com/junzhuang-code/TaxonomyClassification

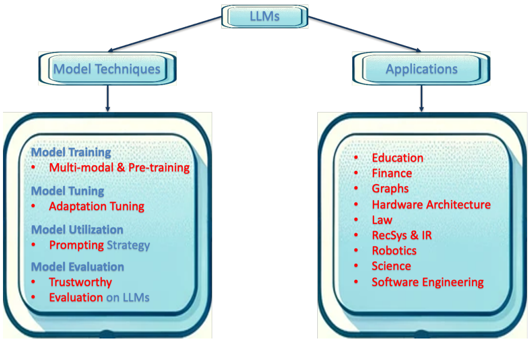


Figure 2: The mind map of survey papers about large language models. Besides "Comprehensive" and "Others" that are not included in the mind map, we highlight fourteen categories in our proposed taxonomy. The total number of categories for the 144 papers is sixteen.

ter understand the relationship among a large number of survey papers. Though survey papers can be taxonomized differently, we noticed two broad categories: *applications* and *model techniques*. The *applications* category further sub-divides into specific domains of focus (e.g., education or science), whereas *model techniques* further sub-divides into ways of effecting models (e.g., fine-tuning).

We visualize our proposed taxonomy and highlight fourteen classes, i.e., the leaf nodes, in Figure 2. The total classes in the labels are sixteen, including *comprehensive* and *others* (not shown in the figure). To better understand the distribution of the classes, we present the class distribution in Figure 3. The distribution indicates that the class is extremely imbalanced, introducing a challenge to the taxonomy classification task.

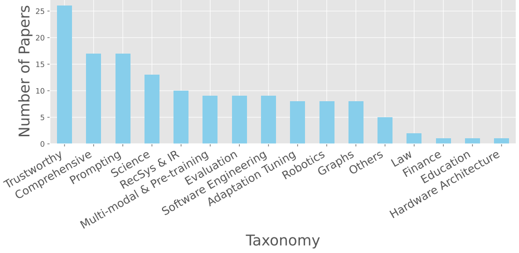


Figure 3: Distribution of classes in the taxonomy.

After visualizing the proposed taxonomy, we further explain the motivation for proposing a new taxonomy instead of using the arXiv categories. In Figure 4, we present the distribution of survey papers across different arXiv categories. Top-2 frequent categories are `cs.CL` (Computation and Language), and `cs.AI` (Artificial Intelligence), which means that most authors choose these two categories for their works. However, these choices cannot help readers to better distinguish the survey papers. For example, papers related to model techniques are indistinguishable in arXiv categories. Thus, designing a new taxonomy is an essential step in this study.

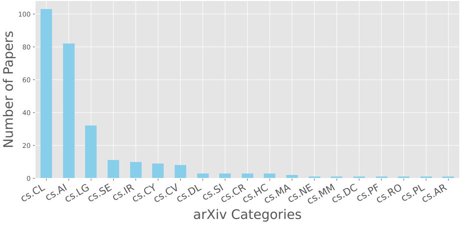


Figure 4: Distribution of survey papers that we found across different arXiv categories.

We also present the word frequency in Figure 5 to show which words have been frequently used in abstracts. These distributions suggest that the abstracts of these papers contain many similar terms, which increases the difficulty of text classification.

Attributes	**Descriptions**
Taxonomy	Proposed taxonomy
Title	Paper title
Authors	Lists of author's name
Release Date	First released date
Links	Links of papers
Paper ID	The arXiv paper ID
Categories	The arXiv category
Summary	Abstract of papers

Table 1: Data attributes and descriptions.

Overall, we present the data description of their attributes in Table 1. After designing the taxonomy and building the dataset, we explain how we classify documents into the taxonomy categories in the following section.

3.2 Building Attributed Graphs

The goal of building the graphs is to utilize the graph structure information to classify the taxonomy. Before building the graphs, We first define the attributed graphs as follows:

Definition 1 *An attributed graph $\mathcal{G}$ is a topological structure that represents the relationships among vertices associated with attributes. $\mathcal{G}$ consists of a set of vertices $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, where N is the number of vertices in $\mathcal{G}$.*

Given the Definition 1, we further define the matrix representation of an attributed graph as follows:

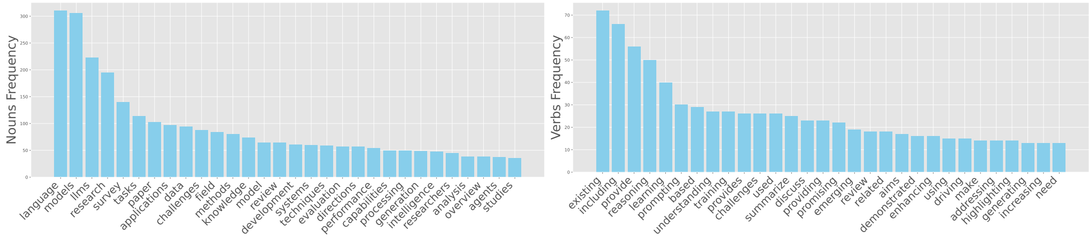


Figure 5: Top 30 keywords frequency in the summary of survey papers.

Definition 2 *Given an attributed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, the topological relationship among vertices can be represented by a symmetric adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$. Each vertex contains an attribute vector, a.k.a., a feature vector. All feature vectors constitute a feature matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$, where d is the number of features for each vertex. Thus, the matrix representation of an attributed graph can be formulated as $\mathcal{G}(\mathbf{A}, \mathbf{X})$.*

Based on the above definitions, we build the graph by creating the term frequency-inverse document frequency (TF-IDF) feature matrices for both title and summary (i.e., abstract) columns, where the term frequency denotes the word frequency in the document, and inverse document frequency denotes the log-scaled inverse fraction of the number of documents containing the word. TF-IDF matrix is commonly used for text classification tasks because it helps capture the distinctive words that can indicate specific classes (Yao et al., 2019). After establishing the TF-IDF matrices, we apply one-hot encoding on the arXiv's categories and then combine three matrices along the feature dimension to build the feature matrix $\mathbf{X}$.

To leverage the topological information among vertices, we proceed to construct the graph structures to connect the attribute vectors. In this study, we are interested in three types of graphs: text graph, co-author graph, and co-category graph. We explain each type as follows.

Text Graph We follow the same settings as TextGCN (Yao et al., 2019) to build the text graph. Specifically, the edges of the text graph are built based on word occurrence (paper-word edges) in the paper's text data, including both title and summary, and word co-occurrence (word-word edges) in the whole text corpus. To obtain the global word co-occurrence information, we slide a fixed-size window on all papers' text data. Moreover, we calculate the edge weight between a paper vertex and a word vertex using the TF-IDF value of the word in the paper and calculate the edge weight between two-word vertices using point-wise mutual information (PMI), a popular metric to measure the associations between two words. Note that in the text graph, we don't use the above feature matrix because only paper vertices contain attribute vectors. To retain consistency, we set all values in the feature matrix as one. For the same reason, only the paper vertices are assigned labels, whereas all word vertices are labeled as a new class, which is not touched during the training or testing phase.

Co-author Graph In the co-author graph, we introduce an edge connecting two vertices (papers) if they share at least one common author.

Co-category Graph In the co-category graph, an edge is added between two vertices with at least one common arXiv category. In the co-authorship and co-category graphs, each vertex is assigned one class (taxonomy) as the label. Note that in this study all edges are undirected.

3.3 Taxonomy Classification via Graph Representation Learning

Given the well-built attributed graphs $\mathcal{G}(\mathbf{A}, \mathbf{X})$, we aim to investigate whether graph representation learning (GRL) using graph neural networks (GNNs) can help classify survey papers into the taxonomy. Before feeding the matrix representation, $\mathbf{A}$ and $\mathbf{X}$, of the attributed graphs $\mathcal{G}$ into GNNs, we first preprocess the adjacency matrix $\mathbf{A}$ as follows:

$$\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}, \qquad (1)$$

where $\tilde{\mathbf{A}} = \mathbf{A} + I_N$, $\tilde{\mathbf{D}} = \mathbf{D} + I_N$. I_N is an identity matrix. $\mathbf{D}_{i,i} = \sum_j \mathbf{A}_{i,j}$ is a diagonal degree matrix.

After preprocessing, we utilize GNNs to learn graph representation. The layer-wise message-passing mechanism of GNNs can be generally formulated as follows:

$$f_{\mathbf{W}^{(l)}}\left(\hat{\mathbf{A}}, \mathbf{H}^{(l)}\right) = \sigma\left(\hat{\mathbf{A}} \mathbf{H}^{(l)} \mathbf{W}^{(l)}\right), \qquad (2)$$

where $\mathbf{H}^{(l)}$ is a node hidden representation in the l-th layer. The dimension of $\mathbf{H}^{(l)}$ in the input layer, middle layer, and output layer is the number of features d, hidden units h, and classes K, respectively. $\mathbf{H}^{(0)} = \mathbf{X}$. $\mathbf{W}^{(l)}$ is the weight matrix in the l-th layer. σ denotes a non-linear activation function, such as ReLU.

In general node classification tasks, GNNs are trained with ground-truth labels $\mathbf{Y} \in \mathbb{R}^{N \times 1}$. In this study, we assign ground-truth labels based on our proposed taxonomy. To simplify the problem, each paper is assigned one primary category as the label, even if the paper sometimes may belong to more than one category. During training, we optimize GNNs with cross-entropy.

In brief, we address the Taxonomy Classification problem via GRL approaches in this study and formally state the problem as follows:

Problem 1 *After building an attributed graph $\mathcal{G}(\hat{\mathbf{A}}, \mathbf{X})$ and the ground-truth labels $\mathbf{Y}$ based on the survey metadata, we train a graph neural network (*GNN*) on the train data and evaluate the taxonomy classification performance on the test data. Our goal is to design a method to better understand (classify) the taxonomy of the survey papers.*

4 Experiment

In this section, we evaluate the graph representation learning (GRL)'s effectiveness compared with two paradigms using language models.

Subsets	Graphs	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert F\rvert$	$\lvert C\rvert$
$Data_{Nov23}$	Text	737	94,943	737	16
	Co-author	112	204	3,065	15
	Co-category	112	4,908	3,065	15
$Data_{Jan24}$	Text	951	137,709	951	17
	Co-author	144	332	3,542	16
	Co-category	144	8,140	3,542	16
$Data_{subset}$	Text	905	128,575	905	12
	Co-author	134	302	3,394	11
	Co-category	134	6,964	3,394	11

Table 2: Statistics of graph datasets. $|\mathcal{V}|$, $|\mathcal{E}|$, $|F|$, and $|C|$ denote the number of nodes, edges, features, and classes, respectively.

Experimental Settings To examine the generalization of our method on various graph structures, we investigate three types of attributed graphs: text graphs, co-author graphs, and co-category graphs, and compare the classification performance of GRL with that of fine-tuning pre-trained language models across three subsets of our data. Both $Data_{Nov23}$ and $Data_{Jan24}$ contain survey papers collected at the end of corresponding months (November 2023 and January 2024). $Data_{Jan24}$ includes a new category, *Hardware Architecture*. We further construct the third subset $Data_{subset}$ by removing some proposed categories with fewer instances in $Data_{Jan24}$; these categories are *Law*, *Finance*, *Education*, *Hardware Architecture*, and *Others*. The motivation for constructing three subsets is to validate the generalization of our method across different subsets since the classification performance may significantly change on small datasets. Also, new categories may emerge at any period because research on LLMs is developing rapidly, and so are related survey papers. For example, a new category, *Hardware Architecture*, emerges in $Data_{Jan24}$. The change of categories may affect the performance as well. Therefore, we investigate our method on three subsets that contain different categories. The statistics of our dataset and corresponding attributed graphs are presented in Table 2. Recall that the text graph consists of paper vertices and word vertices, and thus contains one additional class because all word vertices are labeled as a new class, which is not touched during the training or testing phase.

To evaluate our model, we split the train, validation, and test data as 60%, 20%, and 20%. Due to the potential for random splits to result in an easier task for our model, we ran the experiments five times using random seed IDs from 0 to 4 and reported the mean values with corresponding standard deviations, mean (std). We evaluate the classification performance by accuracy and weighted f1 score. Accuracy is a common metric on classification tasks, whereas the weighted f1 score provides a balanced measure of the class-imbalanced dataset.

4.1 Leveraging Graph Structure Information for Taxonomy Classification

We investigate whether leveraging the graph structure information can help better classify the papers to their corresponding categories in the proposed taxonomy. In this experiment, we construct the attributed graphs based on the text data (including the title and summary) and the relationship of the co-authorship and co-category. To examine the generalization of GRL, we employ GCN (Kipf and Welling, 2016) as a backbone GNN on various graph structures across three subsets. According to Table 3, GNNs fail to learn graph representation on

	$Data_{Nov23}$		$Data_{Jan24}$		$Data_{subset}$	
	Accuracy	Weighted-F1	Accuracy	Weighted-F1	Accuracy	Weighted-F1
Text	20.91 (5.45)	14.20 (4.41)	17.86 (7.14)	16.31 (4.49)	23.08 (4.87)	18.82 (1.50)
Co-author	33.04 (8.06)	33.06 (8.69)	20.00 (8.56)	19.24 (8.79)	29.63 (7.03)	29.24 (6.02)
Co-category (All)	63.48 (18.36)	62.82 (16.96)	75.17 (5.52)	74.60 (4.81)	**79.26 (6.87)**	**77.88 (7.52)**
Co-category (Rm `cs.CL`)	70.43 (9.28)	68.46 (9.63)	67.59 (15.36)	65.81 (17.03)	76.30 (12.31)	73.83 (14.53)
Co-category (Rm `cs.AI`)	**73.91 (18.03)**	**72.41 (18.28)**	**75.86 (8.99)**	**75.79 (9.62)**	77.04 (3.63)	74.15 (4.11)
Co-category (Rm `cs.CL`, `cs.AI`)	26.09 (10.65)	20.19 (10.56)	37.93 (8.45)	35.97 (7.92)	49.63 (7.63)	47.32 (7.59)
Co-category (Rm `cs.IR`)	63.48 (18.36)	62.82 (16.96)	75.17 (5.52)	74.60 (4.81)	79.26 (6.87)	77.88 (7.52)
Co-category (Rm `cs.RO`)	63.48 (18.36)	62.82 (16.96)	75.17 (5.52)	74.60 (4.81)	79.26 (6.87)	77.88 (7.52)
Co-category (Rm `cs.SE`)	65.22 (11.00)	63.21 (10.37)	74.48 (8.33)	75.10 (6.77)	82.96 (6.02)	82.93 (6.22)
Co-category (Rm `cs.IR`, `cs.RO`)	63.48 (18.36)	62.82 (16.96)	75.17 (5.52)	74.60 (4.81)	79.26 (6.87)	77.88 (7.52)
Co-category (Rm `cs.IR`, `cs.SE`)	65.22 (11.00)	63.21 (10.37)	74.48 (8.33)	75.10 (6.77)	82.96 (6.02)	82.93 (6.22)
Co-category (Rm `cs.RO`, `cs.SE`)	65.22 (11.00)	63.21 (10.37)	74.48 (8.33)	75.10 (6.77)	82.96 (6.02)	82.93 (6.22)
Co-category (Rm `cs.IR`, `cs.RO`, `cs.SE`)	65.22 (11.00)	63.21 (10.37)	74.48 (8.33)	75.10 (6.77)	82.96 (6.02)	82.93 (6.22)

Table 3: Evaluation of graph representation learning on three types of attributed graphs across three subsets of our data. We also conducted ablation studies on the graph structure of co-category graphs by removing (denoted by Rm) some arXiv categories. We ran the experiments five times and presented the mean (std).

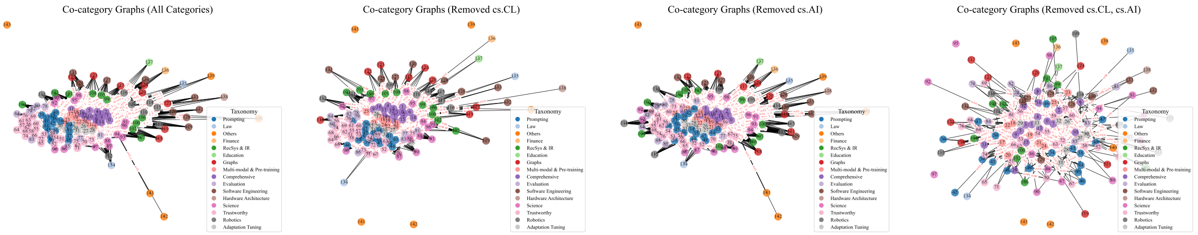

Co-category Graphs (All Categories)
Co-category Graphs (Removed cs.CL)
Co-category Graphs (Removed cs.AI)
Co-category Graphs (Removed cs.CL, cs.AI)
Taxonomy
Prompting
Law
Others
Finance
RecSys & IR
Education
Graphs
Multi-modal & Pre-training
Comprehensive
Evaluation
Software Engineering
Hardware Architecture
Science
Trustworthy
Robotics
Adaptation Tuning

Figure 6: Visualization of four graph structures on co-category graphs by removing the categories.

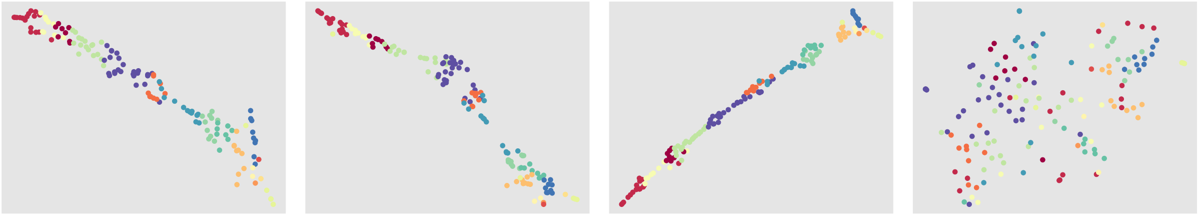
both the text graph and the co-author graph. For the text graph, we argue that the degradation may be caused by excessively similar words in the summary of survey papers. When constructing the text graph, these word vertices connect with many paper vertices, resulting in the paper vertices being less distinguishable. For the co-author graph, we conjecture that it is challenging to categorize papers solely based on the sparse co-authorship in this dataset. Furthermore, we observe that some co-authorships come from a common mentor in the same lab whereas two first authors work on the survey papers in two distinct categories. These reasons weaken the effectiveness of using graph structure information. GNNs, in contrast, are very reliable (evaluated by both accuracy and weighted F1 score) in most co-category graphs.

Ablation Analysis We further examine the graph structures of co-category graphs by conducting ablation studies. First, according to Figure 4, most papers are assigned as `cs.CL` and `cs.AI` in the arXiv categories. Thus, we study how the categories, `cs.CL` and `cs.AI`, affect the performance by muting these two categories in a combinatorial manner. In Table 3, we observe that GNNs can maintain a comparable performance after removing either `cs.CL` or `cs.AI`. However, the performance dramatically drops after removing both categories. This is possible since most node connections are significantly sparsified after these two categories are removed. Even though both `cs.CL` and `cs.AI` do not directly map to the existing classes, either one can connect the nodes and further strengthen the message-passing in GNNs, allowing GNNs to learn better node representations.

We visualize co-category graphs in $Data_{Jan24}$ in Figure 6. The visualization indicates that most nodes are clustered well even if we remove the category either `cs.CL` or `cs.AI`. However, after removing these two categories simultaneously, we observe that node classifications gradually become disordered and several nodes are then isolated. This visualization illustrates the effectiveness of GRL.

We further visualize GCNs' hidden representation on the above co-category graphs in $Data_{Jan24}$ in Figure 7. The figures show that the nodes are well-classified in the hidden space even if either the category `cs.CL` or `cs.AI` is removed. However, the distribution of nodes tends to become chaotic when both of these two categories are removed simultaneously, shown in Table 3.

For completeness, we conducted another abla-



(a) All Categories (b) Removed cs.CL (c) Removed cs.AI (d) Removed cs.CL, cs.AI

Figure 7: Visualization of GCNs' hidden representation on co-category graphs in $Data_{Jan24}$ via t-SNE. Each dot represents one node and is labeled with one color.

	$Data_{Nov23}$		$Data_{Jan24}$		$Data_{subset}$	
	Accuracy	Weighted-F1	Accuracy	Weighted-F1	Accuracy	Weighted-F1
BERT (Devlin et al., 2018)	30.43 (18.45)	25.70 (19.91)	43.45 (18.84)	41.50 (22.31)	58.74 (6.87)	57.51 (6.94)
RoBERTa (Liu et al., 2019)	41.74 (20.32)	39.17 (22.84)	35.86 (17.53)	27.23 (23.67)	25.93 (12.17)	17.02 (15.16)
DistilBERT (Sanh et al., 2019)	**57.39 (8.87)**	**55.59 (10.66)**	**53.10 (2.76)**	**52.07 (4.47)**	**59.26 (7.41)**	**58.15 (8.82)**
XLNet (Yang et al., 2019)	25.22 (16.82)	21.59 (20.54)	27.59 (14.30)	21.59 (15.39)	28.52 (9.37)	21.51 (9.65)
Electra (Clark et al., 2019)	23.04 (4.76)	19.06 (4.06)	44.83 (7.23)	42.01 (8.39)	20.01 (6.87)	12.03 (8.45)
Albert (Lan et al., 2019)	11.30 (8.06)	4.85 (7.21)	15.17 (4.68)	5.14 (2.87)	20.74 (9.83)	11.41 (11.15)
BART (Lewis et al., 2020)	51.30 (17.48)	50.30 (17.62)	51.72 (3.08)	50.62 (2.79)	58.25 (8.11)	57.68 (8.90)
DeBERTa (He et al., 2020)	24.78 (8.06)	19.61 (10.36)	26.21 (11.03)	20.92 (14.25)	25.93 (10.73)	24.30 (10.52)
Llama2 (Touvron et al., 2023)	14.48 (8.72)	4.77 (4.35)	19.22 (5.90)	6.03 (4.21)	23.45 (8.72)	12.59 (7.23)

Table 4: Evaluation of fine-tuning pre-trained language models on the text data across three subsets.

tion study to examine how the categories, cs.IR, cs.RO, and cs.SE, affect the classification performance as their names are similar to that of some classes in our proposed taxonomy (recall that our proposed taxonomy is not based on the arXiv categories). According to Table 3, the classification performances are well-maintained no matter which category is removed, whereas removing cs.SE does slightly change the results (highlighted by gray color). We argue that the results are reasonable since these removals only drop a small number of edges and don't break the topological relationships in the graph. Overall, these studies verify that leveraging the graph structure information in co-category graphs can positively contribute to the taxonomy classification.

4.2 Fine-tuning Pre-trained Language Models

After verifying the effectiveness of GRL, we continue to investigate whether GRL can transcend fine-tuning pre-trained language models on the text data across three subsets in the taxonomy classification task. To preprocess the text data, we follow the same setup as the cleaning process for text graphs. In this fine-tuning paradigm, we use various transformer-based (Vaswani et al., 2017) pre-trained language models as competing models, such as BERT (Devlin et al., 2018), which learns bidirectional representations and significantly enhances performance across a wide range of contextual understanding tasks. The results in Table 3 and Table 4 gave us an affirmative answer to the superiority of GRL. In Table 4, we further observe that medium-size language models, such as DistilBERT (Sanh et al., 2019), work better on smaller text data. However, the performance may dramatically drop when the model size is too small, such as Albert (Lan et al., 2019) or too large, such as Llama2 (Touvron et al., 2023). We conjecture that a smaller pre-trained model may be more sensitive to the domain shift issue (our dataset has distinct class distributions compared to that of the dataset used to pre-train the language models), whereas a large pre-trained model may suffer overfitting issues when it is fine-tuned on a smaller text data.

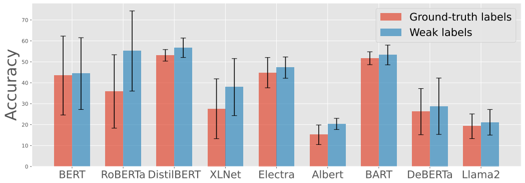


Figure 8: Comparison of fine-tuning pre-trained language models using ground-truth labels and weak labels. Whiskers denote standard deviation.

Fine-tuning with Weak Labels The above experiments confirmed that smaller ("weaker") GNNs can surpass larger ("stronger") language models in the taxonomy classification task. Recently, Burns et al. (2023). verified that training stronger models with pseudo labels, a.k.a. weak labels, generated by weaker models can enable the stronger models to achieve comparable performance as closely as those trained with ground-truth labels. In this experiment, we first generate weak labels by GCN on co-category graphs and then fine-tune pre-trained language models with weak labels. We present the results on $Data_{Jan24}$ in Figure 8 as an example. The results indicate that performance achieved through training with weak labels can surpass that of training with ground-truth labels. One possible reason is that training the model using noisy labels with a low noise ratio can be equivalent to a kind of regularization, improving the classification results (Zhuang and Al Hasan, 2022).[2]

This experiment demonstrates that leveraging weak labels generated by smaller models may effectively enhance the performance of larger models. This is one of the applications related to "weak-to-strong generalization" (Burns et al., 2023).

4.3 LLM Zero-shot/Few-shot Classification and Human Evaluation

	Accuracy	Weighted-F1
Human	**58.73 (19.16)**	**59.50 (19.13)**
Claude w.o. hints	11.61 (1.27)	12.66 (0.14)
Claude w. hints	10.27 (3.15)	12.81 (2.10)
GPT 3.5 w.o. hints	47.32 (3.25)	43.21 (4.33)
GPT 3.5 w. hints	**53.57 (2.81)**	**53.16 (3.13)**
GPT 4 w.o. hints	29.76 (7.22)	26.91 (9.66)
GPT 4 w. hints	33.04 (5.57)	27.78 (7.76)

Table 5: Evaluation of zero-shot and few-shot classification capabilities of large language models, Claude, GPT 3.5, and GPT 4. We compare these results with those from human recognition.

Major	CS	DS	Security	Math	Chemistry
#Students	9	4	2	2	1

Table 6: The number of students in each major. CS, DS, and Security denote the major in Computer Science, Data Science, and Cyber-security, respectively.

In this experiment, we evaluate zero-shot/few-shot classification capabilities of LLMs Claude (Bai et al., 2022), GPT 3.5 (Brown et al., 2020), and GPT 4 (Achiam et al., 2023), on the text data, which contains both title and summary, on $Data_{Nov23}$ as an example. We also compare the results with human participants. We recruited 18 students across five different majors in a graduate-level course. The number of students in each major is shown in Table 6. Each participant was given the titles and abstracts of survey papers and was asked to assign a category to each paper from our taxonomy. We present the mean value with the corresponding standard deviation in Table 5. For the LLMs, we ran the experiments five times. The standard deviation in human recognition is relatively large as some students do not have a strong technical background so they perform worse in this test. Among the LLMs, GPT 3.5 outperforms the other two models given that all models have not seen the data before (zero-shot). We further provide some hints to the models before classification (few-shot). For example, we release the keywords of the class "Trustworthy" to the models before classification. In this setting, both GPT 3.5 and GPT 4 can achieve higher accuracy and a weighted F1 score after obtaining some hints. In brief, GRL can outperform all three LLMs and human recognition, whereas these LLMs couldn't surmount human recognition, which reveals that LLMs still have much room to improve in taxonomy classification.

5 Conclusion

In this work, we aim to develop a method to automatically assign survey papers about Large Language Models (LLMs) to a taxonomy. To achieve this goal, we first collected the metadata of 144 LLM survey papers and proposed a new taxonomy for these papers. We further explored three paradigms to classify survey papers into the categories in the proposed taxonomy. After investigating three types of attributed graphs, we observed that leveraging graph structure information on co-category graphs can significantly help the taxonomy classification. Furthermore, our analysis validates that graph representation learning outperforms pre-trained language models' fine-tuning, zero-shot/few-shot classifications using LLMs, and even surpasses an average human recognition level. Last but not least, our experiments indicate that fine-tuning pre-trained language models using weak labels, which are generated by a weaker model, such as GCN, can be more effective than using ground-truth labels, revealing the potential for weak-to-strong generalization in the taxonomy classification task.

[2] Weak labels can be regarded as a kind of noisy label.

Limitations & Future Work Constructing a graph structure may encounter certain constraints. For instance, we build co-category graphs based on the arXiv categories. When papers come from distinct fields, such as biology, physics, and computer science, the graph structure may be very sparse, weakening the effectiveness of GRL.

In the future, our primary motivation extended from this study is to tailor GPT-based applications to assist readers in understanding survey papers more effectively. Also, we could consider different taxonomy designs, such as adopting the Computer Science Ontology or assigning multiple classes to an instance.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2013. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. 2023. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*.

Muthu Kumar Chandrasekaran, Guy Feigenblat, Dayne Freitag, Tirthankar Ghosal, Eduard Hovy, Philipp Mayr, Michal Shmueli-Scheuer, and Anita de Waard. 2020. Overview of the first workshop on scholarly document processing (sdp). In *Proceedings of the first workshop on scholarly document processing*, pages 1–6.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2019. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Hichem Frigui and Olfa Nasraoui. 2002. Simultaneous categorization of text documents and identification of cluster-dependent keywords. In *2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No. 02CH37291)*, volume 2, pages 1108–1113. IEEE.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Can Jin, Hongwu Peng, Shiyu Zhao, Zhenting Wang, Wujiang Xu, Ligong Han, Jiahui Zhao, Kai Zhong, Sanguthevar Rajasekaran, and Dimitris N Metaxas. 2024. Apeer: Automatic prompt engineering enhances large language model reranking. *arXiv preprint arXiv:2406.14449*.

Rodney Kinney, Chloe Anastasiades, Russell Authur, Iz Beltagy, Jonathan Bragg, Alexandra Buraczynski, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Arman Cohan, et al. 2023. The semantic scholar open data platform. *arXiv preprint arXiv:2301.10140*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Raghu Krishnapuram and Krishna Kummamuru. 2003. Automatic taxonomy generation: Issues and possibilities. In *International Fuzzy Systems Association World Congress*, pages 52–63. Springer.

Krishna Kummamuru, Ajay Dhawale, and Raghu Krishnapuram. 2003. Fuzzy co-clustering of documents and keywords. In *The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ'03.*, volume 2, pages 772–777. IEEE.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Dawn Lawrie, W Bruce Croft, and Arnold Rosenberg. 2001. Finding topic words for hierarchical summarization. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 349–357.

Yoonjoo Lee, Hyeonsu B Kang, Matt Latzke, Juho Kim, Jonathan Bragg, Joseph Chee Chang, and Pao Siangliulue. 2024. Paperweaver: Enriching topical paper alerts by contextualizing recommended papers with user-collected papers. *arXiv preprint arXiv:2403.02939*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Lihui Liu, Boxin Du, Heng Ji, ChengXiang Zhai, and Hanghang Tong. 2021. Neural-answering logical queries on knowledge graphs. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 1087–1097.

Lihui Liu, Boxin Du, Jiejun Xu, Yinglong Xia, and Hanghang Tong. 2022. Joint knowledge graph completion and question answering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1098–1108.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Kyle Lo, Joseph Chee Chang, Andrew Head, Jonathan Bragg, Amy X Zhang, Cassidy Trier, Chloe Anastasiades, Tal August, Russell Authur, Danielle Bragg, et al. 2023. The semantic reader project: Augmenting scholarly documents through ai-powered interactive reading interfaces. *arXiv preprint arXiv:2303.14334*.

Weimin Lyu, Songzhu Zheng, Tengfei Ma, and Chao Chen. 2022. A study of the attention abnormality in trojaned berts. *arXiv preprint arXiv:2205.08305*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Jiaming Shen, Zhihong Shen, Chenyan Xiong, Chi Wang, Kuansan Wang, and Jiawei Han. 2020. Taxoexpan: Self-supervised taxonomy expansion with position-enhanced graph neural network. In *Proceedings of the web conference 2020*, pages 486–497.

Nina Smirnova and Philipp Mayr. 2023. Embedding models for supervised automatic extraction and classification of named entities in scientific acknowledgements. *Scientometrics*, pages 1–25.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Shivakumar Vaithyanathan and Byron Dom. 1999. Model selection in unsupervised learning with applications to document clustering. In *ICML*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Tong Xie, Yuwei Wan, Wei Huang, Zhenyu Yin, Yixuan Liu, Shaozhou Wang, Qingyuan Linghu, Chunyu Kit, Clara Grazian, Wenjie Zhang, et al. 2023. Darwin series: Domain specific large language models for natural science. *arXiv preprint arXiv:2308.13565*.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.

Lingjun Xu, Shiyin Zhang, Guojie Song, Junshan Wang, Tianshu Wu, and Guojun Liu. 2022. Taxonomy-enhanced graph neural networks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2270–2279.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 32.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7370–7377.

Oren Zamir and Oren Etzioni. 1998. Web document clustering: A feasibility demonstration. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 46–54.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81.

Jun Zhuang and Mohammad Al Hasan. 2022. Robust node classification on graphs: Jointly from bayesian label transition and topology-based label propagation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2795–2805.

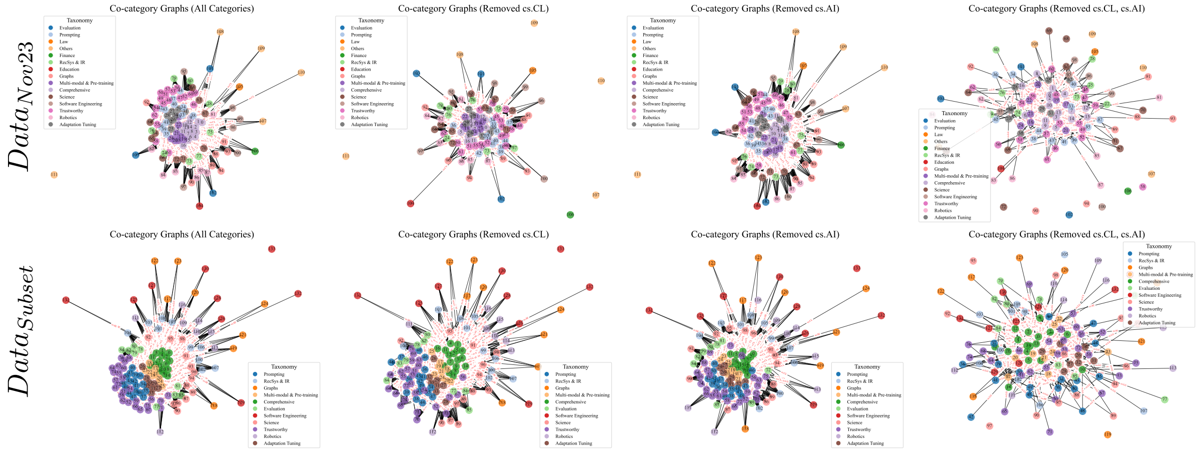


Figure 9: Additional visualization of co-category graphs (extended from Figure 6).

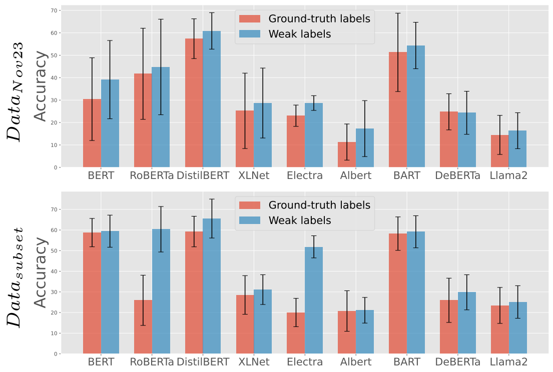


Figure 10: Additional comparison in the fine-tuning paradigm using weak labels (extended from Figure 8).

Language Models	Model Size
BERT (Devlin et al., 2018)	109.49M
RoBERTa (Liu et al., 2019)	124.66M
DistilBERT (Sanh et al., 2019)	66.97M
XLNet (Yang et al., 2019)	117.32M
Electra (Clark et al., 2019)	109.49M
Albert (Lan et al., 2019)	11.70M
BART (Lewis et al., 2020)	140.02M
DeBERTa (He et al., 2020)	139.20M
Llama2 (Touvron et al., 2023)	6.61B

Table 7: Model size of the pre-trained language models. For example, BERT has 109.49 million parameters.

A APPENDIX

In the appendix, we present the GNN and pre-trained language models' hyper-parameters and the hardware and software. We also include the additional comparison results about fine-tuning using weak labels and additional visualization of co-category graphs.

Hyper-parameters and Settings We employ a two-layer GCN (Kipf and Welling, 2016) with 200 hidden units and a ReLU activation function as the backbone GNN to examine the effectiveness of GRL. The GNN is trained by the Adam optimizer with a learning rate, 1×10^{-2} for both co-author graphs and co-category graphs and 2×10^{-2} for text graphs, and converged within 500 training epochs on all subsets. The dropout rate is 0.5.

We fine-tune the full pre-trained language models using the Adam optimizer with a 1×10^{-4} learning rate. We chose the batch size of 8 for the Llama2 and fixed the batch size of 16 for the rest of the models. We implement the pre-trained language models using HuggingFace packages (we choose the base version for all models without adding additional layers) and report the model size in Table 7. All models are tuned with 30 epochs. Note that, during the fine-tuning, we only save the checkpoints with better validation accuracy, i.e., some saved models were only fine-tuned for a small number of epochs.

Hardware and Software The experiment is conducted on a server with the following settings:

- Operating System: Ubuntu 22.04.3 LTS
- CPU: Intel Xeon w5-3433 @ 4.20 GHz
- GPU: NVIDIA RTX A6000 48GB
- Software: Python 3.11, PyTorch 2.1, HuggingFace 4.31, dgl 1.1.2+cu118.

Computational Budgets Based on the above computing infrastructure and settings, computational budgets in our experiments are described as follows. The experiment presented in Table 3 can be reproduced within one hour. The experiment displayed in Table 4 may take 93 hours to complete. Due to limited GPU memory, we implemented Llama2 using the CPU. This consumes around 90 hours in total. The experiment shown in Table 5 (excluding human recognition) can be finished in one hour.

Additional Visualization of Co-category Graphs Besides visualizing four graph structures in $Data_{Jan24}$ in Figure 6, we additionally present the visualization of four corresponding co-category graphs in both $Data_{Nov23}$ and $Data_{subset}$ in Figure 9. The visualization verifies the generalization of GRL across three subsets.

Additional Comparison of Fine-tuning Using Weak Labels Besides the results in Figure 8, we supplement the comparisons on both $Data_{Nov23}$ and $Data_{subset}$ in Figure 10. The comparisons across nice pre-trained language models further validate the effectiveness of fine-tuning using weak labels.

Ethical and Broader Impacts We confirm that we fulfill the author's responsibilities and address the potential ethical issues. In this work, we aim to help researchers quickly and better understand a new research field. Many researchers in academia or industry may potentially benefit from our work.

Statement of Data Privacy Our dataset contains the authors' names in each paper. This information is publicly available so the collection process doesn't infringe on personal privacy.

Disclaimer Regarding Human Subjects Results In Table 5, we include partial results with human subjects. We already obtained approval from the Institutional Review Board (IRB). The protocol number is IRB24-056. We recruited volunteers from a graduate-level course. Before the assessment, we have disclaimed the potential risk (our assessment has no potential risk) and got consent from participants.