Introducing Graph Context into Language Models through Parameter-Efficient Fine-Tuning for Lexical Relation Mining

Jingwen Sun, Zhiyi Tian, Yu He, Jingwei Sun*, and Guangzhong Sun*
University of Science and Technology of China
{ginwind,tianzyi}@mail.ustc.edu.cn
{heyu3761,sunjw,gzsun}@ustc.edu.cn

Abstract

Lexical relation refers to the way words are related within a language. Prior work has demonstrated that pretrained language models (PLMs) can effectively mine lexical relations between word pairs. However, they overlook the potential of graph structures composed of lexical relations, which can be integrated with the semantic knowledge of PLMs. In this work, we propose a parameter-efficient fine-tuning method through graph context, which integrates graph features and semantic representations for lexical relation classification (LRC) and lexical entailment (LE) tasks. Our experiments show that graph features can help PLMs better understand more complex lexical relations, establishing a new state-of-the-art for LRC and LE. Finally, we perform an error analysis, identifying the bottlenecks of language models in lexical relation mining tasks and providing insights for future improvements.

1 Introduction

Lexical relation, a fundamental linguistic concept, refers to how words are related within a language. The task of lexical relation mining (LRM), which seeks to identify and classify the specific lexical relationship between pairs of words, is a challenging problem in natural language processing (NLP). This task holds significant value for downstream applications, including sentiment analysis, ontology annotation, and analogical reasoning.

Our work focuses on lexical relation classification (LRC) and lexical entailment (LE), which are both subtypes of LRM. LRC is formulated as a classification problem, whose relations include various lexical relationships, such as synonymy, antonymy, and hyponymy. LE aims to annotate the importance score of the lexical entailment relationship between two words, which is treated as a regression task.

Previous works (Moskvoretskii et al., 2024; Pitarch et al., 2023; Ushio et al., 2021) have explored the potential of pre-trained language models (PLMs) in solving both LRC and LE, demonstrating significant performance improvements compared to traditional pattern-based methods and distributional models (Yang et al., 2022; Shwartz et al., 2016; Wang et al., 2019). Specifically, these approaches first transform word pairs into specialized sentences. Then, they use parameter-efficient fine-tuning (PEFT) methods to introduce additional training parameters or fully fine-tuned PLM parameters to capture latent lexical relation patterns.

However, neither PEFT nor fully fine-tuned methods account for the influence of graph structures composed of lexical relations. Incorporating graph features, which is distinct from semantic knowledge, enables PLMs to capture knowledge that is seldom explicitly expressed in natural language. Previous work has demonstrated that latent inference patterns within graph structures can significantly aid PLMs in comprehending complex knowledge across various NLP tasks (Fang et al., 2024; Sun et al., 2024). However, since PLMs already contain a vast amount of semantic knowledge, aligning graph-structured knowledge with semantic knowledge in high-dimensional space is also a challenging task (Zhu et al., 2024).

To address this gap, we propose a PEFT method named Efficient Tuning through Graph Context (GET)[1], which captures graph features through a graph neural network (GNN) module and utilizes these graph features as additional context for lexical relation mining. We aim for the model to learn latent lexical relation inference patterns within the graph structure through GNN and align them with the linguistic form of the extra context, thereby providing additional knowledge to the PLMs. The

*Corresponding authors.

[1]Codes and datasets are available at :https://github.com/ginwind/GET

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 10359–10374

experiments demonstrate that the proposed method achieves state-of-the-art performance compared to existing baselines. In addition, we conduct experiments with large-scale PLMs (LLMs) as additional baselines. Supervised experiments show that fine-tuning on small-scale PLMs can be highly competitive, even outperforming LLMs with 10 times the number of parameters. Meanwhile, zero-shot experiments demonstrate that larger LLMs, such as GPT-4o, do not acquire enough lexical relational knowledge.

Our contributions can be concluded as follows:

- We propose a parameter-efficient fine-tuning method, GET, which first constructs a connected lexical graph, then extracts graph features through a relation-sensitive graph neural network module, and finally transforms these graph features into language features understandable by PLMs through multilayer perceptron (MLP) layers, thereby enhancing the ability of PLMs to understand lexical relations.

- The experimental results indicate that graph features can help PLMs better understand lexical relations, achieving the state-of-the-art performance in both LRC and LE tasks. Furthermore, our proposed method, when applied to small-scale PLMs, achieves results comparable to those of LLMs.

- We perform an error analysis to examine the errors made by PLMs in solving LRC and LE tasks, as well as to identify the bottlenecks limiting their performance.

2 Related Works

2.1 Lexical Relation Mining

Early research in lexical relation mining (LRM) focused on specific syntactic patterns in natural language, aiming to extract relations based on frequently occurring syntactic structures or grammar tree paths that could indicate lexical relations (Washio and Kato, 2018; Hearst, 1992). In contexts with limited computational resources and data, such approaches effectively mined clearly stated knowledge relations in text. However, their main drawback is the inability to handle ambiguous natural language expressions, and they require linguistic expertise to aid in modeling.

Subsequently, the introduction of external priors into relation extraction has become a major research trend in this field. Depending on the type of external prior, LRM methods can be classified into two categories: the first category includes distributional model-based relation extraction methods (Shwartz et al., 2016; Yang et al., 2022). These methods outperform traditional pattern-based approaches, but due to the nature of word vectors, they cannot distinguish different relations based on context.

The second category includes PLM-based lexical relation mining methods. PLMs provide context-sensitive knowledge representations for relation extraction. For instance, RelBERT (Ushio et al., 2021) applies the parameter-efficient fine-tuning technique P-Tuning (Liu et al., 2021), arguing that fine-tuning a learnable prompt on a language model can mitigate the negative impact of manually crafted prompts on relation extraction. RelBERT also uses contrastive learning to train a model that can be used for both relation extraction and analogy reasoning. NCGC (Pitarch et al., 2023) argues that language models acquire the ability to recognize relationships between knowledge during pretraining, and the primary factor limiting their performance is the prompt. By fine-tuning the language model with seven different prompts, they achieved optimal performance on LRC and LE. With the growing scale of PLMs, some studies have also explored the potential of LLMs in addressing LRM tasks (Moskvoretskii et al., 2024). However, existing PLM-based research has not explored the impact of graph features on LRM, and no studies have attempted to investigate the boundaries of LLMs' capabilities for LRM.

2.2 Parameter-Efficient Fine-Tuning (PEFT)

PLMs, such as DeBERTa (He et al., 2021) and Llama (Dubey et al., 2024), acquire fundamental knowledge about the real world by understanding large-scale corpora during pretraining. To preserve the knowledge learned by PLMs and improve the efficiency of fine-tuning for downstream tasks, Parameter-Efficient Fine-Tuning (PEFT) methods introduce external trainable parameters to PLMs, keeping the main parameters of PLMs fixed during fine-tuning. For instance, adapter-based methods (Houlsby et al., 2019) insert lightweight neural networks between layers of PLMs as trainable parameters; LoRA (Hu et al., 2022) constructs low-rank matrices during fine-tuning, which are decomposi-

tions of the original matrices in the self-attention modules of PLMs, and updates only these low-rank matrices; and Prompt Tuning (Liu et al., 2021, 2022) optimizes the input text by introducing trainable parameters in the form of soft prompts.

Despite these advancements, integrating PEFT methods with non-Euclidean data, such as graph-structured data, remains an open challenge. Several studies have attempted to combine GNNs with PLMs on textual graphs (Zhu et al., 2024; Ioannidis et al., 2022). However, aligning random initialization GNNs with pretrained PLMs poses significant difficulties. For example, ENGINE (Zhu et al., 2024) introduces GNN-based feature fusion on intermediate outputs of PLMs while keeping the PLM parameters fixed, achieving efficient fine-tuning on specific downstream tasks. Note that these methods primarily focus on using PLMs to solve graph-related problems. To date, constructing external graphs for NLP tasks and using graph features to enhance the performance of PLMs are still being explored.

3 Methodology

3.1 Problem Statement

Let $V = \{w_1, ..., w_n\}$ represent a set of words. A lexical relation r is the relationship that can be treated as a subset of $V \times V$. The set of lexical relations, $R = \{r_1, ..., r_m\}$, is assumed to be *mutually exclusive* and *complete*, meaning that each word pair corresponds to one and only one lexical relation. We define a function $f : V \times V \rightarrow R$ that assigns a specific relation to each word pair. The task of relation mining involves estimating the function f using an approximation $\hat{f}$. This study focuses on two key problems: lexical relation classification (LRC) and lexical entailment (LE). Specifically, the LRC task is formulated as a classification problem, while the LE task is formulated as a regression problem.

3.2 Graph Construction

Given a set of words V and a lexical relation set R, we can naturally construct a graph $G = (V, R)$, where the nodes represent words and the edges represent lexical relations. Specifically, G is a directed graph, with the direction of each edge determined by the corresponding lexical relation. For instance, edges representing the lexical relation *antonymy* are bi-directed, as *antonymy* is symmetric. Additionally, to enrich the lexical information in G, we assign each lexical relation r a detailed description d_r, derived from ConceptNet (Speer et al., 2017), which can be considered a property of the edges.

However, G is often disconnected due to the pattern sparsity (Wang et al., 2021). To capture the structural context based on the local neighborhoods of each connected component, we first apply the PageRank algorithm to G to assess each node's importance. Next, we connect the most important nodes within each weakly connected component by adding an edge e. Notably, e is not an element of R.

3.3 Efficient Tuning through Graph Context (GET)

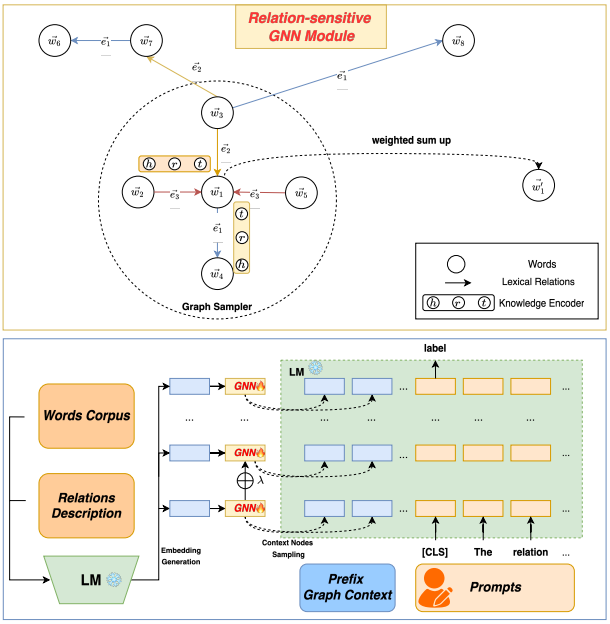

Figure 1: The architecture of proposed method GET. The backbone parameters of the PLM are fixed.

Based on the hypothesis that language models can extract correct knowledge from sufficiently rich contextual information, we propose an efficient tuning method using graph topological features as the language model's context. As depicted in Figure 1, we leverage the external knowledge graph constructed in Section 3.2 as additional knowledge and design a relation-sensitive GNN to capture graph features.

Formally, given a graph $G = (V, R)$ and its lexical relation detailed description d_r, token-level representations of each $w_i \in V$ and lexical relation $r \in R$ will be calculated:

$$H_V^l = PLM\_Layer^l(H_V^{(l-1)}),$$
$$H_R^l = PLM\_Layer^l(H_R^{(l-1)}), \quad (1)$$

where $PLM\_Layer(\cdot)$ denotes the l-th layer of PLM, and $H_V^l \in \mathbb{R}^{|V|\times Q\times D}$, $H_R^l \in \mathbb{R}^{|R|\times Q\times D}$ denotes the token-level representations of each word and lexical relation. Q and D denote the sequence length and the dimensionality of the hidden states, respectively. Especially, H_V^0 is equal to tokens of V and H_R^0 is equal to tokens of d_r.

Subsequently, we design a relation-sensitive graph attention operator to capture graph features from G and H^l:

$$
\begin{aligned}
h_i^{l+1} &= \lambda^l \cdot \sum_{j\in N_i} \alpha_{ij}^l \cdot h_{ij}^l + (1-\lambda^l)\cdot h_i^l, \\
\alpha_{ij}^l &= \frac{exp(e_{ij}^l)}{\sum_{j\in N_i} exp(e_{ij}^l)}, \\
e_{ij}^l &= \sigma(\vec{a}^T[Wh_i^l || Wh_{ij}^l]) \\
h_{ij}^l &= g(h_i^l, h_j^l, h_r^l)
\end{aligned}
\tag{2}
$$

where λ and W are learnable parameters, $g(\cdot)$ is the combination function for a special knowledge triple. h_i^l, h_j^l are the row vectors of H_V^l and h_r^l is the row vector of H_R^l. After applying the graph attention operator, we get a vector representation h_i for each word $w_i \in V$.

Finally, we select the features of top N nodes with the highest PageRank scores in the graph as additional knowledge for the relation extraction prompts. These features are transformed into a language representation via MLP layers and subsequently fed into the language model, along with the prompt. The prompt is constructed using the best performing template in previous work (Pitarch et al., 2023), which transforms the word pair (w_i, w_j) into a specialized sentence.

Sub-graph Sampling. However, the computational cost of the aforementioned algorithm is prohibitively high for large-scale dataset. To reduce the computation, we adopt the following strategies: performing GNN computations only on the subgraph formed by word pairs within the batch and their N-hop neighbors.

Inference The graph constructed in Section 3.2 is composed of words and lexical relations from the training set. To handle the case where words in the test set do not exist in the training set, we introduce a special word, "unknown words," into the set of words V. This special word is treated as an independent node in graph G, and it is connected to other connected components through e. When a new word in the test set does not exist in the training set, it is mapped to the "unknown words" node for inference.

4 Experiments

In this section, we implemented our proposed method on LRC and LE tasks and compare it with previous best-performance baselines. Then, we conducted an ablation study to demonstrate that each component contributes positively to the overall performance. Finally, a sensitivity analysis was performed to examine the impact of different hyperparameters under varying conditions.

4.1 Datasets and Baselines

LRC We adopt five commonly used LRC datasets to evaluate GET: CogALexV (Santus et al., 2016a), BLESS (Baroni and Lenci, 2011), EVALution (Santus et al., 2015), K&H+N (Necsulescu et al., 2015), and ROOT9 (Santus et al., 2016b). Except for EVALution, all datasets contain *random* lexical relation, which means that there is no lexical relation between two words. GET is compared with five baselines in two different categories. The first category is the non-contextual distributional model, including LexNet (Shwartz et al., 2016) and SphereRE (Wang et al., 2019). The second category is fully fine-tuned PLM-based methods, including KEML (Wang et al., 2021), RelBERT (Ushio et al., 2021), and NCGC (Pitarch et al., 2023).

LE We use the HyperLex benchmark (Vulic et al., 2017) for our experiments. This dataset annotates lexical relations between word pairs as well as their graded lexical entailment ratings, which are provided by at least 10 annotators to answer the question: *To what degree is X a type of Y?* HyperLex is divided into train, validation, and test datasets in two configurations: the first is a random split, where word pairs are randomly allocated to the train, validation, and test datasets; the second is a lexical split, which ensures that words in the test dataset do not appear in the train or validation datasets.

Similar to LRC, we compare GET with two types of baselines: the first category includes non-contextual methods using word embeddings, namely LEAR (Vulic and Mrksic, 2018) and HF (Yang et al., 2022); the second category includes context-sensitive PLM-based methods, namely NCGC (Pitarch et al., 2023) and TaxoLlama (Moskvoretskii et al., 2024). The latter is a Llama version trained on WordNet to address lexical problems, and we follow the authors' setting by using a zero-shot approach to evaluate it.

Extra Baselines To facilitate a more compre-

hensive comparison of the proposed methods, we introduce two additional baseline categories. The first category incorporates word embeddings from PLMs and applies traditional GNN-based methods for relationship extraction, including GCN (Kipf and Welling, 2016), GAT (Velickovic et al., 2017), and SAGE (Hamilton et al., 2017). The second category fine-tunes Llama3-8B with LoRA (Hu et al., 2022) both in the sequence classification setting and the instruction setting. Furthermore, we compare GET with LoRA and context-based prompt-tuning-v2 (Liu et al., 2022) on DeBERTa (He et al., 2021).

4.2 Implement Details

To construct the external knowledge graph, we use *NetworkX* to build the original lexical graph and apply the PageRank algorithm along with the weakly connected component algorithm to refine the graph. Subsequently, we obtain token-level representations from various PLMs using *transformers* for both traditional GNN-based methods and our proposed approach. The training module is implemented by *pytorch* and *dgl*. **All templates and instructions used for tuning are provided in Appendix A**.

Our experiments were conducted using an NVIDIA A100 GPU. **The hyperparameters used for training the different datasets are provided in Appendix C**. We employ a grid search to identify the optimal hyperparameters for our proposed method. For each PLM-based method, we train for 10 epochs using a linear growth scheduler followed by cosine decay.

4.3 Performance Analysis

LRC Table 1 presents the experimental results comparing our method with non-contextual baselines and PLM-based baselines. We report the weighted F1-score for LRC. In particular, we remove the RANDOM relation for the CogALex dataset before reporting the results as advised by its authors (Santus et al., 2016a).

The results show that each PLM-based method demonstrates superior performance compared to non-contextual baselines. The results on Llama3 show that the difference between sequence classification and token generation settings has little effect on the outcomes. Besides, DeBERTa-xlarge enhanced with GET achieves the best performance on two datasets. KEML achieved the best performance on the K&+N dataset due to its use of positive and negative examples of metalearning for the existence of the relationship. When comparing the results of NCGC (Roberta-355M), DeBERTa-xlarge (750M), and Llama3-8B, we can conclude that an increase in model parameters provides limited benefits for LRC performance. The complete experimental table can be found in Appendix D.

Furthermore, to investigate the generalization capability of larger LLMs, we conduct zero-shot experiments on the LRC task using GPT-4o, applying the same instruction-tuning prompts as used for Llama-8B-Instruct. However, GPT-4o performs worse than even the least effective supervised models, suggesting that larger model size does not guarantee the acquisition of lexical relational knowledge. We argue that the limitation arises from the data, as the lexical relation knowledge contained in the corpora used for PLM pretraining is not explicitly expressed.

How does graph features perform in non-contextual baselines?

When comparing with non-contextual baselines, we observe that GNN-based methods underperform all baselines. This suggests that directly applying GNNs to the LRC domain does not yield satisfactory results. We hypothesize that, for LRC tasks, topological features do not effectively assist word embeddings in understanding lexical relations between terms. In contrast, SphereRE, which learns embeddings in a three-dimensional space, outperforms GNN-based methods. The suboptimal performance of GNN-based methods on CogALexV can be primarily attributed to their tendency to overfit the RANDOM relation category while we do not account for this relation in the evaluation metrics.

How does the performance of GET compare with other PLM-based approaches?

We further analyze that incorporating graph features as contextual inputs to language models positively impacts LRC tasks. This is particularly evident when comparing with PT2, which also uses contextual inputs as additional trainable parameters. This indicates that the latent reasoning patterns within graph topological features can enhance the PLM's ability to generalize lexical relations. Transforming graph topological features into textual contexts for language models shows greater potential for LRC tasks compared to directly using GNNs to obtain word embeddings.

LE Table 2 presents the experimental results comparing our proposed methods with baselines using the Spearman ρ correlation metric, which evaluates

Methods	BLESS	CogALexV	EVALution	K&H+N	ROOT09
GCN	0.508	0.045	0.240	0.911	0.591
GAT	0.485	0.024	0.427	0.912	0.642
SAGE	0.845	0.148	0.538	0.964	0.726
LexNET	0.893	0.445	0.600	0.985	0.813
SphereRE	0.938	0.471	0.620	0.990	0.861
KEML	0.944	0.500	0.660	**0.993**	0.878
RelBERT	0.921	0.664	0.701	0.949	0.910
NCGC	0.956	0.762	0.771	0.989	0.937
+ Verbalizer	0.951	0.756	0.746	0.985	0.926
Llama3-8B	0.953	**0.790**	0.772	0.989	<u>0.947</u>
Llama3-8B-Instruct	**0.963**	<u>0.777</u>	0.756	<u>0.991</u>	0.945
DeBERTa-xlarge*	0.957	0.761	0.784	0.987	0.945
+ LoRA	0.955	0.757	<u>0.799</u>	0.988	0.937
+ PT2	0.951	0.669	0.775	0.987	0.938
+ GET	<u>0.959</u>	0.765	**0.805**	0.988	**0.954**
GPT-4o	0.234	0.081	0.202	0.397	0.368

Table 1: The weighted F1-score of the LRC experimental results is reported. + Verbalizer refers to the application of a manual verbalizer, which transforms labels into tokens to formulate the token prediction problem. Llama3-8B is trained using a sequence classification setting, while Llama3-8B-Instruct is trained with instruction tuning. * means that the language model is fully fine-tuned. We use **boldface** and <u>underlining</u> to denote the best and the second-best performance, respectively.

Methods	lexical	random
GCN	—	0.391
GAT	—	0.119
SAGE	—	0.242
LEAR	0.174	0.686
HF	—	0.690
NCGC	0.755	0.774
+ Verbalizer	0.794	0.828
Llama3-8B	0.873	0.905
TaxoLlama	0.702	0.593
DeBERTa-xlarge*	0.881	0.898
+ LoRA	0.876	0.904
+ PT2	0.864	0.882
+ GET	0.887	0.901
IAA	0.864	

Table 2: Results for HyperLex dataset. We report the Spearman ρ correlation for both *lexical* and *random* settings.

how well the relationship between the model's predicted regression values and the median of human-annotated values can be described using a monotonic function. The authors of HyperLex provide the Inter-Annotator Agreement (IAA), calculated as the average Spearman ρ correlations of an annotator with the average of all other raters. The IAA represents the level of consistency among annotators when grading lexical entailment ratings.

Similar to LRC, PLM-based methods generally outperform non-contextual methods, Except for TaxoLlama, which was tested in a zero-shot setting, all other fully trained PLM-based methods exhibit superior results. Our experiments on Llama3-8B and DeBERTa-xlarge achieve performance exceeding the IAA. The graph features introduced by GET have a positive impact on the model's performance. Compared to other fine-tuning methods, DeBERTa fine-tuned with GET achieved the best performance on the lexical setting and the second-best performance on the random setting.

How does graph features perform in non-contextual baselines?

When comparing non-contextual methods, we observe significant limitations. For instance, GNN-based baselines are ineffective in the lexical setting because the test dataset's words are not modeled as graph nodes, making it impossible for the model to distinguish different test dataset words. Addition-

ally, LEAR, which can operate in the lexical setting, also performs poorly. Similar to the LRC task, GNN-based baselines fail to capture clear graded lexical entailment features, resulting in poor performance even when directly applied to the random setting.

What are the advantages of PLMs in solving the LE task?

We hypothesize that the LE task can be understood as determining whether a word can be substituted by another in a given context, which aligns closely with the masked language model and causal language model objectives during PLM pretraining. This allows the model to learn substitution relationships from the corpus, which can then be applied to the LE task. Due to page limitations, the analysis of performance bottlenecks in prior works and the analysis of results surpassing IAA are provided in Appendix E.

4.4 Ablation Study

In this section, we design two variants to evaluate the impact of components in the proposed method: (1) w/o PLM: utilizing the node embeddings obtained directly from Equation 2 as the final representation; (2) w/o λ: removing the learnable parameter λ from Equation 2. It should be noted that the removal of GNN components can be represented by PT2 or fully fine-tuned models, which will not be discussed in detail here.

As demonstrated in Table 3, results show that each component contributes positively to the overall performance. Particularly, when removing the PLM component (w/o PLM), the model degenerates into a conventional GNN architecture, which exhibits several inherent limitations for LRM. Performance degradation in variant w/o λ can be attributed to disrupted information flow between different PLM layers, resulting in inferior performance compared to our proposed method.

4.5 Sensitivity Analysis

The Dimension of GNN. Figure 2 presents the results. Since the hidden size of the PLMs we use is 1024, setting the GNN dimension too low limits the model's capacity to capture semantic and graph features effectively. On the other hand, a large GNN dimension may lead to model degradation (Zhang et al., 2021), restricting its ability to integrate graph features into contextual representations. For our experimental datasets, an appropriately balanced GNN dimension achieves the best performance.

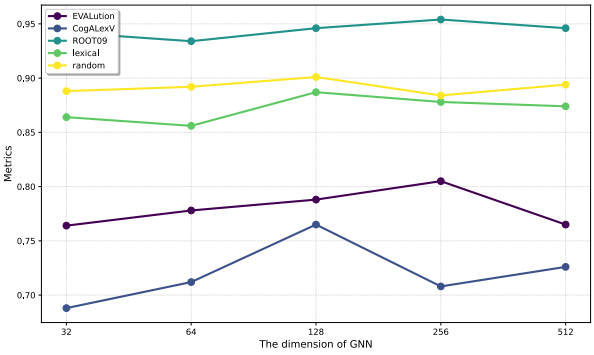

Figure 2: The impact of the dimension of GNN.

The Number of GNN Layers. Figure 3 illustrates that the optimal number of GNN layers varies across different datasets. We hypothesize that this variation is related to the complexity of the datasets. Since the number of GNN layers determines how many hops of neighboring nodes can be accessed in a forward pass, more complex datasets, such as CogALexV and EVALution, require a greater number of GNN layers to effectively capture graph features. In contrast, for simpler datasets like HyperLex and ROOT09, a single-layer GNN is sufficient to achieve strong performance.

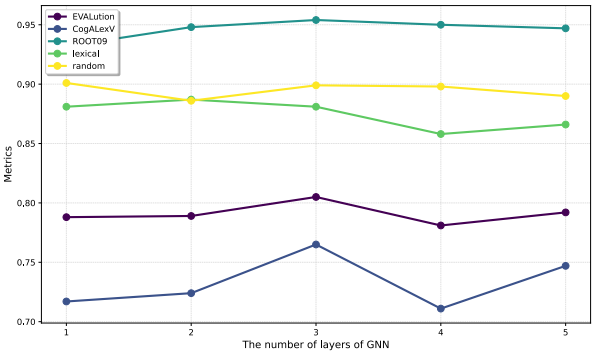

Figure 3: The impact of the number of GNN layers.

5 Error Analysis

This section presents a comprehensive error analysis of model predictions and investigate the underlying causes of these errors.

LRC The experimental results in Table 1 reveal that both CogALexV and EVALution datasets exhibit inferior performance compared to other datasets. Since CogALexV is a subset of EVALution, we focus our analysis on EVALution for deeper insights. Figure 4 presents the confusion matrices of the top four performing methods in the EVALution dataset. All models demonstrate

Methods	LRC					LE	
	BLESS	CogALexV	EVALution	K&H+N	ROOT09	lexical	random
DeBERTa-xlarge (+GET)	**0.959**	**0.765**	**0.805**	**0.988**	**0.954**	**0.887**	**0.901**
w/o PLM	0.711	0.207	0.524	0.924	0.682	—	0.207
w/o λ	0.945	0.749	0.742	0.986	0.944	0.864	0.897

Table 3: Experimental results of ablation study.

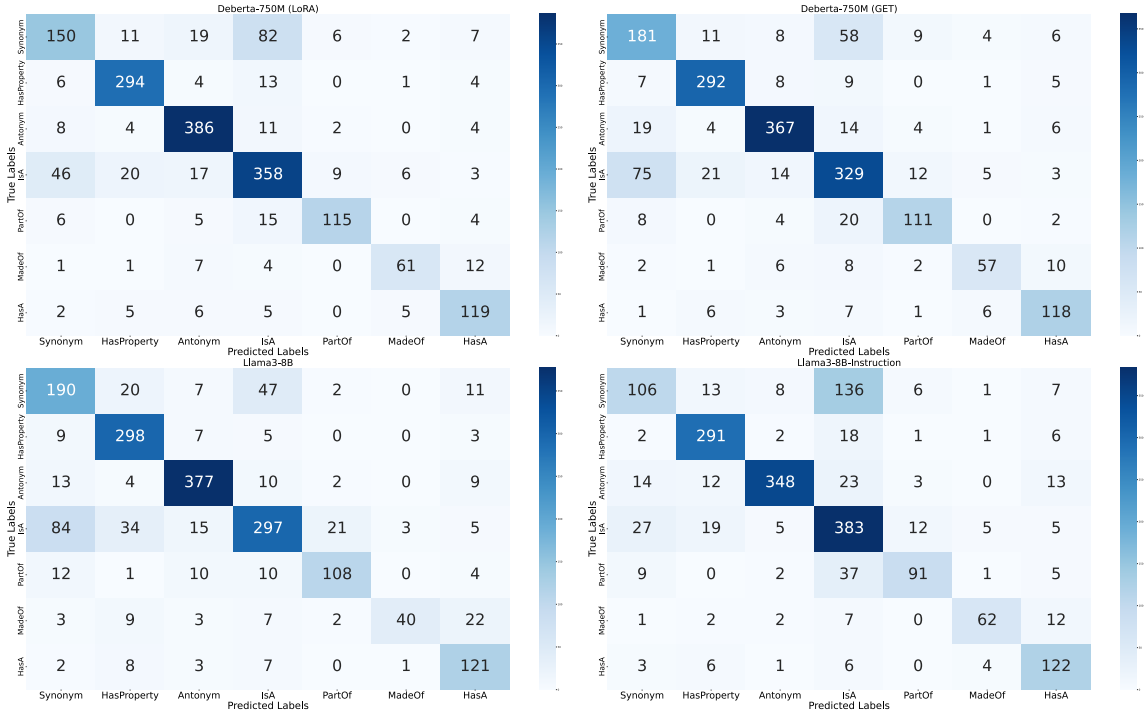

Figure 4: Confusion matices of EVALution.

significant challenges in distinguishing between Synonym and IsA categories, frequently misclassifying Synonyms as IsA and vice versa. The χ^2 test confirms a statistically significant relationship between prediction accuracy and category type. And there is a great difference in the predictions for Synonym, as detailed in Appendix F. This phenomenon is most evident in Llama3-8B-Instruction, which misclassifies over half of Synonym pairs as IsA. In contrast, LoRA and GET exhibit the lowest misclassification rates between Synonym and IsA, contributing to their superior performance. While previous work (Pitarch et al., 2023) identified Synonym and Antonym as the most challenging categories, our proposed method successfully mitigates this specific issue.

Through analysis of misclassified Synonym and IsA instances, we observe that certain annotations in the dataset exhibit inherent ambiguity. For instance, word pairs like *(jacket, coat)* and *(study, learn)* are labeled as IsA but predicted as Synonym by our method, while *(dot, point)* and *(create, make)*, annotated as Synonym, are classified as IsA. These cases may not represent errors but rather reflect the nuanced semantic relationships between words, where both classifications could be considered valid. This observation suggests that reformulating the LRC benchmark as a multi-label classification task could potentially address such ambiguities and provide more nuanced evaluation of lexical relations.

LE Figure 5 illustrates the scatter plot of standard deviations and squared residuals for each word pair in the random and lexical test sets of HyperLex. The standard deviation is calculated based on annotator-provided labels, while the squared residuals are derived from the differences between the model's predicted values and the mean value of annotations. It is evident that the standard deviation is positively correlated with the squared residuals,

meaning that word pairs with higher annotator disagreement tend to have larger prediction errors. We hypothesize that for word pairs with high standard deviation, the mean of the ten annotated values still carries some bias. Introducing lexical entailment (LE) into specific semantic contexts and formulating it as a multi-label classification task may help mitigate this issue.

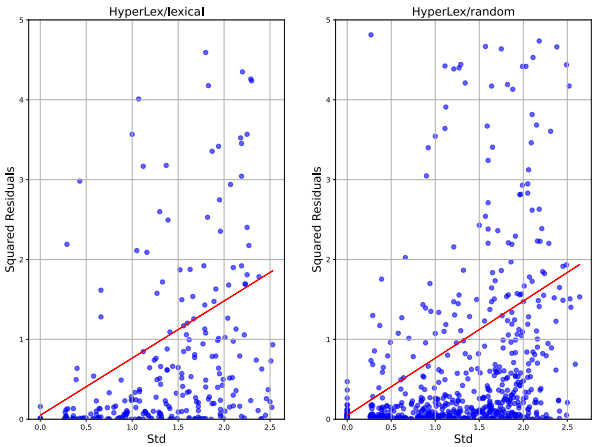

Figure 5: Standard deviation and squared residuals. The red line represents the linear regression line.

6 Conclusion

In this paper, we propose a parameter-efficient fine-tuning method named Efficient Tuning through Graph Context (GET), which integrates graph features and semantic representations via a relation-sensitive graph neural network module. We apply this method to the tasks of lexical relation classification (LRC) and lexical entailment (LE). Experiments show that incorporating graph features has a positive impact for PLMs on both LRC and LE tasks. Finally, we conduct an Error Analysis to explore the performance limitations of language models in LRM, and suggest constructing LRC and LE benchmarks as multi-label classification tasks to avoid potential ambiguities.

Limitations

We find that the main limitations of our work are as following:

- Although our method has shown success on small-scale PLMs, its application to LLMs has not yielded satisfactory results.The main reason, as we analyze, is that after GNN captures the latent lexical relation knowledge from the graph structure, it struggles to map it into a language-based context that LLMs can understand. This issue arises due to the larger hidden state in LLMs, where feature learning becomes increasingly difficult, and convergence is even harder to achieve. A detailed analysis can be found in the Appendix G.

- We analyzed the bottlenecks faced by PLMs in the LRC and LE tasks, as well as the potential ambiguities that might arise from the current problem definitions. We believe that framing the LRC and LE tasks as multi-label classification problems could help mitigate the ambiguity issues. However, we did not construct any lexical relation multi-label classification benchmarks in this work.

- The LRC and LE datasets we use have certain limitations, primarily in two aspects: (1) As analyzed in Section 5, in the LRC task, the same word pair can be associated with multiple lexical relations. However, due to the constraints of multi-class tasks, the model is forced to select only one lexical relation as the label. This leads to the model learning the biases present in the annotator's labeling process. (2) In the EVALution dataset (Santus et al., 2015), there are word pairs that are identical but labeled with different lexical relations. Similarly, the HyperLex (Vulic et al., 2017) dataset also contains word pairs with the same wording but different label values. Upon analyzing the source code of prior works, we found that they did not address these noisy data, and we have continued with their settings. However, the existence of these noisy data points can negatively impact the model's performance.

- Our experiments focused solely on the LRC and LE tasks and did not encompass all lexical relation-related tasks.

- Our experiments were conducted only for the English language.

Ethical Statement

This work does not pose any ethical issues. In the writing of this paper, we used ChatGPT and DeepSeek for language translation and grammar checking. The prompts provided to these tools were specifically designed to meet these two objectives.

Acknowledgements

This research was supported by the 2023 Top-Notch Student Training Program 2.0 for Basic Disciplines (20231008).

References

Marco Baroni and Alessandro Lenci. 2011. How we blessed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics, Edinburgh, UK, July 31, 2011*, pages 1–10. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Jinyuan Fang, Zaiqiao Meng, and Craig MacDonald. 2024. REANO: optimising retrieval-augmented reader models through knowledge graph generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 2094–2112. Association for Computational Linguistics.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1024–1034.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: decoding-enhanced bert with disentangled attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *14th International Conference on Computational Linguistics, COLING 1992, Nantes, France, August 23-28, 1992*, pages 539–545.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Vassilis N. Ioannidis, Xiang Song, Da Zheng, Houyu Zhang, Jun Ma, Yi Xu, Belinda Zeng, Trishul Chilimbi, and George Karypis. 2022. Efficient and effective training of language and graph neural network models. *CoRR*, abs/2206.10781.

Thomas N. Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 61–68. Association for Computational Linguistics.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT understands, too. *CoRR*, abs/2103.10385.

Viktor Moskvoretskii, Ekaterina Neminova, Alina Lobanova, Alexander Panchenko, and Irina Nikishina. 2024. Taxollama: Wordnet-based model for solving multiple lexical semantic tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 2331–2350. Association for Computational Linguistics.

Silvia Necsulescu, Sara Mendes, David Jurgens, Núria Bel, and Roberto Navigli. 2015. Reading between the lines: Overcoming data sparsity for accurate classification of lexical relationships. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics, *SEM 2015, June 4-5, 2015, Denver, Colorado, USA*, pages 182–192. The *SEM 2015 Organizing Committee.

Lucia Pitarch, Jordi Bernad, Lacramioara Dranca, Carlos Bobed Lisbona, and Jorge Gracia. 2023. No clues good clues: out of context lexical relation classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 5607–5625. Association for Computational Linguistics.

Enrico Santus, Anna Gladkova, Stefan Evert, and Alessandro Lenci. 2016a. The cogalex-v shared task on the corpus-based identification of semantic relations. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon, CogALex@COLING 2016, Osaka, Japan, December 12, 2016*, pages 69–79. The COLING 2016 Organizing Committee.

Enrico Santus, Alessandro Lenci, Tin-Shing Chiu, Qin Lu, and Chu-Ren Huang. 2016b. Nine features in a random forest to learn taxonomical semantic relations. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

Enrico Santus, Frances Yung, Alessandro Lenci, and Chu-Ren Huang. 2015. Evalution 1.0: an evolving semantic dataset for training and evaluation of distributional semantic models. In *Proceedings of the 4th Workshop on Linked Data in Linguistics: Resources and Applications, LDL@IJCNLP 2015, Beijing, China, July 31, 2015*, pages 64–69. Association for Computational Linguistics.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4444–4451. AAAI Press.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Asahi Ushio, Jose Camacho-Collados, and Steven Schockaert. 2021. Distilling relation embeddings from pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9044–9062, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph attention networks. *CoRR*, abs/1710.10903.

Ivan Vulic, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. 2017. Hyperlex: A large-scale evaluation of graded lexical entailment. *Comput. Linguistics*, 43(4).

Ivan Vulic and Nikola Mrksic. 2018. Specialising word vectors for lexical entailment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1134–1145. Association for Computational Linguistics.

Chengyu Wang, Xiaofeng He, and Aoying Zhou. 2019. Spherere: Distinguishing lexical relations with hyperspherical relation embeddings. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1727–1737. Association for Computational Linguistics.

Chengyu Wang, Minghui Qiu, Jun Huang, and Xiaofeng He. 2021. KEML: A knowledge-enriched meta-learning framework for lexical relation classification. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13924–13932. AAAI Press.

Koki Washio and Tsuneaki Kato. 2018. Filling missing paths: Modeling co-occurrences of word pairs and dependency paths for recognizing lexical semantic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1123–1133. Association for Computational Linguistics.

Dongqiang Yang, Ning Li, Li Zou, and Hongwei Ma. 2022. Lexical semantics enhanced neural word embeddings. *Knowl. Based Syst.*, 252:109298.

Wentao Zhang, Zeang Sheng, Yuezihan Jiang, Yikuan Xia, Jun Gao, Zhi Yang, and Bin Cui. 2021. Evaluating deep graph neural networks. *CoRR*, abs/2108.00955.

Yun Zhu, Yaoke Wang, Haizhou Shi, and Siliang Tang. 2024. Efficient tuning and inference for large language models on textual graphs. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pages 5734–5742. ijcai.org.

A Templates and Instructions

Templates for sequence classifiction According to prior works, we select the best-performing template to transform a specific word pair into a prompt. For the sequence classification setting in our experiments, we use the following template:

```
Today, I finally discovered the relation
between <A> and <B>.
```

<A> and <B> represent the head and tail words of a word pair, respectively. We then use the special token of different PLMs to classify the prompt. For Deberta, we select the <CLS> token. For Llama, following its classification setting similar to GPT, we choose the rightmost token (excluding the <PAD> token) as the classification token.

Instructions for Instruction-Tuning For Llama3-8B-Instruct, we formulate the LRC task as a token generation task and employ a causal language model (LM) to solve LRC. To prompt the model to produce human-comprehensible outputs, we use the following instruction:

```
<SYSTEM>:
You are a linguistics expert. Please
give the semantic relationship between
the following two words A and B. You
can only answer with these few relations:
<R>.
Here are the descriptions of each
relation.
<r_1>: <d_r1>
<r_2>: <d_r2>
...
<r_m>: <d_rm>
<USER>:
A:<w_i>, B:<w_j>
<SYSTEM>:
Answers:
```

<SYSTEM> and <USER> tokens are replaced by the built-in chat template of the model. <R> represents specific lexical relations such as HasA and IsA. Unlike previous works, the causal LM does not require a verbalizer to construct lexical relations. Each lexical relation r_i and its corresponding description d_{r_i} are obtained from ConceptNet[2]. Finally, during tuning, we batch every 10 examples into a single instruction to accelerate training.

[2]https://github.com/commonsense/conceptnet5/wiki/relations

Dataset	train	validation	test
BLESS	18582	1327	6637
CogALexV	3054	-	4260
EVALution	5160	372	1846
K&H+N	40256	2876	14377
ROOT09	8933	638	3191
HyperLex (lexical)	1133	85	269
HyperLex (random)	1831	130	655

Table 4: Datasets statistics: Number of pairs for each dataset in the train/validation/test splits.

B Datasets Description

For the LRC datasets[3], except for the K&H+N dataset, the other four datasets used for LRC are, to some extent, extensions and modified versions of the BLESS dataset. The BLESS dataset was designed to study analogy reasoning through distributional models and additionally incorporates WordNet and ConceptNet as supplementary data sources, with random relations from crowdsourcing adding noise. The EVALution dataset is an extension of BLESS, introducing synonymy and antonymy relations and adding domain-independent linguistic data. CogALexV is a challenging subset of the EVALution dataset provided at the 2016 ACL workshop on lexical relation classification. On the other hand, the K&H+N dataset was derived from hierarchical relationships in WordNet, specifically focusing on word relations in the fields of animals, plants, and transportation. All of the above datasets avoid the use of multi-word phrases during construction.

For the LE datasets[4], this dataset annotates lexical relations between word pairs along with their graded lexical entailment ratings, which are provided by at least 10 annotators to answer the question: To what degree is X a type of Y? Although the dataset provides additional lexical relations, they are not utilized in our method. Table 4 shows the dataset statistics.

C Hyperparameters and Detailed Experimental setup

For DeBERTa, the hyperparameters of GET are shown in Table 5. In addition to the hyperparameters listed in the table, the remaining settings are as follows: the initial learning rate is set to 1e-4, with

[3]https://huggingface.co/datasets/relbert/lexical_relation_classification

[4]https://github.com/cambridgeltl/hyperlex

a maximum learning rate of 5e-4. For the ROOT09 dataset, the warm-up rate is 0.3, while for all other datasets, it is set to 0.2.

For fully fine-tuned models, we use an initial learning rate of 5e-6 and a maximum learning rate of 2e-5. For LoRA, we set the LoRA alpha and LoRA rank to 16 and 64, respectively, with a LoRA dropout rate of 0.1. For PT2, we use 20 prefix tokens. The learning rate settings for LoRA and PT2 are the same as those for GET. All DeBERTa models are trained with a batch size of 32.

For Llama, the sequence classification setting follows the same configuration as LoRA, with a batch size of 8. For instruction tuning, the batch size is set to 2, and every 10 examples are merged into a single prompt, requiring the model to learn and predict the lexical relations for all ten examples in one forward pass.

All methods above are trained for 10 epochs using a scheduler with linear growth followed by cosine decay.

Datasets	dim	layers	context nodes
BLESS	128	3	12
CogALexV	128	4	20
EVALution	256	3	20
K&H+N	128	3	20
ROOT09	256	3	20
lexical	128	2	20
random	128	1	20

Table 5: Hyperparameters of GET. dim indicates the dimension of GNN. layers indicates the number of layers of GNN. context nodes indicates the number of context nodes used for prompting as the graph context.

For GNN-based methods, since GNNs are not pre-trained, we train them for 30 epochs. For SGAE, we use an LSTM to embed the vectors of adjacent nodes.

All our experiments are conducted using the set_seed function from the *transformers* library, with the seed set to 42. The reported results correspond to a single run with this seed. For datasets with a validation set, we perform hyperparameter search using the validation set to determine the optimal hyperparameters, and the validation set results corresponding to the reported experimental outcomes can be found in Table 6, 7. The training time for all models is approximately 40 hours, with GNN-based methods taking around 10 hours and the Llama3-8B model requiring an additional 10 hours.

D Complete Results

The complete experimental results for LRC on test datasets can be found in Table 8. Similar to the results presented in the main text, after incorporating weighted precision and weighted recall, the best performance is still achieved by DeBERTa and Llama3-8B. Notably, some of our experiments on DeBERTa-large (390M) even outperform DeBERTa-xlarge (750M) and Llama3-8B. We attribute this to the limitations in the lexical relation knowledge contained in the training corpora of PLMs, suggesting that model size is not the primary constraint on performance.

Moreover, the proposed PEFT method, GET, which incorporates graph context, demonstrates a clear advantage over PT2, which constructs prefix tokens using MLP layers. However, the experimental results of GET are highly similar to those of LoRA and full fine-tuning (FT). This indicates that for LRM tasks, semantic knowledge plays a dominant role, while graph features provide only a limited positive impact on lexical relation mining.

E Detailed Performance Analysis

An analysis of performance bottlenecks in prior works.

In addition, previous work consider the lexical relation types in HyperLex when modeling regression tasks, which contributes to a better understanding of the lexical relations but results in a lower Spearman ρ correlation. For example, NCGC first classifies lexical relations and then trains an additional regression module to fit the LE task. In addition, PEFT methods allow training without altering the PLM's original parameters, enabling us to directly regress on the PLM's hidden states, bypassing the LRC component. As a result, we achieve a performance improvement of over 5 points compared to prior works, even surpassing the IAA.

An analysis of the results surpassing IAA.

The results surpassing IAA indicates that the model's predicted scores are closer to the median of the annotated scores. If the LE task is viewed purely as a regression task, our experiments achieve the best performance. Furthermore, we observe that different PEFT methods show minimal differences in the LE task, and the graph features used in GET do not have as consistently positive an im-

Methods	BLESS	EVALution	K&H+N	ROOT09
GCN	0.504	0.275	0.908	0.596
GAT	0.493	0.484	0.909	0.646
SAGE	0.847	0.581	0.965	0.731
Llama3-8B	0.958	0.820	0.989	**0.959**
Llama3-8B-Instruct	0.956	0.776	**0.992**	0.957
DeBERTa-xlarge*	0.960	0.796	0.989	0.953
+ LoRA	0.959	**0.827**	0.990	0.956
+ PT2	0.954	0.819	0.988	0.949
+ GET	**0.961**	0.818	0.990	0.947

Table 6: The weighted F1-score of the LRC experimental results on validation datasets.

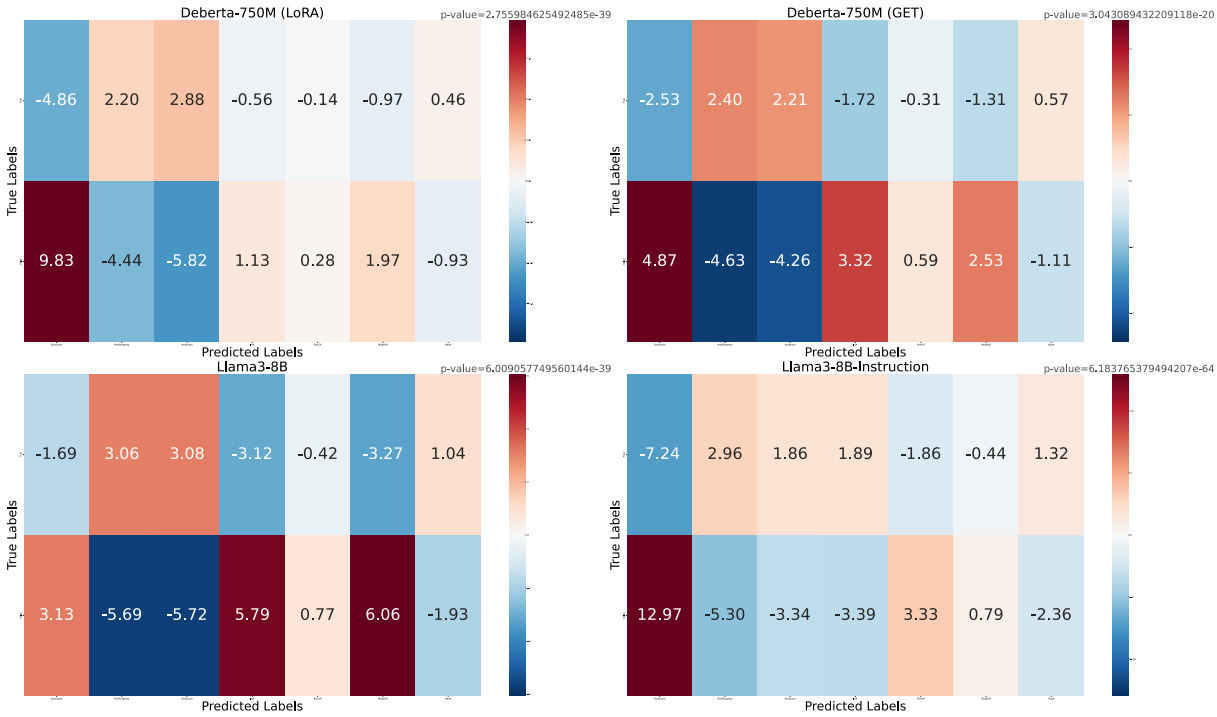

Figure 6: Pearson Residuals of EVALution.

Methods	lexical	random
GCN	—	0.378
GAT	—	0.212
SAGE	—	0.301
Llama3-8B	0.903	0.920
DeBERTa-xlarge*	0.925	0.921
+ LoRA	0.908	0.931
+ PT2	0.883	0.903
+ GET	0.913	0.922
IAA	0.864	

Table 7: Results for HyperLex validation dataset.

pact as they do in LRC. Therefore, we conclude that LE, compared to LRC, is a task more reliant on semantic knowledge. Increasing the volume of pre-training data may help models achieve better performance, while graph features provide limited assistance in LE tasks.

Through detailed experimental analysis, we identify two potential reasons for this: (1) GET can infer some latent patterns from the graph, thereby enhancing performance in LRC and LE; (2) small-scale PLMs already store sufficient lexical relation knowledge from massive corpora. We also perform a sensitivity analysis to examine the effects of different hyperparameters on various datasets.

F Detailed Error Analysis

Figure 6 presents the results of the χ^2 test conducted on the transformed confusion matrix from Figure 4, where the x-axis represents lexical relations and the y-axis indicates prediction correctness. Our null hypothesis posits that prediction accuracy is independent of lexical relation categories. The obtained p-values are consistently significant ($p << 0.01$), leading us to reject the null hypothesis and conclude that prediction accuracy is significantly associated with specific lexical relations.

To further quantify the nature of these associations, we employ Pearson residuals to measure the discrepancy between observed and expected frequencies of correct predictions across different lexical relations. The analysis reveals distinct patterns across models: for DeBERTa, the largest discrepancies are observed in predicting Synonym and Antonym relations, while for Llama, the most pronounced differences occur in classifying Synonym and IsA relations. These findings suggest model-specific patterns in handling different types of lexical relationships, potentially reflecting inherent architectural biases or limitations in capturing specific semantic nuances.

G Apply GET on LLMs

In the main body of the text, the proposed method GET was not applied to LLMs in the LRE experiments for the following reasons: (1) The experiments demonstrated that the LRE results on small-scale PLMs could match or even surpass those on LLMs, suggesting that increasing the scale of LLMs does not significantly improve LRE performance. We hypothesize that the implicit knowledge of lexical relations is already stored within the parameters of smaller PLMs, and therefore, does not require a larger model size. (2) Applying context-based PEFT methods, such as PT, PT2, and GET, to LLMs presents inherent disadvantages, as explained below:

The cost and benefit of applying context-based PEFT methods to LLMs do not align proportionally. This is because mapping a neural network to a context and prompt that LLMs can understand is challenging due to the large parameter sizes of LLMs. Smaller networks are unable to accomplish this task, while larger networks lead to a significant increase in training costs, contradicting the original goal of PEFT methods to reduce computational overhead.

We attempted to use the best hyperparameter settings from the DeBERTa experiments on Llama3-8B-Instruction, but the model's outputs were essentially garbled. We suspect that this is due to PEFT methods (PT2, GET) failing to generate a context form that LLMs can comprehend, which, in turn, interferes with the LLM's inherent causal LM capabilities. Furthermore, experiments on Llama3-8B had difficulty converging. Increasing the GNN dimension and the number of GNN layers yielded some improvements, but the associated training cost was prohibitive. For these reasons, we did not extend GET to LLMs.

Methods	BLESS			CogALexV			EVALution		
	pre	rec	F1	pre	rec	F1	pre	rec	F1
GCN	0.518	0.598	0.508	0.051	0.038	0.045	0.269	0.314	0.240
GAT	0.471	0.561	0.485	0.049	0.075	0.024	0.443	0.433	0.427
SAGE	0.849	0.846	0.845	0.173	0.140	0.148	0.550	0.544	0.538
LexNET	0.894	0.893	0.893	—	—	0.445	0.601	0.607	0.600
SphereRE	0.938	0.938	0.938	—	—	0.471	0.860	0.862	0.861
KEML	0.944	0.943	0.944	—	—	0.500	0.663	0.660	0.660
RelBERT	—	—	0.921	—	—	0.664	—	—	0.701
NCGC	0.956	0.955	0.956	—	—	0.762	0.773	0.771	0.771
+ Verbalizer	0.951	0.950	0.951	—	—	0.756	0.774	0.754	0.746
Llama3-8B	0.955	0.953	0.953	0.792	0.791	**0.790**	0.782	0.775	0.772
Llama3-8B-Instruct	**0.963**	**0.963**	**0.963**	0.784	0.768	0.777	0.772	0.763	0.756
DeBERTa-xlarge*	0.957	0.957	0.957	0.760	0.768	0.761	0.791	0.784	0.784
+ LoRA	0.955	0.955	0.955	0.782	0.732	0.757	0.800	0.803	0.799
+ PT2	0.954	0.951	0.951	0.714	0.645	0.669	0.776	0.775	0.775
+ GET	0.959	0.958	0.959	0.771	0.769	0.765	**0.806**	**0.804**	**0.805**
DeBERTa-large*	0.955	0.953	0.954	0.738	**0.797**	0.761	0.789	0.785	0.786
+ LoRA	0.955	0.953	0.954	**0.803**	0.722	0.743	0.792	0.795	0.793
+ PT2	0.956	0.954	0.955	0.647	0.662	0.654	0.588	0.599	0.584
+ GET	0.958	0.958	0.958	0.760	0.740	0.744	0.783	0.785	0.784

Methods	K&H+N			ROOT09		
	pre	rec	F1	pre	rec	F1
GCN	0.911	0.924	0.911	0.615	0.584	0.591
GAT	0.916	0.920	0.912	0.643	0.640	0.642
SAGE	0.966	0.968	0.964	0.734	0.720	0.726
LexNET	0.985	0.986	0.985	0.813	0.814	0.813
SphereRE	0.990	0.989	0.990	0.860	0.862	0.861
KEML	**0.993**	**0.993**	**0.993**	0.878	0.877	0.878
RelBERT	—	—	0.949	—	—	0.910
NCGC	0.989	0.989	0.989	0.938	0.937	0.937
+ Verbalizer	0.986	0.986	0.985	0.926	0.926	0.926
Llama3-8B	0.989	0.989	0.989	0.946	0.947	0.947
Llama3-8B-Instruct	0.991	0.991	0.991	0.946	0.945	0.945
DeBERTa-xlarge*	0.987	0.987	0.987	0.945	0.945	0.945
+ LoRA	0.988	0.988	0.988	0.937	0.937	0.937
+ PT2	0.988	0.987	0.987	0.939	0.938	0.938
+ GET	0.988	0.988	0.988	**0.955**	**0.954**	**0.954**
DeBERTa-large*	0.989	0.989	0.989	0.943	0.943	0.943
+ LoRA	0.987	0.987	0.987	0.943	0.943	0.943
+ PT2	0.987	0.987	0.987	0.944	0.947	0.944
+ GET	0.987	0.987	0.987	0.941	0.939	0.940

Table 8: Complete results of LRC. + Verbalizer refers to the application of a manual verbalizer, which transforms labels into tokens to formulate the token prediction problem. Llama3-8B is trained using a sequence classification setting, while Llama3-8B-Instruct is trained with instruction tuning. * means that the language model is fully fine-tuned. We use **boldface** and underlining to denote the best and the second-best performance, respectively.