AMR-RE: Abstract Meaning Representations for Retrieval-Based In-Context Learning in Relation Extraction

Peitao Han[1], Lis Kanashiro Pereira[2], Fei Cheng[3], Wan Jou She[4], Eiji Aramaki[1]
[1] Nara Institute of Science and Technology, Japan
[2] National Institute of Information and Communications Technology (NICT), Japan
[3] Kyoto University, Japan
[4] Kyoto Institute of Technology, Japan
han.peitao.hr3@is.naist.jp, liskanashiro@nict.go.jp, feicheng@i.kyoto-u.ac.jp
wjs2004@kit.ac.jp, aramaki@is.naist.jp

Abstract

Existing in-context learning (ICL) methods for relation extraction (RE) often prioritize language similarity over structural similarity, which may result in overlooking entity relationships. We propose an AMR-enhanced retrieval-based ICL method for RE to address this issue. Our model retrieves in-context examples based on semantic structure similarity between task inputs and training samples. We conducted experiments in the *Supervised* setting on four standard English RE datasets. The results show that our method achieves state-of-the-art performance on three datasets and competitive results on the fourth. Furthermore, our method outperforms baselines by a large margin across all datasets in the more demanding *Unsupervised* setting.

1 Introduction

Large language models (LLMs) exhibit strong in-context learning (ICL) abilities across various NLP tasks simply by being given a few examples of the task. However, the quality of few-shot demonstrations can substantially impact the performance of ICL, and tasks requiring high precision, such as relation extraction, remain challenging.

Relation extraction (RE) is a task to identify a predefined semantic relation between entity pairs mentioned in the context. Relations between entity pairs are often implicitly expressed, which can lead to suboptimal ICL performance. Existing ICL methods for RE often overlook the semantic associations between entity pairs, relying primarily on entity mentions or overall sentence semantics for representation (Han et al., 2023; Wan et al., 2023; Li et al., 2024; Ma et al., 2023; Sun et al., 2023).

Abstract Meaning Representation (AMR) (Banarescu et al., 2013) provides a detailed semantic graph structure that represents semantics through nodes and edges, where nodes correspond to semantic elements such as events, entities and arguments, and edges indicate the relationships between them. AMR graphs offer precise descriptions of entities by incorporating their arguments and semantic roles, making them well suited for the RE task (Hu et al., 2023; Zhang and Ji, 2021; Gururaja et al., 2023).

As shown in Figure 1, the input sentence, "... get great joy from eating ...", is parsed into a semantic graph, where the node "source" connects to two entity nodes ("joy" and "eat-01"). This structure explicitly represents the Cause-Effect relation between these two arguments, illustrating how semantic graphs can capture underlying relational meanings beyond surface text.

To bridge the contextual gap caused by missing semantic structure, we propose ***AMR-RE***, an AMR-enhanced retrieval-based ICL method that leverages AMR graphs to select in-context examples based on semantic structure similarity. Evaluations on four English RE datasets show that our method surpasses state-of-the-art methods on three datasets with the *Supervised* AMR-based retriever (Section 4.1). To comprehensively assess our approach, we further evaluate AMR-RE in the more challenging *Unsupervised* setting. Our simple yet effective architecture (Section 4.2) consistently achieves higher F1 scores compared to sentence embedding-based ICL baselines.

2 Preliminaries

2.1 Task Definition

Given a set of pre-defined relation classes $\mathbb{R}$, relation extraction aims to predict the relation $y \in \mathbb{R}$ between the given pair of subject and object entities (e_{sub}, e_{obj}) within the input context $\mathbf{C}$, or if there is no pre-defined relation between them, predict $y =$ NULL. We formalize RE as a language generation task, and introduce the prompt construction in the next section.

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop), pages 333–342

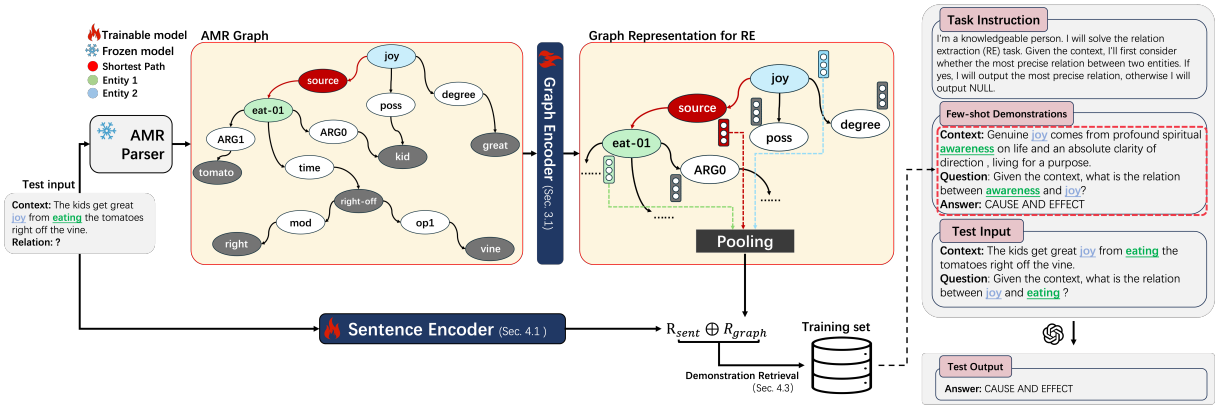

Figure 1: **An overview of our proposed method in the *Supervised* Setting (Section 3, Section 4.1).** Given a test input, we first adopt our AMR-enhanced demonstration retrieval method to select proper demonstrations from the training set. Subsequently, all retrieved demonstrations are included in the prompt construction.

2.2 Prompt Construction

We construct a prompt for each test example. Each prompt consists of three components:

Instructions: We provide a precise description of the RE task and a set of pre-defined relation classes $\mathbb{R}$. The model is required to output the relation corresponding to these predefined classes; if the relation does not belong to any of these classes, the model will output NULL.

ICL Demonstrations: Given one test example, we search k-Nearest Neighbor (kNN) demonstrations via two different frameworks: *Supervised* (Section 4.1) and *Unsupervised* (Section 4.2). All demonstrations are included in the prompt.

Test Input: We provide the test input in the same format as the ICL Demonstrations, and the LLM is expected to output the relation.

3 The AMR-RE Model

This section gives an overview of our ***AMR-RE*** method (Figure 1). Given an input text, AMR-RE first generates its AMR graph using an off-the-shelf AMR parser. A self-supervised graph model then encodes this graph to obtain the graph embeddings. These embeddings are then used to retrieve kNN examples from the training set for ICL (Wan et al., 2023).

Our method leverages the shortest path between two entities for retrieving RE demonstrations, as it aligns with the core objective (supplying semantic structure) of the RE task.

3.1 AMR Graph Encoding

AMR Graph Construction: To generate the AMR graph from the input text, we adopt an off-the-shelf AMR parser[1]. We parse the input sentence into an AMR graph $G = \{V, E, R\}$, where V, E, R are the sets of nodes, edges, and relation types, respectively. In G, the edge labeled $(u, r, v) \in E$, where $u, v \in V$ and $r \in R$, means that there is an edge labeled r from node u to node v.

Self-supervised Graph Encoder: After constructing the AMR graph from the input text, we use a graph encoder to produce the graph embeddings. Shou and Lin (2023) employ a self-supervised approach to train an AMR graph-based neural network; this model assesses the AMR similarity through the encoded representations, hereafter referred to as the *SS-GNN* model. We adapt *SS-GNN* for the RE task by optimizing it on our proposed graph RE representations. Notably, this training framework only depends on the corpus without annotated relation labels. This method explicitly optimizes representations by assessing the similarity between two AMR graphs via a contrastive loss. Training details are added in Appendix A. Given an AMR graph $G = [(u_1, r_1, v_1), \cdots, (u_n, r_n, v_n)]$, G is linearized by a depth-first traversal algorithm $G = [u_1, r_1, v_1, \cdots, u_n, r_n, v_n; \mathcal{A}]$, where $\mathcal{A}$ is the adjacency matrix. G will be fed to SS-GNN to obtain the node representations $H_{node} = \{h^{u_1}_{node}, h^{r_1}_{node}, \cdots, h^{v_n}_{node}\}$ where h^{a}_{node} denotes as the node representation of node a.

$$H_{node} = \text{SS-GNN}([u_1, r_1, v_1, \cdots, v_n]; \mathcal{A}) \quad (1)$$

3.2 Graph Representation for RE

The *SS-GNN* model originally employs mean pooling of all nodes in the AMR graph as the graph representation, which is also used for self-supervised

[1] https://github.com/IBM/transition-amr-parser

training. While this approach has demonstrated significant advancements in overall AMR similarity assessment, it is not optimized for identifying relationships between two specific entities. To address this limitation, we construct graph RE representations specifically designed for RE, focusing on capturing the structural and semantic information of entities and their relationships.

Inspired by previous works, the shortest path between two entities in the semantic structure (Hu et al., 2023) or the syntactic structure (Cheng and Miyao, 2017) often contains crucial information needed to determine relations. Based on these insights, we focus on leveraging the shortest AMR path (SAP) as the most informative subgraph for retrieving the relevant RE demonstrations.

To investigate the optimal way of representing a relation with AMR graph representations for the RE task, we establish fine-grained setups for the graph RE representation R_{graph}. Typically, the shortest path between the entity pair (e_{obj}, e_{sub}) can be denoted as $V_{path} = \{e_{obj}, p_1, p_2, \cdots, p_n, e_{sub}\}$ where $V_{path} \in V$, and p_i represents intermediate nodes on the shortest AMR path (SAP). We investigated two different pooling strategies and two path modeling strategies.

Pooling Strategy: To analyze the impact of the pooling strategy on R_{graph}, we adopt two pooling methods:

(1) *Mean Pooling*: We use the average of all node representations from the shortest path for retrieval, formally $R_{graph} = \frac{1}{|V_{path}|} \sum_{v_i \in V_{path}} h^{v_i}_{node}$.

(2) *Concatenation*: The node representations of the entity pair, $h^{e_{obj}}_{node}$ and $h^{e_{sub}}_{node}$, are concatenated with the mean pooling of the nodes along the shortest AMR path to form the final graph representation, formally $R_{graph} = h^{e_{obj}}_{node} \oplus h^{e_{sub}}_{node} \oplus h_P$, where $h_P = \frac{1}{n} \sum_{i=1}^{n} h^{p_i}_{node}$.

Path Modeling: We use two distinct methods to explore how to effectively leverage information from the shortest path:

(1) *SAP*: This approach strictly isolates all the information from the components not in the shortest path between entity nodes, and only the shortest AMR path is fed to *SS-GNN*, which encodes the node representations along the path. The final graph RE representation R_{graph} is constructed by pooling the node representations within the path.

(2) *SAP+CTX*: We use the whole AMR graph as the input for *SS-GNN*. In this setup, the node representations benefit from bidirectional attention and the GNN adapter, allowing them to integrate contextual information from neighbor nodes. The pooling of the node representations within the shortest AMR path is then formed as the graph RE representation.

By combining the pooling and path modeling strategies, we obtained four distinct configurations, with detailed results provided in Table 5.

4 AMR-Based Demonstration Retrieval

In this section, we introduce two settings for incorporating AMR graph information to retrieve ICL demonstrations. First, we present the *Supervised* setting, where AMR-RE benefits from both graph and sentence RE representations (Section 4.1). To further evaluate the effectiveness of our method, we assess AMR-RE under the more challenging *Unsupervised* setting (Section 4.2). AMR-RE retrieves in-context examples by kNN retrieval from the training set using the relation representation R_{rel} (Section 4.3).

4.1 *Supervised* Setting

In the *Supervised* setting, we integrate both sentence-level and structural information to achieve optimal performance and explore the potential interactions between these two types of representations.

Sentence RE Representations: We use PURE (Zhong and Chen, 2021), an entity marker-based RE model. For example, given the input sentence "And we will see you then", the subject entity "we" and object entity "you", the sentence becomes: "[CLS] And [SUB_ORG] we [/SUB_ORG] will see [OBJ_PER] you [/OBJ_PER] then [SEP]". The final hidden representations of the BERT encoder are denoted as $H_{sent} = \{h^1_{sent}, \cdots, h^m_{sent}\}$ where h^i_{sent} denotes the i-th hidden representation. Let s_{obj} and s_{sub} be the indices of the beginning of the entity markers [SUB_ORG] and [OBJ_PER]. We define the sentence representation as $R_{sent} = h^{s_{obj}}_{sent} \oplus h^{s_{sub}}_{sent}$, where $\oplus$ denotes the concatenation of representations along the first dimension.

Graph RE Representations: We obtain graph RE representations R_{graph} from the SS-GNN as we introduced in Section 3.1.

We use the concatenation of AMR graph embeddings R_{graph} from SS-GNN and sentence embeddings R_{sent} from PURE, formally $R_{rel} = R_{graph} \oplus R_{sent}$. SS-GNN and PURE are fine-tuned

on RE datasets by predicting the relation probability from R_{rel} through a feedforward network. Notably, SS-GNN is first self-supervised trained, then subsequently fine-tuned on RE task.

4.2 *Unsupervised* Setting

We further evaluate our approach in the more challenging *Unsupervised* setting for comprehensively analyzing the effectiveness of AMR graph. In this setting, AMR-RE retrieves examples using only graph RE representations R_{graph}, which means $R_{rel} = R_{graph}$. Note that SS-GNN is only self-supervised on the corpus without annotated relation labels in *Unsupervised* setting. We compare our model with Sentence RE Representations-based baselines.

4.3 Demonstration Retrieval

The relation representation R_{rel} is used to perform kNN retrieval, where the top-k most similar demonstrations are selected and included in the prompt. To efficiently implement kNN demonstration retrieval, we adopt FAISS (Johnson et al., 2019) library for efficient search.

5 Experiments

Backbone LLM: We use OpenAI's GPT-4 as the LLM model in AMR-RE and in all baselines, and we set the number of demonstrations to $k = 10$ in the main results. For a fair comparison, all results are reproduced by ourselves. Baselines such as Wan et al. (2023) originally used GPT-3.5 (text-davinci-003), however, this model is not available through the OpenAI API anymore. In addition, GPT-4 has been shown to outperform its previous versions in several NLP tasks and was the SOTA backbone for ICL at the time. Our method can be easily applied to other backbones as well, however, models such as Llama currently cannot match GPT-4's performance in ICL (Chatterjee et al., 2024).

Evaluation Datasets: We evaluate our model on four English RE datasets. Two general domain RE datasets: SemEval 2010 Task 8 (Hendrickx et al., 2010) and ACE05[2], one temporal RE dataset: TimeBank-Dense (Cassidy et al., 2014), and one scientific domain dataset: SciERC (Luan et al., 2018). Due to the high cost of the OpenAI API, following Wan et al. (2023), we sample a subset of ACE05 dataset (due to its large size) for our experiments. Details of each dataset are provided in Appendix B. We adopt Micro-F1 as evaluation metrics. The hyperparameter settings are provided in the Appendix C.

[2] https://catalog.ldc.upenn.edu/LDC2006T06

6 Main Results

6.1 Results in the *Supervised* Setting

Baselines in *Supervised* Setting: To analyze the effectiveness of the AMR graph, we select two baseline methods for comparison with AMR-RE.

(1) *Supervised* RE Baseline w/o ICL: We implement *PURE* (Zhong and Chen, 2021) as a directly comparable baseline to show the impact of ICL.

(2) Baseline with *Supervised* Retrievers: We implement *GPT-RE_FT* (Wan et al., 2023) as the baseline with a *Supervised* retriever. GPT-RE_FT employs representations encoded by PURE (Zhong and Chen, 2021).

Results: Table 1 shows our results. Overall, AMR-RE outperforms the baselines in the *Supervised* setting. This indicates that the more explicit representation of AMR graphs enhances the quality of the retrieved demonstrations. In the *Supervised* setting, AMR-RE achieves SOTA performance on the SemEval, SciERC and TB-Dense datasets while delivering competitive results on the ACE05 dataset. The results indicate that the fine-tuned structure representation benefits from both structural and semantic information. However, ACE05 contains a large proportion of the samples annotated as NULL relation, which introduces significant noise. This can mislead the model during both retriever training and ICL inference, resulting in decreased performance compared to the fully-supervised baseline, PURE.

6.2 Results in the *Unsupervised* Setting

Baselines in *Unsupervised* Setting: We select three baselines that are comparable to AMR-RE in *Unsupervised* setting. The details of each baseline are introduced below:

(1) *GPT-Random:* we randomly select few-shot ICL demonstrations with additional constraints to ensure a more uniform label distribution;

(2) *GPT-Sent:* we follow Gutierrez et al. (2022) to retrieve kNN demonstrations with SimCSE (Gao et al., 2021), which is a widely used sentence embedding model;

(3) *GPT-RE_Entity+:* we adopt the entity-prompted sentence embedding proposed by Wan et al. (2023) that incorporates both the entity pair and contextual information for retrieval.

Method	Retriever	SemEval (Δ%)	TB-DENSE (Δ%)	SciERC (Δ%)	ACE05 (Δ%)	Avg
Supervised Setting						
PURE	-	90.77	66.70	67.08	**68.62**	73.57
GPT-RE_FT	*PURE*	91.46	67.58	67.32	68.59	73.74
AMR-RE (Ours)	*SS-GNN+PURE*	**91.97** (↑ **0.6**)	**71.54** (↑ **5.9**)	**68.10*** (↑ **1.1**)	67.94* (↓ 0.9)	**74.89**
Unsupervised Setting						
GPT-Random	-	67.83	22.03	16.48	9.73	29.02
GPT-Sent	SimCSE	77.64	28.73	21.60	10.04	34.50
GPT-RE_Entity+	SimCSE	80.25	31.19	26.15	13.10	37.67
AMR-RE (Ours)	*SS-GNN*	**84.68** (↑ **5.5**)	**38.17** (↑ **22.4**)	**27.89*** (↑ **6.7**)	**15.04*** (↑ **14.8**)	**41.45**

Table 1: **Main results.** We set the number of demonstrations to $k = 10$. For AMR-RE, we only report the best results from the four distinct configurations obtained by combining the pooling and path modeling strategies, explained in Section 3.1 (see Table 5 for detailed results). Underlined results refer to the *SAP* graph RE representation, otherwise, *SAP+CTX* is applied. The Δ% indicates the corresponding differences in percentage when compared to GPT-RE_FT and GPT-RE_Entity+ in *Supervised* and *Unsupervised* settings respectively. The Avg column shows the average score for all datasets. The highest results are in **bold**. $*$ denotes that this result is implemented by concatenation pooling, otherwise, mean pooling is used.

Method	SemEval (Δ%)	SciERC (Δ%)
Supervised Setting		
AMR-RE	**91.97**	**68.10**
w/o self-sup	90.82 (↓ 1.3)	67.04 (↓ 1.6)
w/o R_{sent}	89.71 (↓ 2.5)	67.19 (↓ 1.3)
w/o R_{graph}	91.46 (↓ 0.6)	67.32 (↓ 1.2)
Unsupervised Setting		
AMR-RE	**84.68**	**27.56**
w/o self-sup	81.67 (↓ 3.6)	26.01 (↓ 5.6)

Table 2: **Ablation study.** For the full model, we show the best configuration results from Table 1. *w/o self-sup* indicates that the retriever is not self-supervised on the target dataset. The Δ% is the percentage of corresponding difference.

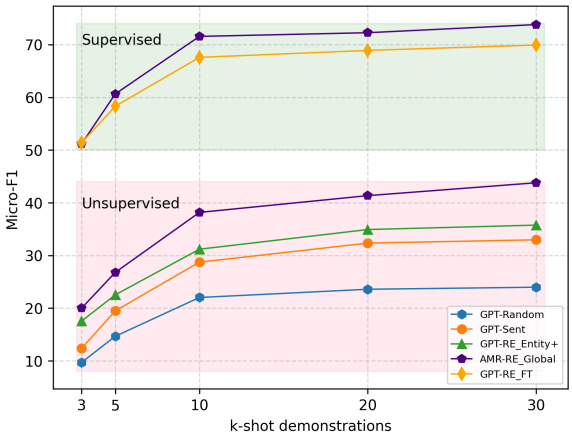

Figure 2: Performance for the different number of few-shot examples on TB-Dense.

Results: Table 1 shows our results in the *Unsupervised* setting. AMR-RE consistently outperforms the baselines on all four datasets. These findings underscore the efficacy of AMR-enhanced graph RE representations in effectively capturing relational information. In particular, by focusing on the shortest AMR path, AMR-RE highlights core entities and the semantic relations between them, thereby reducing noise and providing clearer relational cues compared to conventional sentence-embedding-based approaches.

7 Ablation Study

Table 2 illustrates the impact of self-supervision on the graph encoder and the roles of sentence and graph RE representations in the relation representations. The results show that self-supervision enhances performance, with graph (R_{graph}) and sentence (R_{sent}) representations both being crucial in the *Supervised* setting. We also investigated the impact of the number of demonstrations on performance. Figure 2 shows that AMR-RE consistently outperforms the baselines across all k-shots, demonstrating the effectiveness of incorporating AMR graphs for retrieval.

8 Case Study

To demonstrate how semantic structure similarity enables the retrieval of highly relevant demonstrations and surpasses sentence-based baselines on RE ICL, we present two representative case studies in the *Unsupervised* Setting. Figure 3 illustrates that our proposed AMR enhanced retrieval method effectively captures both the similarity of event structure and the semantics of the entities. This shows that demonstrations with high semantic structure similarity serve as more suitable and informative RE demonstrations for ICL. Figure 4 highlights the effectiveness of **AMR-RE**. Our proposed method successfully retrieves few-shot RE demonstrations with semantically equivalent entities (e.g., "proto-

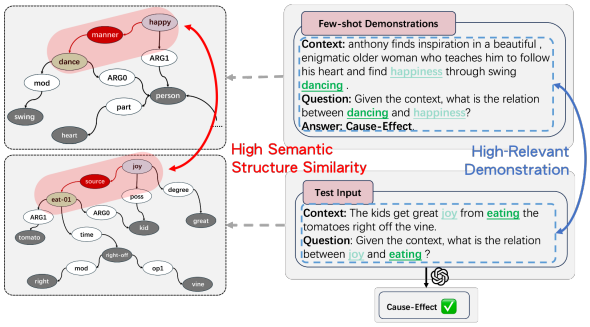

Figure 3: A case study of semantic structure similarity. The demonstration with similar semantic structure enables the LLM to correctly generate the gold label, "Cause-Effect".

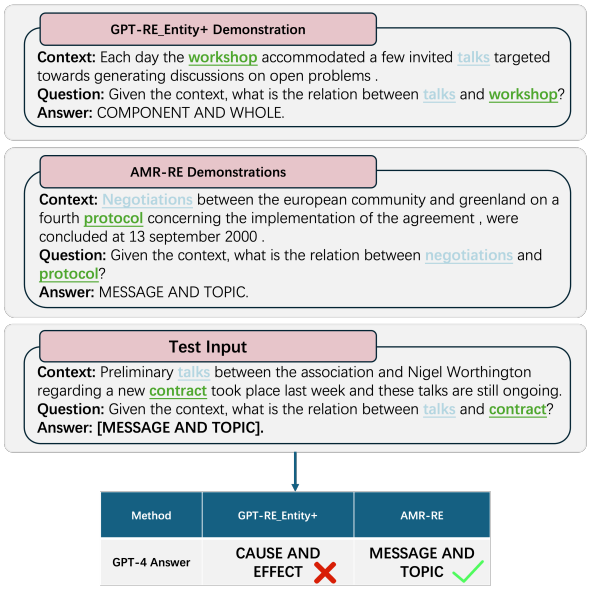

Figure 4: A case study of **AMR-RE** retrieved demonstration quality. MESSAGE AND TOPIC is the gold label.

col"–"contract", "negotiations"–"talks"), while also capturing implicit relational connections. It demonstrates AMR-RE's ability to align both explicit and implicit semantic information for improved relation extraction. In contrast, the sentence-based retrieval method fails to model such information.

9 Conclusions

We proposed ***AMR-RE***, an AMR-enhanced retrieval-based ICL method that uses AMR graphs to select demonstrations based on semantic structure similarity. Evaluations on four English RE datasets show that AMR-RE outperforms the baselines. This underscores the effectiveness of combining graph learning with LLMs for relation extraction. Our experiments further demonstrate that AMR graph information can lead to more accurate and robust relation extraction, even in *Unsupervised* settings.

10 Limitations

We focused our work on: 1) demonstrating the effectiveness of graph similarity in retrieval-based ICL on the RE task. However, our work can be generalized beyond RE, as AMR is a universal semantic analysis tool applicable to other tasks, and ICL is also not restricted to RE; 2) evaluating our method on English RE datasets, mainly because AMR parsers only offer promising performance in English (Cai et al., 2021). There are other semantic tools, such as multilingual dependency parser (Üstün et al., 2020), for constructing graphs that extend beyond English.

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Deng Cai, Xin Li, Jackie Chun-Sing Ho, Lidong Bing, and Wai Lam. 2021. Multilingual AMR parsing with noisy knowledge distillation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2778–2789, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Taylor Cassidy, Bill McDowell, Nathanael Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 501–506, Baltimore, Maryland. Association for Computational Linguistics.

Anwoy Chatterjee, Eshaan Tanwar, Subhabrata Dutta, and Tanmoy Chakraborty. 2024. Language models can exploit cross-task in-context learning for data-scarce novel tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11568–11587, Bangkok, Thailand. Association for Computational Linguistics.

Fei Cheng and Yusuke Miyao. 2017. Classifying temporal relations by bidirectional LSTM over dependency paths. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–6, Vancouver, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sireesh Gururaja, Ritam Dutt, Tinglong Liao, and Carolyn Rosé. 2023. Linguistic representations for fewer-shot relation extraction across domains. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7502–7514, Toronto, Canada. Association for Computational Linguistics.

Bernal Gutierrez, Nikolas McNeal, Clayton Washington, You Chen, Lang Li, Huan Sun, and Yu Su. 2022. Thinking about GPT-3 in-context learning for biomedical IE? think again. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4497–4512, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ridong Han, Tao Peng, Chaohao Yang, Benyou Wang, Lu Liu, and Xiang Wan. 2023. Is information extraction solved by chatgpt? an analysis of performance, evaluation criteria, robustness and errors. *arXiv preprint arXiv:2305.14450*.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

Zhilei Hu, Zixuan Li, Xiaolong Jin, Long Bai, Saiping Guan, Jiafeng Guo, and Xueqi Cheng. 2023. Semantic structure enhanced event causality identification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10901–10913, Toronto, Canada. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Guozheng Li, Peng Wang, Wenjun Ke, Yikai Guo, Ke Ji, Ziyu Shang, Jiajun Liu, and Zijie Xu. 2024. Recall, retrieve and reason: Towards better in-context relation extraction. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 6368–6376. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Xilai Ma, Jing Li, and Min Zhang. 2023. Chain of thought with explicit evidence reasoning for few-shot relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2334–2352, Singapore. Association for Computational Linguistics.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. 2003. The timebank corpus. In *Corpus linguistics*, volume 2003, page 40. Lancaster, UK.

Ziyi Shou and Fangzhen Lin. 2023. Evaluate AMR graph similarity via self-supervised learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16112–16123, Toronto, Canada. Association for Computational Linguistics.

Xiaofei Sun, Linfeng Dong, Xiaoya Li, Zhen Wan, Shuhe Wang, Tianwei Zhang, Jiwei Li, Fei Cheng, Lingjuan Lyu, Fei Wu, et al. 2023. Pushing the limits of chatgpt on nlp tasks. *arXiv preprint arXiv:2306.09719*.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. UDapter: Language adaptation for truly Universal Dependency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2302–2315, Online. Association for Computational Linguistics.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. SemEval-2013 task 1: TempEval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9, Atlanta, Georgia, USA. Association for Computational Linguistics.

Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi. 2023. GPT-RE: In-context learning for relation extraction using large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3534–3547, Singapore. Association for Computational Linguistics.

Xiaozhi Wang, Yulin Chen, Ning Ding, Hao Peng, Zimu Wang, Yankai Lin, Xu Han, Lei Hou, Juanzi Li, Zhiyuan Liu, Peng Li, and Jie Zhou. 2022. MAVEN-ERE: A unified large-scale dataset for event coreference, temporal, causal, and subevent relation extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 926–941, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zixuan Zhang and Heng Ji. 2021. Abstract Meaning Representation guided graph encoding and decoding for joint information extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 39–49, Online. Association for Computational Linguistics.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online. Association for Computational Linguistics.

A Self-supervised Training for AMR Graph Encoding

SS-GNN (Shou and Lin, 2023) adopts a self-supervised approach, Contrastive Tension (CT) to optimize the representation of an AMR graph. The main assumption is that AMR graphs with adjacent distributions have similar meanings. In our work, we adapt this approach to our novel AMR graph representation.

Two independent transformer-based encoders that also incorporate graph neural networks are identically initalized. The training objective is to maximize the dot product between positive pairs (G_p, G_p^+) while minimizing the dot product between negative pairs (G_p, G_p^-). For each randomly selected AMR graph G_p, we use $G_p^+ = G_p$ to create a positive pair. Then, we construct negative instances by pairing G_p with K randomly sampled different graphs. The $K+1$ instances are included in the same batch. The training contrastive loss $\mathcal{L}$ is binary cross-entropy between similarity scores and labels.

$$\mathcal{L} = \begin{cases} -\log \sigma(h_{graph} \cdot h^+_{graph}) \\ -\log \sigma(1 - h_{graph} \cdot h^-_{graph}) \end{cases} \quad (2)$$

Hyperparameter	Value
Engine Name	GPT-4-0314
Temperature	0
Top_P	1
Frequency_penalty	0
Presence_penalty	0
Best_of	1

Table 3: GPT-4 hyperparameters.

where σ refers to the Logistic function; h_{graph} is the graph representation. The model is then updated to compute the similarity between the two graphs.

B Evaluation Datasets

In this section, we describe the evaluation datasets used in our experiments. Table 4 shows the statistics for each dataset.

SemEval 2010 Task 8 (Hendrickx et al., 2010): This data set focuses on the semantic relations between pairs of nominals. It was annotated from general domain resources. The task is to classify the semantic relations into one of nine directed relation types: Cause-Effect, Instrument-Agency, Product-Producer, Content-Container, Entity-Origin, Entity-Destination, Component-Whole, Member-Collection, Message-Topic, and Other (to indicate that there is no relation between the pair of nominals). An example of a sentence with an event pair that holds the Cause-Effect relation is shown below:

> The **(e1:discomfort)** from the **(e2:injury)** was now precluding him from his occupation which involved prolonged procedures in the standing position.

ACE05: This dataset contains entities, relations, and events annotated from resources from domains including newswire, broadcast news, broadcast conversation, weblog, discussion forums, and conversational telephone speech. It requires identifying semantic relations into the following six types: artifact, general-affiliation, organization-affiliation, part-whole, person-social, physical. The following example contains an entity pair with the part-whole relation:

> Witnesses say they heard blasts around a presidential complex in the **(e1:center)** of the **(e2:city)**.

Dataset	# Relation	# Train	# Dev	# Test (# Subset)
SemEval	9	6,507	1,493	2,717 (2,717)
TB-Dense	6	7,553	898	2,299 (2,299)
SciERC	7	16,872	2,033	4,088 (4,088)
ACE05	6	121,368	27,597	24,420 (2,442)

Table 4: Statistics of the evaluation datasets. # Subset denotes the number of instances sampled from the original test set, due to the high cost of the OpenAI API.

TB-Dense (Cassidy et al., 2014): TB-Dense is a public benchmark for temporal relation extraction (TRE). It was annotated from TimeBank (Pustejovsky et al., 2003) and TempEval (UzZaman et al., 2013). We use a preprocessed version from (Wang et al., 2022) for experiments. TB-Dense annotates temporal relations for event pairs within adjacent sentences. To handle this, we separately parse the two sentences into AMR graphs and then connect the two graphs through a shared root node following (Cheng and Miyao, 2017). Given a passage and two event points, the task is to classify the relations between events into one of six types: BEFORE, AFTER, SIMULTANEOUS, VAGUE, IS_INCLUDED, and INCLUDES. An example with two events, e1 and e2 (in bold) that hold the SIMULTANEOUS relation is shown below:

> Nobody **(e1:hurried)** her up. No one **(e2:held)** her back.

SciERC (Luan et al., 2018): This dataset includes annotations for scientific entities and their relations annotated from 500 scientific abstracts taken from Artificial Intelligence conferences and workshops proceedings. The relation types are: *used-for*, *feature-of*, *hyponym-of*, *part-of*, *compare*, *conjunction* and *corefence*. Following example contains the *feature-of* relation between two entities:

> They improve the reconstruction results and enforce their consistency with a **(e1:priori knowledge)** about **(e2:object shape)**.

C Hyperparameters

GPT-4: We used GPT-4 by the OpenAI API [3] during the experiments. The hyperparameters used can be found in Table 3, we report the result of the single run for all experiments.

Unsupervised Sentence Embedding Model: We use the sentence embedding method SimCSE in our experiments. We use the *sup-simcse-bert-base-uncased* model as the base encoder.

Graph Encoder (SS-GNN): During training, we set the positive ratio to 4/16, meaning each batch of 16 contains 4 positive graph pairs and 12 negative pairs. Specifically, we sampled 4 graphs and generated one positive pair and three negative pairs for each graph. The transformer parameters were initialized using the uncased BERT base model (Devlin et al., 2019), while the graph adapter parameters were initialized randomly. Hyperparameters were set as follows: 1 epoch, learning rate as 1e-5, dropout rate as 0.1, and graph adapter size as 128. We experimented with sequence length of 128 for SemEval and 256 for the other three datasets. The training was done using NVIDIA Quadro RTX 8000.

Supervised RE Model (PURE): To maintain consistency across datasets, we use a single-sentence setup for Semeval, as it is a sentence-level relation extraction dataset. For pre-trained language models (PLMs), we follow PURE by using scibert-scivocab-uncased (Beltagy et al., 2019) as the base encoder for SciERC and bert-base-uncased (Devlin et al., 2019) for the other three datasets. We also adhere to the hyperparameters specified in their paper.

D Results of All AMR-RE configurations

Table 5 shows the results for all the configurations in our experiments.

[3] https://platform.openai.com/docs/api-reference/introduction

Setting	Path	Pooling	SemEval	TB-DENSE	SciERC	ACE05	Avg
Supervised	SAP+CTX	*Mean*	90.84	**71.54**	67.92	67.37	74.22
		Concatenation	90.03	70.56	**68.10**	**67.94**	74.36
	SAP	*Mean*	**91.97**	68.23	67.81	66.80	73.70
		Concatenation	91.70	67.89	68.04	67.21	73.71
Unsupervised	SAP+CTX	*Mean*	81.40	**38.17**	27.64	14.82	40.51
		Concatenation	79.48	37.78	**27.89**	**15.04**	40.05
	SAP	*Mean*	**84.68**	35.64	27.56	14.65	40.63
		Concatenation	83.51	33.75	27.61	14.69	39.89

Table 5: **AMR-RE results with all configurations.** The results in **bold** are reported in the main results.