

**Bridging Neurons and Symbols
for Natural Language Processing and Knowledge Graphs
Reasoning @ COLING 2025**

January 20th, 2025

©2025 The Association for Computational Linguistics

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 979-8-89176-210-7

Message from the Program Chairs

Recent exploration shows that LLMs, e.g., ChatGPT, may pass the Turing test in human-like chatting but have limited capability even for simple reasoning tasks (Biever, 2023). It remains unclear whether LLMs reason or not (Mitchell, 2023). Human reasoning has been characterized as a dual-process phenomenon (see (Sun, 2023) for a general overview) or as mechanisms of fast and slow thinking (Kahneman, 2011). These findings suggest two directions for exploring neural reasoning: starting from existing neural networks to enhance the reasoning performance with the target of symbolic-level reasoning, and starting from symbolic reasoning to explore its novel neural implementation (Dong et al., 2024). These two directions will ideally meet somewhere in the middle and will lead to representations that can act as a bridge for novel neural computing, which qualitatively differs from traditional neural networks, and for novel symbolic computing, which inherits the good features of neural computing. Hence the name of our workshop, with a focus on Natural Language Processing and Knowledge Graph reasoning. This workshop promotes research in both directions, particularly seeking novel proposals from the second direction.

Organizing Committee

Kang Liu (Chinese Academy of Sciences)

Yangqiu Song (The Hong Kong University of Science and Technology)

Zhen Han (Amazon Inc.)

Rafet Sifa (University of Bonn)

Shizhu He (Institute of Automation, Chinese Academy of Sciences)

Yunfei Long (University of Essex)

Table of Contents

<i>Chain of Knowledge Graph: Information-Preserving Multi-Document Summarization for Noisy Documents</i>	
Kangil Lee, Jinwoo Jang, Youngjin Lim and Minsu Shin	1
<i>CEGRL-TKGR: A Causal Enhanced Graph Representation Learning Framework for Temporal Knowledge Graph Reasoning</i>	
Jinze Sun, Yongpan Sheng, Lirong He, Yongbin Qin, Ming Liu and Tao Jia	6
<i>Reasoning Knowledge Filter for Logical Table-to-Text Generation</i>	
Yu Bai, baoqiang liu, Shuang Xue, Fang Cai, Na Ye and Guiping Zhang	18
<i>From Chain to Tree: Refining Chain-like Rules into Tree-like Rules on Knowledge Graphs</i>	
Wangtao Sun, Shizhu He, Jun Zhao and Kang Liu	31
<i>LAB-KG: A Retrieval-Augmented Generation Method with Knowledge Graphs for Medical Lab Test Interpretation</i>	
Rui Guo, Barry Devereux, Greg Farnan and Niall McLaughlin	40
<i>Bridging Language and Scenes through Explicit 3-D Model Construction</i>	
Tiansi Dong, Writwick Das and Rafet Sifa	51
<i>VCRMNER: Visual Cue Refinement in Multimodal NER using CLIP Prompts</i>	
Yu Bai, Lianji Wang, Xiang Liu, Haifeng Chi and Guiping Zhang	61
<i>Neuro-Conceptual Artificial Intelligence: Integrating OPM with Deep Learning to Enhance Question Answering Quality</i>	
Xin Kang, Veronika Shteyngardt, Yuhan Wang and Dov Dori	71
<i>Emergence of symbolic abstraction heads for in-context learning in large language models</i>	
Ali Al-Saeedi and Aki Harma	86
<i>Linking language model predictions to human behaviour on scalar implicatures</i>	
Yulia Zinova, David Arps, Katharina Spalek and Jacopo Romoli	97
<i>Generative FrameNet: Scalable and Adaptive Frames for Interpretable Knowledge Storage and Retrieval for LLMs Powered by LLMs</i>	
Harish Tayyar Madabushi, Taylor Hudson and Claire Bonial	107

Conference Program

Chain of Knowledge Graph: Information-Preserving Multi-Document Summarization for Noisy Documents

Kangil Lee, Jinwoo Jang, Youngjin Lim and Minsu Shin

CEGRL-TKGR: A Causal Enhanced Graph Representation Learning Framework for Temporal Knowledge Graph Reasoning

Jinze Sun, Yongpan Sheng, Lirong He, Yongbin Qin, Ming Liu and Tao Jia

Reasoning Knowledge Filter for Logical Table-to-Text Generation

Yu Bai, baoqiang liu, Shuang Xue, Fang Cai, Na Ye and Guiping Zhang

From Chain to Tree: Refining Chain-like Rules into Tree-like Rules on Knowledge Graphs

Wangtao Sun, Shizhu He, Jun Zhao and Kang Liu

LAB-KG: A Retrieval-Augmented Generation Method with Knowledge Graphs for Medical Lab Test Interpretation

Rui Guo, Barry Devereux, Greg Farnan and Niall McLaughlin

Bridging Language and Scenes through Explicit 3-D Model Construction

Tiansi Dong, Writwick Das and Rafet Sifa

VCRMNER: Visual Cue Refinement in Multimodal NER using CLIP Prompts

Yu Bai, Lianji Wang, Xiang Liu, Haifeng Chi and Guiping Zhang

Neuro-Conceptual Artificial Intelligence: Integrating OPM with Deep Learning to Enhance Question Answering Quality

Xin Kang, Veronika Shteyngardt, Yuhan Wang and Dov Dori

Emergence of symbolic abstraction heads for in-context learning in large language models

Ali Al-Saeedi and Aki Harma

Linking language model predictions to human behaviour on scalar implicatures

Yulia Zinova, David Arps, Katharina Spalek and Jacopo Romoli

Generative FrameNet: Scalable and Adaptive Frames for Interpretable Knowledge Storage and Retrieval for LLMs Powered by LLMs

Harish Tayyar Madabushi, Taylor Hudson and Claire Bonial

Chain of Knowledge Graph: Information-Preserving Prompting for Noisy Multi-Document

* Youngjin Lim¹, * Kangil Lee¹, * Jin-Woo Jang¹, * MinSu Shin¹

¹LG Energy Solution, Seoul, Republic of Korea

Abstract

With the advent of large language models, the complexity of multi-document summarization task has been substantially reduced. The summarization process must effectively handle noisy documents that are irrelevant to the main topic while preserving essential information. Recently, Chain-of-Density (CoD) and Chain-of-Event (CoE) have proposed prompts to effectively handle the noisy documents by using entity-centric approaches for the summarization. However, CoD and CoE are prone to information loss during entity extraction due to their tendency to overly filter out entities perceived as less critical but that could still be important.

In this paper, we propose a novel instruction prompt termed as Chain of Knowledge Graph (CoKG) for multi-document summarization. Our prompt extracts entities and constructs relationships between entities to form a Knowledge Graph (KG). Next, the prompt enriches these relationships to recognize potentially important entities and assess the strength of each relation. If the acquired KG meets a pre-defined quality level, the KG is used to summarize the given documents. This process helps alleviate the information loss in multi-document summarization. Experimental results demonstrate that our prompt effectively preserves key entities and is robust to noisy documents.

1 Introduction

The rise of foundation Large Language Models (LLMs) is redefining the landscape of various natural language processing (NLP) tasks. LLM-powered approaches are surpassing conventional supervised learning methods in tasks such as reasoning, sentiment analysis, and others, often with just a single prompt.

With this advancement, text summarization has entered a new phase (Pu et al., 2023). Instead of

*All authors contributed equally to this work.

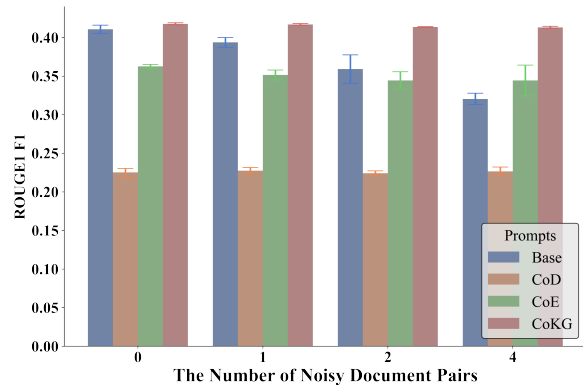


Figure 1: ROUGE-1 score charts showing the impact of adding noisy documents. Standard deviations are represented by error bars on each bar. ‘Base’ refers to a generic instruction such as “Summarize these documents”. The low ROUGE1 scores of CoD and CoE indicate that these methods suffer from information loss. It is observed that the performance of ‘Base’ decrease as the number of noisy documents increase.

relying on ‘golden’ answers, texts can now be summarized with greater flexibility using LLMs. Furthermore, instructions enable precise tailoring of summary length and style. Examining prompt-based text summarization methods, we find that most of the recently proposed methods fall under the Chain-of-Thought (CoT) category (Zhang et al., 2024). Among the CoT approaches, CoD is a representative method (Adams et al., 2023). CoD starts with a sparse entity set and refines it iteratively to obtain a denser entity set while balancing detail and abstraction for summarization.

In real life, Multi-Document Summarization (MDS) is needed across various fields for diverse objectives. However, when collecting various documents from the web, such as news articles and community posts related to a specific event, it is common to come across documents that are irrelevant to the main topic. We define these documents as noisy documents. To effectively deal with MDS,

CoE is recently proposed (Bao et al., 2024). CoE consists of four sequential steps: event extraction, event abstraction, statistical event analysis and final document summarization.

Although CoE and CoD prompts are designed to be robust to noisy documents, their prompts can lead to significant information loss in MDS 3. In CoE, relying solely on frequency to determine entity importance can result in the omission of contextually significant entities. Meanwhile, CoD can suffer from lack of entities due to stringent conditions to be entities.

For noise-robust and information-preserving summarization, we propose a novel instruction prompt termed as CoKG for MDS. CoKG aims to extract enriched entities to minimize information loss. Since our approach is entity-centric summarization, it is also robust to noisy documents. Our contributions are as follows.

- We propose a novel entity-centric MDS prompt that is relatively free from information loss by using knowledge graph.
- We demonstrate that chaining and expanding entities reduce information loss and enhance robustness to noisy documents.

2 Related Works

Text Summarization. Text summarization has two distinct tracks: extractive and abstractive summarization. In the context of neural machine approaches, extractive summarization is regarded as a combination of sequence labeling and selection tasks. (Nallapati et al., 2017; Zhou et al., 2018). Abstractive summarization is regarded as a sequence-to-sequence problem (See et al., 2017; Liu and Lapata, 2019) formulated as a source document $x = [x_1, \dots, x_n]$ to a target summary $y = [y_1, \dots, y_m]$, where n and m are the number of tokens.

Despite the effectiveness of supervised methods, scalability issue is still challenging. With the rise of LLMs, since LLMs can generate a summary with a few lines of instruction, prompt-based summarization has gained attention to address the issue (Kuznia et al., 2022; Liu et al., 2022; Adams et al., 2023).

MDS is similar to general text summarization but differs in that it integrates and deals with diverse perspectives on a single topic. For example, CoE minimizes irrelevant information and focuses

on key events for a concise summary (Bao et al., 2024).

3 Chain of Knowledge Graph

3.1 Preliminary

Terminologies are defined as follows.

- $D = \{T_1, T_2, \dots, T_N\}$ represents either a single document or set of documents.
- E_i represents a set of entities or events identified at iteration i .
- S_i represents a summary at iteration i .

For MDS, CoD starts by extracting an initial summary S_0 from D , with E_0 as an empty set. At each step $i \geq 1$, CoD identifies missing entities M_i by comparing D and previous summary S_{i-1} and following five conditions and six guidelines. These missing entities M_i are used to update E_{i-1} , resulting in E_i . Using E_i , CoD refines S_{i-1} into a new summary S_i . This process is repeated five times, resulting in a final summary after the last iteration. Meanwhile, CoE first extracts many events E_0 from D and consolidates E_0 into a set of key events A . Then, CoE identifies statistically the most common abstract events and utilizes the events to generate a summary S_0 .

The stringent five conditions of CoD for adding entities limit the capacity to incorporate additional ones which leads to information loss during summarization. Thus, we propose a novel instruction prompt termed as CoKG.

3.2 Instruction Prompt

To address the information loss, our prompt constructs KG to preserve critical information as much as possible by enriching entities and relations. The constructed KG is effectively utilized to summarize the given documents.

CoKG prompt consists of six steps as follows (See Figure 2).

1. **Identify entities:** Identify and extract key entities E_i^o to minimize the influence of irrelevant information from D .
2. **Construct relations:** Construct relations between the elements of E_i^o . The relations are expressed as verbs, adjectives, and phrasal verbs.

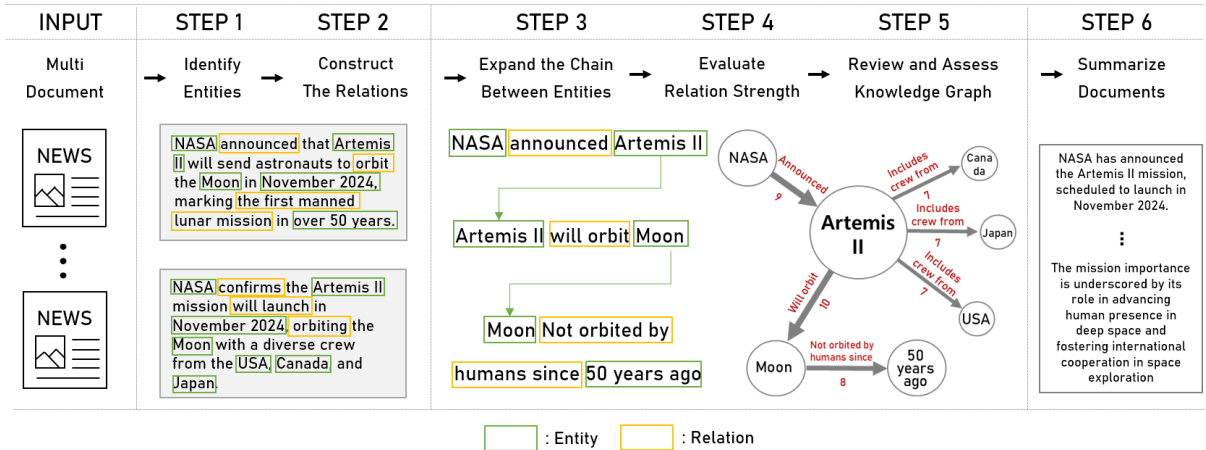


Figure 2: Overall process to create a summary for multi-document. First, identify key entities and construct relationships between entities (STEP 1 and 2). Second, expand and chain the entities to create a knowledge graph, then evaluate the strength of the relationships and review the overall knowledge graph (STEP 3, 4, and 5). Third, generate a summary with knowledge graph (STEP 6).

3. **Expand the chain between entities:** Since each entity can have multiple relationships, it is necessary to sufficiently expand and chain entities ($E_i^o \rightarrow E_i$) to provide rich contextual information for summary.
4. **Evaluate relation strength:** Evaluate the entity connections by assigning scores ranging from 1 to 10, where 1 represents the weakest, 10 the strongest, and 5 a moderate link. This score depends on the strength of the relationships as found within the context of the documents.
5. **Review and assess knowledge graph:** Quantitatively evaluate the KG obtained from the previous four steps. If the KG effectively captures the main context and key entities of the given documents by including all relevant entities and relationships, assign a high score. The score ranges from 1 to 10, and if it does not achieve at least 7, return to Step 1 and reconstruct the KG.
6. **Summarize documents:** Finally, generate a summary based on the final KG and the given documents.

To prevent information loss, CoKG chains and expands entities. Furthermore, since CoKG is entity-centric summarization prompt, it is robust to noisy documents. Thus, the CoKG prompt can be considered an effective prompt for MDS.

Table 1: Evaluation results on the Multi-News dataset for assessing information loss. Even though CoKG compresses documents into key entity-focused graph, we find that CoKG experiences relatively no loss of information compared to the Base prompt.

	Base	CoD	CoE	CoKG
ROUGE-1	0.417	0.226	0.360	0.418
ROUGE-2	0.114	0.055	0.099	0.114
ROUGE-L	0.189	0.123	0.178	0.189
METEOR	0.266	0.111	0.195	0.269

4 Experiments

4.1 Datasets

To evaluate our instruction prompt for MDS, we use two datasets: Multi-News and PeerSum.

Multi-News consists of news articles and professional human-written summaries (Fabbri et al., 2019). Each summary includes links to the original articles cited. We use 100 randomly sampled sets of news collection from the test dataset.

PeerSum consists of review comments from OpenReview (Li et al., 2023). These comments range from official reviewers to public readers on a paper. The meta-review is considered as the reference summary. We used 100 randomly sampled sets of review collection from the test dataset.

4.2 Experimental Setup

Evaluation Metrics. We use the widely adopted ROUGE (Lin, 2004) and METEOR (Banerjee and Lavie, 2005) for evaluation. In addition, we uti-

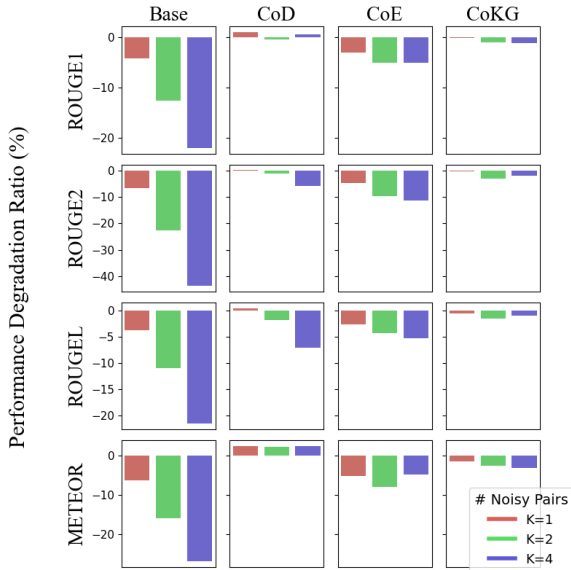


Figure 3: Performance degradation ratio resulting from the addition of noisy document pairs to the original set of documents. We use Multi-News dataset to obtain these results. K means the number of noisy document pairs. Performance degradation ratio represents the rate of performance drop when noisy documents are introduced.

Table 2: Evaluation results on the Multi-News and PeerSum datasets. We assess each prompt using G-Eval to evaluate summary quality from a human-friendly perspective. \uparrow indicates that higher value is better, and \downarrow indicates that lower value is better.

	Base	CoD	CoE	CoKG
Multi-News				
Coherence (\uparrow)	4.301	3.990	4.467	4.555
Consistency (\uparrow)	4.630	4.50	4.644	4.685
Fluency (\uparrow)	2.731	2.587	2.680	2.743
Relevance (\uparrow)	4.722	4.537	4.781	4.818
Average Rank (\downarrow)	2.75	4	2.25	1
PeerSum				
Coherence (\uparrow)	3.923	3.153	3.786	4.090
Consistency (\uparrow)	2.240	1.833	2.047	2.107
Fluency (\uparrow)	2.893	2.784	2.858	2.946
Relevance (\uparrow)	3.103	2.412	2.898	3.059
Average Rank (\downarrow)	1.5	4	3	1.5

lize G-eval (Liu et al., 2023) as a metric that has a high correlation with human evaluations for Natural Language Generation (NLG).

Comparison Prompts. To evaluate CoKG, we compare Base, CoD, and CoE prompts. The Base prompt is a generic instruction for summarization : "Summarize the document below, which includes multiple texts on similar topics."

Model Selection. CoKG requires two abilities : decomposing instructions into several parts to easily handle each step and understanding the logical flow and connections. We selected Claude Sonnet 3.5 (ANTHROPIC, 2024) based on its state-of-the-art performance on decompositional and diagrammatic reasoning (Huang et al., 2024).

Noise Test. To evaluate robustness against noisy documents, we introduced text noise into a set of documents by appending unrelated article pairs both before and after the given document set.

4.3 Experimental Results

We evaluated how well the proposed prompt preserves information by recall-oriented metrics. Table 1 shows that CoKG exhibits almost no information loss compared to the Base prompt. In contrast, CoD and CoE show significant information loss. Table 2 demonstrates that the CoKG achieves competitive performance on two MDS datasets. In the Multi-News, our prompt achieves the best performance across all metrics. However, on the PeerSum, our approach performs relatively worse than the Base on two metrics. We infer that these results attribute to the entity expansion process.

Meanwhile, Figure 3 illustrates that CoKG is robust to noisy documents. As K increases, the performance drop for Base and CoE prompts becomes more pronounced, whereas CoKG shows relative robustness to noisy documents. Since CoD performs poorly when $K = 0$ compared to other prompts, it can be inferred that CoD does not perform well in MDS. Based on this, CoD may not be robust to noisy documents but rather unsuitable for MDS.

In conclusion, we find that CoKG effectively preserves information while also being robust to noisy documents.

5 Conclusion

MDS complexity is eased by advent of LLM-based approaches. However, the previous approaches often suffer from information loss. In addition, since generic prompt tends to show inferior performance to severely noisy document set, entity-centric approach is necessary.

Thus, we propose CoKG that is robust to noisy documents and has information-preserving property. CoKG maximally extracts topic-related entities to minimize information loss. In noise test, we observe that our approach is resilient to noise. In addition, the results from the Multi-News and

PeerSum benchmarks demonstrate that CoKG effectively preserves information and that its summaries closely align with human-generated ones. These findings suggest CoKG produces a reliable summary for multi-document.

References

- Griffin Adams, Alexander Fabbri, Faisal Ladhak, Eric Lehman, and Noémie Elhadad. 2023. From sparse to dense: Gpt-4 summarization with chain of density prompting. *arXiv preprint arXiv:2309.04269*.
- ANTHROPIC. 2024. [Claude 3.5 sonnet](#). Introducing Claude 3.5 Sonnet.
- Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.
- Songlin Bao, Tiantian Li, and Bin Cao. 2024. Chain-of-event prompting for multi-document summarization by large language models. *International Journal of Web Information Systems*, (ahead-of-print).
- Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. [Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.
- Zhen Huang, Zengzhi Wang, Shijie Xia, Xuefeng Li, Haoyang Zou, Ruijie Xu, Run-Ze Fan, Lyumanshan Ye, Ethan Chern, Yixin Ye, et al. 2024. Olympiarena: Benchmarking multi-discipline cognitive reasoning for superintelligent ai. *arXiv preprint arXiv:2406.12753*.
- Kirby Kuznia, Swaroop Mishra, Mihir Parmar, and Chitta Baral. 2022. Less is more: Summary of long instructions is better for program synthesis. *arXiv preprint arXiv:2203.08597*.
- Miao Li, Eduard Hovy, and Jey Han Lau. 2023. Summarizing multiple documents with conversational structure for meta-review generation. *arXiv preprint arXiv:2305.01498*.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Xiaochen Liu, Yang Gao, Yu Bai, Jiawei Li, Yinan Hu, Heyan Huang, and Boxing Chen. 2022. Psp: Pre-trained soft prompts for few-shot abstractive summarization. *arXiv preprint arXiv:2204.04413*.
- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.
- Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*.
- Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.
- Xiao Pu, Mingqi Gao, and Xiaojun Wan. 2023. Summarization is (almost) dead. *arXiv preprint arXiv:2309.09558*.
- Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.
- Haopeng Zhang, Philip S Yu, and Jiawei Zhang. 2024. A systematic survey of text summarization: From statistical methods to large language models. *arXiv preprint arXiv:2406.11289*.
- Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. [Neural document summarization by jointly learning to score and select sentences](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663, Melbourne, Australia. Association for Computational Linguistics.

CEGRL-TKGR: A Causal Enhanced Graph Representation Learning Framework for Temporal Knowledge Graph Reasoning

Jinze Sun¹, Yongpan Sheng^{1,2}✉, Lirong He³, Yongbin Qin⁴, Ming Liu⁵, Tao Jia¹

¹College of Computer and Information Science, Southwest University, Chongqing 400715, China

²School of Big Data & Software Engineering, Chongqing University, Chongqing 401331, China

³School of Information Science and Engineering, Chongqing Jiaotong University, Chongqing 400074, China

⁴College of Computer Science and Technology, The State Key Laboratory of Public Big Data, Guizhou University

⁵Deakin University, 3125, Victoria, Australia

ssunjinze@outlook.com {shengyp2011,ronghe1217}@gmail.com ybqin@gzu.edu.cn m.liu@deakin.edu.au tjia@swu.edu.cn

Abstract

Temporal knowledge graph reasoning (TKGR) is increasingly gaining attention for its ability to extrapolate new events from historical data, thereby enriching the inherently incomplete temporal knowledge graphs. Existing graph-based representation learning frameworks have made significant strides in developing evolving representations for both entities and relational embeddings. Despite these achievements, there’s a notable tendency in these models to inadvertently learn biased data representations and mine spurious correlations, consequently failing to discern the causal relationships between events. This often leads to incorrect predictions based on these false correlations. To address this, we propose an innovative **Causal Enhanced Graph Representation Learning** framework for TKGR (named CEGRL-TKGR). This framework introduces causal structures in graph-based representation learning to unveil the essential causal relationships between events, ultimately enhancing the performance of the TKGR task. Specifically, we first disentangle the evolutionary representations of entities and relations in a temporal knowledge graph sequence into two distinct components, namely causal representations and confounding representations. Then, drawing on causal intervention theory, we advocate the utilization of causal representations for predictions, aiming to mitigate the effects of erroneous correlations caused by confounding features, thus achieving more robust and accurate predictions. Finally, extensive experimental results on six benchmark datasets demonstrate the superior performance of our model in the link prediction task.

1 Introduction

Knowledge graphs (KGs) have gained significant promise in natural language processing or knowledge engineering perception tasks (Chen et al., 2022a). They model real-world factual knowledge

using multi-relationship graph structures. However, factual knowledge in reality is constantly evolving, resulting in the form of event knowledge. This has led to the development and application of temporal knowledge graphs (TKGs). TKG encodes the relationship information of entities and events and their timing for capturing the dynamics of entities and their relationships over time (Gastinger et al., 2022). Thus, analyzing the TKG provides a comprehensive understanding of the evolving events, based on which various time-dependent applications have been developed, including time-sensitive semantic search (Barbosa et al., 2013), policy making (Deng et al., 2020), stock forecasting (Feng et al., 2019), and more (Chen et al., 2022a).

The reliability of applications depends on accurate predicting, which highly relies on data integrity. However, existing TKGs are inevitably incomplete due to the partial observation of real-world (Liang et al., 2022). To address this limitation and enhance the representation capability of the TKG, temporal knowledge graph reasoning (TKGR) models are proposed and aim to extrapolate new facts and relationships in the TKG according to their historical temporal information. Existing models explore different strategies to achieve satisfactory results on the TKGR task. GHNN (Han et al., 2020) and GHT (Sun et al., 2022) model historical facts as point-in-time processes. TKGR-RHETNE (Sun et al., 2023) jointly models the relevant historical event and temporal neighborhood event context of events in the TKG. RE-NET (Jin et al., 2020) and REGCN (Li et al., 2021) introduce graph neural networks (GNN) into sequence models to capture structural and temporal dependencies between entities. TKGR-GPRSCL (Xiong et al., 2024) captures complex structure-aware information by encoding paths across entities and obtaining temporal correlations in the complex plane. TLogic (Liu et al., 2022) and TITer (Sun et al., 2021) design

interpretable models based on logical rules and reinforcement learning, respectively. Despite the achievements of previous studies, they have overlooked the reality that there are numerous confounding factors in the TKG, such as shallow patterns and noisy links. However, these confounding factors commonly misguide the reasoning process in the TKG, resulting in the acquisition of incorrect dependencies and the generation of non-causal predictions (Sui et al., 2022).

To address the aforementioned issues, we advocate for the integration of causal theory into TKGR to guide learning of the essential causal relationships between events and mitigate the impact of confounding factors on the TKGR task. Specifically, we first construct a structural causal model (Zečević et al., 2021) to comprehensively analyze and model the TKGR task from a causal perspective. Then, based on the causal model, we propose a new framework, namely **Causal Enhanced Graph Representation Learning (CEGRL-TKGR)**, to disentangle confounding factors from the essential causal factors in the TKG. *To the best of our knowledge, this is the first study to incorporate causal intervention in a graph representation learning framework for learning the evolutionary representations of entities and relations in the TKG.* To conclude, our contributions in this paper are 3-folds:

- We propose a novel **Causal Enhanced Graph Representation Learning** framework for **Temporal Knowledge Graph Reasoning**, called **CEGRL-TKGR**, to uncover the essential causal relationships between events and mitigate the impact of confounding factors.
- The proposed **CEGRL-TKGR** framework disentangles the evolutionary representations of entities and relations into causal and confounding representations. Then, it applies causal interventions to perform backdoor adjustments of representations, prioritizing predicted causal features while minimizing the impact of spurious correlations introduced by confounding features.
- Comprehensive experimental results demonstrate that **CEGRL-TKGR** outperforms state-of-the-art baselines on six real-world datasets in the link prediction task. Further, comprehensive studies confirm the contribution of

the introduced causal structures and interventions¹.

2 Related Work

2.1 Temporal Knowledge Graph Reasoning

TKGR in extrapolation settings focuses on predicting new facts about the future based on historical events. Specifically, **CyGNet** (Zhu et al., 2021) uses a copy-generating mechanism to capture the global repetition rate of facts. **GHNN** (Han et al., 2020) and **GHT** (Sun et al., 2022) construct a temporal point process (TPP) to capture the temporal dynamics of successive events, predicting future facts by estimating the conditional probability of the TPP. In recent years, with the successful application of GNN in many dynamic scenarios (Zhang et al., 2022), they have also been introduced into structural-semantic dependency models in TKGR. **RE-NET** (Jin et al., 2020) used a neighborhood aggregator and cyclic event encoder to model historical facts as subgraph sequences. **RE-GCN** (Li et al., 2021) uses **RGCN** (Schlichtkrull et al., 2018) to learn evolutionary representations of entities and relationships at each timestamp. **CEN** (Li et al., 2022a) uses length-aware convolutional neural networks (CNNS) to process evolutionary patterns of different lengths. There are also some studies to solve the TKGR problem through path search. For example, **TLogic** (Liu et al., 2022) completes link prediction tasks based on temporal logic rules learned from temporal knowledge graphs. **TITer** (Sun et al., 2021) proposes a TKG prediction model based on reinforcement learning, which uses time-shaped rewards based on Dirichlet distribution to guide model training. All of the methods discussed above have limitations in modeling entity and relationship representations, in particular ignoring cause-and-effect relationships between different entities, which we believe is key to making correct predictions.

2.2 Causal Representation Learning

In graph causal representation learning, researchers have explored various methods to improve the explanatory power and generalization performance of GNNs. By applying the principles of causal reasoning to graph-structured data, the researchers

¹To illustrate the evaluation of our **CEGRL-TKGR** framework and facilitate further research on this topic, we have made the experimental details and source code of the framework publicly available at <https://github.com/shengyp/CEGRL-TKGR>.

sought to address the challenges GNNs face when dealing with complex systems such as social networks, molecular maps, and syntax trees of program code. DIR (Wu et al., 2022) is proposed to reveal the intrinsic interpretability of GNNs by discovering invariant reasons, which involves splitting input graphs into causal and non-causal fruit graphs and training the two classifiers through invariant risk loss functions. GOOD (Chen et al., 2022b) improves the cross-domain generalization of graphs by distinguishing invariant subgraphs from other parts of graphs that are susceptible to domain transfer. CAL (Sui et al., 2022) introduces de-confounding training to distinguish the key and secondary parts of the graph and eliminate the confounding effect of the secondary parts on model prediction. CFLP (Zhao et al., 2022) points out that the causal relationship between graph structure and link presence is often ignored, and proposed to generate counterfactual links to enhance training data and reduce reliance on false associations. Zevcevic *et al.* (Zečević et al., 2021) theoretically analyze the relationship between GNNs and structural causal models (SCMs) and design a new class of neuro-causal models. However, none of the work has been done to combine causal learning with the TKGR task.

3 Preliminary

3.1 Notations and Task Formulation

A TKG \mathcal{G} can be formalized as a sequence of knowledge graph slices $\{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_T\}$, where $\mathcal{G}_t = \{(e_s, r, e_o, t) \in \mathcal{G}\}$ denotes a knowledge graph slice that consists of facts that occurred at the timestamp t range from t_0 to t_n . Here, e_s and e_o represent the subject and object entities, respectively, and r denotes the predicate as a relation type. Besides, \mathbf{e}_s , \mathbf{r} , \mathbf{e}_o written in bold represent their embeddings. The objective of TKGR task is to predict either the subject in a give query $(?, r, e_o, t)$ or the object in a given query $(e_s, r, ?, t)$ with $t > t_n$.

3.2 A Causal Perspective on the GNN-Based TKGR Task

3.2.1 GNN-based TKGR Paradigm

Inspired by previous GNN-based modeling in a causal look (Didelez and Pigeot, 2001; Sui et al., 2022), we abstract the GNN-based TKGR process through a structural causal figure, as shown in Fig. 1, encompassing five distinct variables. The connectivity from one variable to another epitomizes

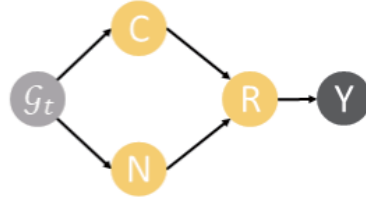


Figure 1: The GNN-based structural causal graph for the TKGR task.

mizes the causal relationship, delineated as the cause \rightarrow effect. The variables are described as follows:

- Graph data \mathcal{G}_t : The knowledge graph at each timestamp t , manifests as a directed multi-relationship figure.
- Causal Feature C : These features epitomize the causal essence of the targeted entity, providing a fundamental understanding of its inherent dynamics.
- Confounding Feature N : These features, discerned from GNN, embody the confounding attributes, unveiling the potential biases or trivial patterns ingrained in graph-based learning methodologies.
- Representation R : These representations are the entity and relational representations of the output of the final GNN layer after learning for \mathcal{G}_t .
- Prediction Y : Denoted as TKGR as the link prediction, this aspect transitions through the decoder, rendering the ultimate reasoning based on the preceding representation.

The causal embedding encapsulates the causal features C , authentically mirroring the implicit knowledge inherent in the knowledge graph \mathcal{G}_t . Conversely, N symbolizes the confounding features, which might be spawned by data biases, data noise, or superficial patterns within graph-based learning methodologies. These confounding features forge a backdoor pathway between C and Y , fostering spurious correlations that don't contribute to accurate reasoning. Functionally, the structural operation denoted by $C \rightarrow R \leftarrow N$ portrays a GNN, wherein both the causal features C and the confounding features N , as discerned by the target entity from the graph data, exert a direct impact on the output R of the GNN. Subsequently,

the output R of GNN directly sways the model inference outcome, illustrated as $R \rightarrow Y$.

In the graph-based TKGR paradigm, causal and confounding features are not decoupled for each entity or relationship embedding. Using causal graphs, we aspire to explicitly separate causal embeddings and confounding embeddings from entity or relational representations, and aim to mitigate the effects of confounding features by performing causal interventions. This endeavor not only clarifies the inference process but also endeavors to refine the accuracy and reliability of the GNN-based TKGR mechanism.

3.3 Causal Intervention Strategies

Beyond fostering a novel comprehension of GNN-based TKGR, causal theory avails analytical instruments predicated on causal figures, such as causal intervention. Causal intervention facilitates a profound examination of the factors precipitating inference outcomes. As delineated by Fig. 1., confounding feature N and causal feature C can be discerned from the knowledge graph \mathcal{G}_t . These features are contemplated in the representation R of entities and relations, thereby establishing a backdoor pathway represented as $N \leftarrow \mathcal{G}_t \rightarrow C \rightarrow R \rightarrow Y$, with N serving as the quick bridge between C and Y .

To orchestrate a causal prognosis hinging on the causal feature C , it necessitates the modeling of $P(Y | C)$. However, the backdoor path distorts the probability distribution $P(Y | C)$ through the confounding effect of N , thereby necessitating the disentanglement of the backdoor pathway from N to Y . It is imperative to stymie this backdoor pathway to mitigate the repercussions of the hybrid embedding, thereby enabling the model to reason robustly by leveraging the causal feature to the fullest. Causality theory is a potent toolkit to address this backdoor path dilemma.

We engage the do-calculus for executing causal interventions on variable C , intending to sever the backdoor path $N \leftarrow \mathcal{G}_t \rightarrow C \rightarrow R \rightarrow Y$. Our objective is to estimate $P(Y | do(C))$, as opposed to muddling it with $P(Y | C)$. By using Bayes' theorem with the causal postulation, we can extrapolate the ensuing expression:

$$P(Y | do(C)) = \sum_{n \in N} P(Y | C, n)P(n). \quad (1)$$

The equation above illustrates that to gauge the causal influence of C on Y , it's requisite to take

into account the inference outcomes of both causal and confounding features. This can be perceived as re-coupling the disentanglement feature embeddings, utilizing them for deductive reasoning at future timestamps. However, C and N are usually unobservable, and it is difficult to obtain them directly at the data level, which makes the calculation of the Eq. (1) very challenging. In the next section, we discuss ways to overcome this problem.

4 The Proposed CEGRL-TKGR Framework

4.1 The Overall Architecture of TKGR-GPRSCL

We detail the CEGRL-TKGR framework for learning representations of entities and relationships based on causal features and confounding features. CEGRL-TKGR consists of three parts: (1) The representation learning part that learns the structure dependence in each \mathcal{G}_t ; (2) The decoupling learning part that learns the entity and relation representations; (3) The decoder part that is designed based on the time interval. The overall architecture of the framework is shown in Fig. 2.

4.2 Entity and Relation Evolution Representation

Within each \mathcal{G}_t , representation learning of entities and relationships involves the aggregation of multiple relationships, as well as information from multiple hop neighbors under a single timestamp. Between adjacent \mathcal{G}_t , we expect to accurately capture the order dependencies inherent in the subgraph with different timestamps. Drawing inspiration from the RE-GCN model (Li et al., 2021), we employ the ω -layer RGCN, which hinges on structure modeling and a recurrent mechanism to progressively update the representations of entities and relations. This approach allows for a more nuanced understanding and modeling of the dynamic interactions within the graph over time.

$$\mathbf{e}_{o,t}^{l+1} = \text{RReLU} \left(\sum_{(e_s, r, e_o) \in \mathcal{G}_t} \frac{1}{d_{e_o}} \mathbf{W}_1^l \left(\Phi \left(\mathbf{e}_{s,t}^l, \mathbf{r}_t \right) \right) + \mathbf{W}_2^l \mathbf{e}_{o,t}^l \right), \quad (2)$$

$$\mathbf{E}_t = \text{GRU} \left(\mathbf{E}_{t-1}, \mathbf{E}_t' \right). \quad (3)$$

In the Eq. (2), we describe how the embedding $\mathbf{e}_{o,t}^{l+1}$ of entity e_o at time step t and layer $l+1$ is computed. We integrate the information of all entities

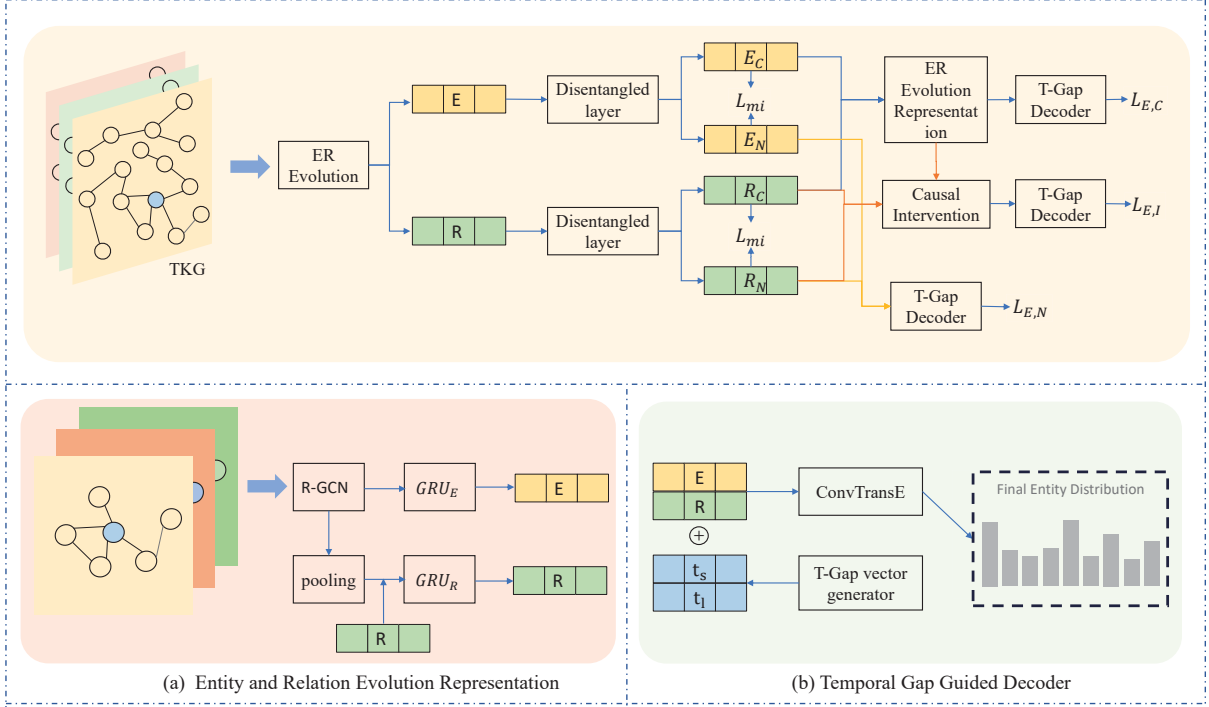


Figure 2: The overall architecture of our proposed CEGRL-TKGR framework.

and relations connected to entity e_o in the knowledge graph \mathcal{G}_t . $\mathbf{W}_1^l, \mathbf{W}_2^l$ is learnable weights and Φ has the option of addition or one-dimensional convolution. In the Eq. (3), we showcase how the entity embedding matrix \mathbf{E}_t is updated via the GRU. Specifically, we take the entity embedding matrix \mathbf{E}_{t-1} at the previous time step $t-1$ and the aggregated entity embedding matrix \mathbf{E}'_t as inputs to obtain the entity embedding matrix \mathbf{E}_t at the current time step t .

For relations, ensuring consistency with the entity embedding updates within the subgraph sequence is crucial. To achieve this consistency, a specialized GRU tailored for relations is employed for the update process. This mechanism facilitates a harmonized evolution of both entity and relation causal embeddings over the sequence of subgraphs:

$$\mathbf{r}'_t = \text{pooling}(\mathbf{E}_{t-1}, R_t) \oplus \mathbf{r}, \quad (4)$$

$$\mathbf{R}_t = \text{GRU}(\mathbf{R}_{t-1}, \mathbf{R}'_t), \quad (5)$$

where \mathbf{r}'_t is an aggregation of all entities connected to relation r via a mean pooling operation, and \mathbf{R}'_t is obtained by concatenating this result with the embeddings of all relations. Eventually, we update the relation embedding matrix \mathbf{R}_t using a GRU.

4.3 Disentangled Causal and Confounding Features

In the previous subsection, the entity and relation representations are learned based on GNN-

contained causal and confounding factors, and we separate them at the presentation level, which provides a solution to the previously mentioned problem of not being able to separate these two features at the data level. To do this, we introduce a decoupling module to decouple causal and confounding features. Taking the entity embedding matrix as an example, it is represented as follows:

$$\mathbf{D}_{E,C}, \mathbf{D}_{E,N} = \text{softmax}(\text{MLP}(\mathbf{E})), \quad (6)$$

$$\mathbf{E}_C = \mathbf{E} \odot \mathbf{D}_{E,C}, \mathbf{E}_N = \mathbf{E} \odot \mathbf{D}_{E,N}. \quad (7)$$

We want the two embeddings learned from the decoupling module to be as independent as possible, which is essential to accurately separate causal and confounding features (Chen et al., 2023). Mutual information is a basic quantity to measure the nonlinear correlation of two random variables. Minimizing mutual information is a feasible scheme to decouple causal features from confounding features. Specifically, we implement this process with contrastive log-ratio upper-bound MI estimator (Cheng et al., 2020; Wu et al., 2021), which utilizes variational distributions q and a neural network to approximate the true distribution. We define the objective function as follows:

$$\mathcal{L}_{mi} = \mathbb{E}_{p(\mathbf{E}_C, \mathbf{E}_N)} [\log q_\theta(\mathbf{E}_N | \mathbf{E}_C)] - \mathbb{E}_{p(\mathbf{E}_C)} \mathbb{E}_{p(\mathbf{E}_N)} [\log q_\theta(\mathbf{E}_N | \mathbf{E}'_C)]. \quad (8)$$

We perform the same operation with relation

embedding decoupling, after which we obtain \mathbf{R}_C and \mathbf{R}_N .

4.4 Temporal Gap Guided Decoder

After the causal and confounding embeddings of entities and relations in the derived data, we use a specially crafted decoder to determine the likelihood score of potential entities and relations. Events or facts in a data stream may span different periods. For example, major political events may occur in rapid succession over a short period, while certain rare natural phenomena may occur sporadically and at longer intervals. With this in mind, it is reasonable to consider the time intervals of events to get an accurate picture of their temporal relationship. The key to the design of our decoder is the time interval vector, which guides the decoding process in considering the event time interval while calculating the fraction. Formulaic as:

$$\mathbf{t}_s = \alpha_s t + \beta_s, \mathbf{t}_l = \alpha_l t + \beta_l. \quad (9)$$

Here, $\alpha_s, \beta_s, \alpha_l$, and β_l signify learnable parameters. Adopting ConvTransE as our decoder, we introduce four variables, which traverse a one-dimensional convolutional layer followed by a fully connected layer, culminating in the extraction of a probability vector encompassing all entities. This process is mathematically articulated as:

$$\mathbf{p}_C(e_o | e_s, r, t) = \text{ReLU} \left(\text{ConvTransE}(\mathbf{e}_{s,C,t}, \mathbf{r}_{C,t}, \mathbf{t}_s, \mathbf{t}_l) \right) \mathbf{E}_{C,t}. \quad (10)$$

We apply the same decoding process to the confounding features to get $\mathbf{p}_N(e_o | e_s, r, t)$.

4.5 Causal Intervention and Training Objective

Causal-based embedding learns the intrinsic causes that cause events to occur, so the reasoning results obtained from causal-based embedding are expected to yield reasonable input results. We define the supervised classification loss as follows:

$$\mathcal{L}_{E,C} = \sum_{(e_s, r, e_o, t) \in \mathcal{G}} \mathbf{y}_t \log \mathbf{p}_C(e_o | e_s, r, t), \quad (11)$$

where \mathbf{y}_t is label vector. Conversely, confounding features are conceptualized to address conceivable biases or superficial patterns emanating from the training dataset. Given their inability to aid in inference, we proceed to compute their output average

across all entity categories and encapsulate the loss as:

$$\mathcal{L}_{E,N} = \frac{1}{|\mathbf{E}_{N,t}|} \sum_{(e_s, r, e_o, t) \in \mathcal{G}} \text{KL} \left(\mathbf{y}_u, \log \mathbf{p}_N(e_o | e_s, r, t) \right), \quad (12)$$

where KL denotes the KL-Divergence, \mathbf{y}_u represents the uniform distribution.

We believe that causal intervention is the manifestation of causal features under the influence of confounding features, but we cannot directly conduct causal intervention at the data level to mitigate confounding effects. Therefore, we obtain intervention features that combine causal features and confounding features at the representation level of entities and relationships. Specifically, according to the backdoor adjustment Eq. (1), we first introduce a random addition procedure to obtain the intervention feature, and for the intervention feature we expect the decoder to still output the correct result:

$$\mathbf{E}_{I,t} = \phi(\mathbf{E}_{C,t}, \mathbf{E}'_{N,t}), \quad (13)$$

$$\mathbf{p}_I(e_o | e_s, r, t) = \text{ReLU} \left(\text{ConvTransE}(\mathbf{e}_{s,I,t}, \mathbf{r}_{I,t}, \mathbf{t}_s, \mathbf{t}_l) \right) \mathbf{E}_{I,t}, \quad (14)$$

where $\mathbf{E}'_{N,t}$ is the confounding feature of the entities randomly sampled from $\mathbf{E}_{N,t}$. Then we define the loss as follows:

$$\mathcal{L}_{E,I} = \sum_{(e_s, r, e_o, t) \in \mathcal{G}} \mathbf{y}_t \log \mathbf{p}_I(e_o | e_s, r, t). \quad (15)$$

Finally, the loss function of the model for the link prediction task is as follows:

$$\mathcal{L}_E = \mathcal{L}_{E,C} + \lambda_1 \mathcal{L}_{E,N} + \lambda_2 \mathcal{L}_{mi} + \lambda_3 \mathcal{L}_{E,I}, \quad (16)$$

where $\lambda_1, \lambda_2, \lambda_3$ are designated as hyperparameters, and the first two are used to determine the strength of decoupled learning of the model and the latter is used to determine the strength of causal intervention of the model.

5 Experiments and Analysis

5.1 Experimental Settings and Implementation Details

Datasets. We evaluate our model and baselines on six benchmark datasets, including ICEWS14 (Garcia-Duran et al., 2018),

ICEWS18 (Jin et al., 2019), ICEWS05-15 (Garcia-Duran et al., 2018), YAGO (Mahdisoltani et al., 2014), WIKI (Leblay and Chekol, 2018) and GDELT (Leetaru and Schrod, 2013). Statistics of the datasets are summarized in Table 1.

Table 1: Statistics of datasets in the experiments.

Dataset	# Entity	# Predict	# Train	# Valid	# Test	Time interval
ICEWS14	7128	230	63685	13823	13222	24 hours
ICEWS18	23033	256	373018	45995	49545	24 hours
ICEWS05-15	10488	251	322958	69224	69147	24 hours
YAGO	10623	10	161540	19523	20026	1 year
WIKI	12554	24	539286	67583	63110	1 year
GDELT	7691	240	1734399	238765	305241	15 mins

Baselines. For the link prediction task, we compare CEGRL-TKGR model with two categories of KGR models: (1) *static KGR models*, including TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016) and R-GCN (Schlichtkrull et al., 2018). We apply these models in static KGs that ignore timestamp information. (2) *TKGR models*, including TTransE (Leblay and Chekol, 2018), TA-DistMult (Garcia-Duran et al., 2018), TNTComplEx (Lacroix et al., 2019), RE-GCN (Li et al., 2021), GHT (Sun et al., 2022), EvoKG (Park et al., 2022), TITer (Sun et al., 2021), xERTE (Han et al., 2021), TLogic (Liu et al., 2022) and CEN (Li et al., 2022b).

Evaluation Metrics. The mean reciprocal rank (MRR) and Hits@k are standard metrics for the TKG link prediction task. MRR is the average reciprocal of the correct query answer rank. Hits@k indicates the proportion of correct answers among the top k candidates. We use a more reasonable time-aware filter setting to report our experimental results².

Implementation Details. The whole of training hyper-parameters and model configurations are summarized in Appendix A.1.

5.2 Experimental Results and Discussion

Table 2 and Table 3 report the experimental results of the link prediction task on six benchmark datasets. Static KG embedding methods fell far behind CEGRL-TKGR due to their inability to capture temporal dynamics in the TKG. Our method is also superior to other TKGR models in predicting events. The improved performance shows that surface patterns and noise are widely present in several real-world TKG datasets. The previous models

²The time-aware filtering setting filters out only the four groups that occur at query time and can simulate extrapolated prediction tasks in the real world (Sun et al., 2021).

are generally inadequate in design. CEGRL-TKGR based on evolutionary representation will learn the inherent confounding features in the TKG when gathering neighborhood information and transmitting historical information, and the model based on rule-based inference will mine the false correlation in the data, all of which will lead to the model-making non-causal predictions in the reasoning stage. Our model incorporates causal theory into the TKGR task and visibly separates causal features from confounding features. This helps to protect the model from surface patterns and noise present in the dataset and to uncover the real reasons that affect the formation of links between entities. TiTer and EvoKG show excellent performance on YAGO datasets because the former’s historical fact search strategy works well on smaller datasets, while the latter’s modeling of event timing works well on datasets containing events at relatively regular time intervals. More model configurations and experimental results are summarized in the Appendix.

6 Conclusion

In this paper, we revisit the GNN-based TKGR model from the causality perspective, on this basis, we propose a novel CEGRL-TKGR framework. By synergistically integrating causal structures with graph representation learning of the TKG, our framework overcomes the problem of existing TKGR models’ learning biased data representations and mining for false correlations unintentionally. Comprehensive experimental results and analysis have proved the effectiveness of CEGRL-TKGR.

Limitations and Future Work. The proposed CEGRL-TKGR is an innovational causal enhanced graph representation learning framework for optimizing feature representations directly using causal technology for the TGKR task. The limitations of CEGRL-TKGR are as follows:

- From the dataset’s perspective, our research primarily focuses on TKG datasets, which may not verify the generalization ability of the CEGRL-TKGR framework to those time-interval insensitive graph datasets. Additionally, we aim to further conduct case studies to enhance the interpretability of the framework in the reasoning procedure as future work.
- From the model’s perspective, our research evaluates the TKGR task alone. Theoretically,

Table 2: Experimental results of link prediction on ICEWS series dataset. The best result in each column is boldfaced.

Model	ICEWS14				ICEWS18				ICEWS05-15			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TransE	22.48	13.36	25.63	41.23	12.24	5.84	12.81	25.10	22.55	13.05	25.61	42.05
Distmult	27.67	18.16	31.15	46.96	10.17	4.52	10.33	21.25	28.73	19.33	32.19	47.54
ComplEx	30.84	21.51	34.48	49.59	21.01	11.87	23.47	39.97	31.69	21.44	35.74	52.04
R-GCN	28.03	19.42	31.95	44.833	15.05	8.31	16.49	29.00	27.13	18.83	30.41	43.16
TTransE	13.43	3.11	17.32	34.55	8.31	1.92	8.56	21.89	15.71	5.00	19.72	38.02
TA-DistMult	26.47	17.09	30.22	45.41	16.75	8.61	18.41	33.59	24.31	14.58	27.92	44.21
TNTComplEx	32.12	23.35	36.03	49.13	21.23	13.28	24.02	36.91	27.54	19.52	30.80	42.86
Evo-KG	26.90	16.69	30.57	47.39	25.46	16.25	29.15	43.21	26.32	15.82	31.96	50.80
xERTE	40.79	32.70	45.67	57.30	29.31	21.03	33.51	46.48	46.62	37.84	52.31	63.92
TITer	40.59	31.41	45.47	57.62	29.55	21.37	33.10	44.87	46.62	36.46	52.29	65.23
TLogic	41.80	31.93	47.23	60.53	28.41	18.74	32.71	47.97	45.99	34.49	52.89	67.39
RE-GCN	42.00	31.63	47.20	61.65	32.62	22.39	36.79	52.68	48.03	37.33	53.90	68.51
CEN	41.93	31.71	46.86	61.36	29.41	19.60	33.91	49.97	47.04	36.58	52.60	67.18
GHT	38.28	28.43	42.85	57.47	28.38	18.78	32.01	47.27	42.90	31.76	46.77	64.64
CEGRL-TKGR	42.74	32.32	48.05	62.68	32.90	22.70	36.91	52.95	48.35	37.63	54.22	68.47

Table 3: Experimental results of link prediction on YAGO, WIKI, and GDELTA datasets. The best result in each column is boldfaced.

Model	YAGO				WIKI				GDELTA			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TransE	38.97	26.87	42.45	56.05	23.46	16.53	28.45	35.71	-	-	-	-
Distmult	44.05	39.19	49.70	59.94	27.96	18.84	32.45	39.51	8.61	3.91	8.27	17.04
ComplEx	44.09	39.33	49.57	59.64	27.69	18.67	31.99	38.61	9.84	5.17	9.58	18.23
R-GCN	20.25	11.25	24.01	37.30	13.96	7.21	15.75	22.05	12.17	8.64	12.37	20.63
TTransE	31.19	18.12	40.91	51.21	29.27	21.67	34.43	42.39	-	-	-	-
TA-DistMult	54.92	48.15	59.61	66.71	44.53	39.92	48.73	51.71	10.34	6.25	10.44	21.63
TNTComplEx	57.98	52.92	61.33	66.69	45.03	40.04	49.31	52.03	19.53	12.41	20.75	33.42
Evo-KG	68.81	54.49	81.40	92.41	67.44	54.63	79.36	85.98	18.94	11.31	20.08	34.01
GHT	57.22	51.64	60.68	67.17	48.50	45.08	50.87	53.69	20.04	12.68	21.37	34.42
xERTE	84.19	80.09	88.02	89.78	73.60	69.05	78.03	79.73	19.45	11.92	20.84	34.18
TITer	87.47	80.09	89.96	90.27	73.91	71.70	75.41	76.96	18.19	11.52	19.20	31.00
RE-GCN	82.30	78.83	84.27	88.58	78.53	74.50	81.59	84.70	19.69	12.46	20.93	33.81
CEN	83.49	79.66	86.10	90.04	78.52	74.65	81.44	84.59	19.96	11.39	20.97	33.77
CEGRL-TKGR	86.25	82.92	88.72	91.70	79.66	75.73	82.83	85.59	20.11	12.73	21.46	34.51

cally, the GNN-based reasoning paradigm incorporated in the causal structure can be applied to any other graph representation learning tasks, such as triple classification (Jaradeh et al., 2021), triple set prediction (Zhang et al., 2024), and graph classification (Liu et al., 2023). In future work, we desire to explore powerful disentanglement methods and more advanced causal intervention strategies to improve the CEGRL-TKGR’s performance for more rich graph representation learning-based tasks. Besides, the increased complexity of causal reasoning in the TKG is untouched.

- From the adaptation’s perspective, to adapt the CEGRL-TKGR framework to more models, there are some hyper-parameters to control causal intervention and training. These hyper-parameters are sensitive to different models and datasets, hence it needs to take sufficient time to experiment to find the optimal values and combinations among them. Therefore, how to reduce the consumption in the above adaptation procedure upon the framework is

worthy of consideration.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (Grant No. 62202075, No. 62171111, No. 62376043), the Natural Science Foundation of Chongqing, China (No. CSTB2022NSCQ-MSX1404, No. CSTB2023NSCQ-MSX0091), Fundamental Research Funds for the Central Universities (No. SWU-KR24008), Science Research Project, Chongqing College of International Business and Economics (No. KYZK2024001), Key Laboratory of Data Science and Smart Education, Hainan Normal University, Ministry of Education (No. DSIE202206), and the State Key Laboratory of Public Big Data, Guizhou University (No. PBD2024-0501), Ming Liu’s research was funded in part by Australian Research Council Linkage (LP220200746). Yongbin Qin’s work was supported by the National Natural Science Foundation of China (No. 62166007, No. 62066008) and the National Key R&D Program of China (No. 2023YFC3304500).

References

- Denilson Barbosa, Haixun Wang, and Cong Yu. 2013. Shallow information extraction for the knowledge web. In *Proceedings of IEEE 29th International Conference on Data Engineering*, pages 1264–1267.
- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*, pages 2787–2795.
- Guoxin Chen, Yongqing Wang, Fangda Guo, Qinglang Guo, Jiangli Shao, Huawei Shen, and Xueqi Cheng. 2023. Causality and independence enhancement for biased node classification. In *Proceedings of the 32th ACM International Conference on Information and Knowledge Management*, pages 203–212.
- Yonghong Chen, Hao Li, Han Li, Wenhao Liu, Yirui Wu, Qian Huang, and Shaohua Wan. 2022a. An overview of knowledge graph reasoning: key technologies and applications. *Journal of Sensor and Actuator Networks*, 11(4):78.
- Yongqiang Chen, Yonggang Zhang, Yatao Bian, Han Yang, MA KAILI, Binghui Xie, Tongliang Liu, Bo Han, and James Cheng. 2022b. Invariance principle meets out-of-distribution generalization on graphs. In *ICML 2022: Workshop on Spurious Correlations, Invariance and Stability*.
- Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. 2020. Club: A contrastive log-ratio upper bound of mutual information. In *International conference on machine learning*, pages 1779–1788. PMLR.
- Songgaojun Deng, Huzefa Rangwala, and Yue Ning. 2020. Dynamic knowledge graph based multi-event forecasting. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1585–1595.
- Vanessa Didelez and Iris Pigeot. 2001. Causality: models, reasoning, and inference.
- Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. 2019. Temporal relational ranking for stock prediction. *ACM Transactions on Information Systems (TOIS)*, 37(2):1–30.
- Alberto Garcia-Duran, Sebastijan Dumančić, and Mathias Niepert. 2018. Learning sequence encoders for temporal knowledge graph completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4816–4821.
- Julia Gastinger, Timo Sztyler, Lokesh Sharma, and Anett Schuelke. 2022. On the evaluation of methods for temporal knowledge graph forecasting. In *NeurIPS 2022 Temporal Graph Learning Workshop*.
- Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2021. xerte: Explainable reasoning on temporal knowledge graphs for forecasting future links. In *Proceedings of the 9th International Conference on Learning Representations*.
- Zhen Han, Yunpu Ma, Yuyi Wang, Stephan Günnemann, and Volker Tresp. 2020. Graph hawkes neural network for forecasting on temporal knowledge graphs. In *Proceedings of the 2nd Conference on Automated Knowledge Base Construction*.
- Mohamad Yaser Jaradeh, Kuldeep Singh, Markus Stocker, and Sören Auer. 2021. Triple classification for scholarly knowledge graph completion. In *Proceedings of the 11th Knowledge Capture Conference*, pages 225–232.
- Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2019. Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. <https://arxiv.org/abs/1904.05530>.
- Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020. Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6669–6683.
- Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2019. Tensor decompositions for temporal knowledge base completion. In *International Conference on Learning Representations*.
- Julien Leblay and Melisachew Wudage Chekol. 2018. Deriving validity time in knowledge graph. In *The Web Conference*, pages 1771–1776.
- Kalev Leetaru and Philip A Schrod. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer.
- Zixuan Li, Saiping Guan, Xiaolong Jin, Weihua Peng, Yajuan Lyu, Yong Zhu, Long Bai, Wei Li, Jiafeng Guo, and Xueqi Cheng. 2022a. Complex evolutionary pattern learning for temporal knowledge graph reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 290–296.
- Zixuan Li, Saiping Guan, Xiaolong Jin, Weihua Peng, Yajuan Lyu, Yong Zhu, Long Bai, Wei Li, Jiafeng Guo, and Xueqi Cheng. 2022b. Complex evolutionary pattern learning for temporal knowledge graph reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 290–296.
- Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021. Temporal knowledge graph reasoning based on evolutionary representation learning. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 408–417.

- Ke Liang, Lingyuan Meng, Meng Liu, Yue Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, Xinwang Liu, and Fuchun Sun. 2022. Reasoning over different types of knowledge graphs: Static, temporal and multi-modal. <https://arxiv.org/abs/2212.05767>.
- Xingyu Liu, Juan Chen, and Quan Wen. 2023. A survey on graph classification and link prediction based on gnn. <https://arxiv.org/abs/2307.00865>.
- Yushan Liu, Yunpu Ma, Marcel Hildebrandt, Mitchell Joblin, and Volker Tresp. 2022. Tlogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs. In *Proceedings of AAAI*, volume 36, pages 4120–4127.
- Farzaneh Mahdisoltani, Joanna Biega, and Fabian Suchanek. 2014. Yago3: A knowledge base from multilingual wikipedias. In *7th biennial conference on innovative data systems research*. CIDR Conference.
- Namyong Park, Fuchen Liu, Purvanshi Mehta, Dana Cristofor, Christos Faloutsos, and Yuxiao Dong. 2022. Evokg: Jointly modeling event time and network structure for reasoning over temporal knowledge graphs. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 794–803.
- Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web: 15th International Conference, ESWC*, pages 593–607. Springer.
- Yongduo Sui, Xiang Wang, Jiancan Wu, Min Lin, Xiangnan He, and Tat-Seng Chua. 2022. Causal attention for interpretable and generalizable graph classification. In *Proceedings of the 28th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1696–1705.
- Haohai Sun, Shangyi Geng, Jialun Zhong, Han Hu, and Kun He. 2022. Graph hawkes transformer for extrapolated reasoning on temporal knowledge graphs. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7481–7493.
- Haohai Sun, Jialun Zhong, Yunpu Ma, Zhen Han, and Kun He. 2021. Timetraveler: Reinforcement learning for temporal knowledge graph forecasting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8306–8319.
- Jinze Sun, Yongpan Sheng, Ling Zhan, and Lirong He. 2023. Tkgr-rhetne: A new temporal knowledge graph reasoning model via jointly modeling relevant historical event and temporal neighborhood event context. In *International Conference on Neural Information Processing*, pages 331–343. Springer.
- Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 2071–2080.
- Junkang Wu, Wentao Shi, Xuezhi Cao, Jiawei Chen, Wenqiang Lei, Fuzheng Zhang, Wei Wu, and Xiangnan He. 2021. Disenkgat: knowledge graph embedding with disentangled graph attention network. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, pages 2140–2149.
- Yingxin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. 2022. Discovering invariant rationales for graph neural networks. In *Proceedings of the 10th International Conference on Learning Representations*.
- Lizhu Xiong, Yongpan Sheng, and Lirong He. 2024. Tkgr-gprscf: Enhance temporal knowledge graph reasoning with graph structure-aware path representation and supervised contrastive learning. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 200–212. Springer.
- Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the 3rd International Conference on Learning Representations*.
- Matej Zečević, Devendra Singh Dhami, Petar Veličković, and Kristian Kersting. 2021. Relating graph neural networks to structural causal models. <https://arxiv.org/abs/2109.04173>.
- Mengqi Zhang, Shu Wu, Xueli Yu, Qiang Liu, and Liang Wang. 2022. Dynamic graph neural networks for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4741–4753.
- Wen Zhang, Yajing Xu, Peng Ye, Zhiwei Huang, Zezhong Xu, Jiaoyan Chen, Jeff Z Pan, and Huajun Chen. 2024. Start from zero: Triple set prediction for automatic knowledge graph completion. *IEEE Transactions on Knowledge and Data Engineering*.
- Tong Zhao, Gang Liu, Daheng Wang, Wenhao Yu, and Meng Jiang. 2022. Learning from counterfactual links for link prediction. In *Proceedings of the 39th International Conference on Machine Learning*, pages 26911–26926. PMLR.
- Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhang. 2021. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, volume 35, pages 4732–4740.

A Appendix

A.1 Implementation Details

We set the dimension of all embeddings and hidden states to 200. The number of layers of the R-GCN is set to 1 for YAGO and 2 for the other datasets. The optimal number of historical snapshots is set to 8, 10, 10, 1, 2, and 6 for ICEWS14, ICEWS18, ICEWS05-15, YAGO, WIKI, and GDELT, respectively. To fair comparison, static graph constraints are added for ICEWS14, ICEWS18, and ICEWS05-15. The channel number for decoding is set to 50, and the kernel size is set to 4×3 . We try several different values for λ_1 , λ_2 , and λ_3 , and finally chose 0.5, 0.5, 0.3. We use Adam to optimize the parameters, with a learning rate of 0.001. All of the experiments are processed on a Linux server with CPU Xeon Gold 6142, RAM 64G, and Nvidia 4090 GPU.

A.2 Ablation Study

We investigate the effectiveness of causally enhanced and time-interval guided decoders for the link prediction task. Specifically, CEGRL-TKGR w/o TD means that no time interval vector is used to guide the decoder to work, and CEGRL-TKGR w/o CE means that the model removes causal decoupling and causal intervention parts. Table 4 shows the results of ablation experiments, which indicate the effectiveness of these two components. As can be seen from the results in the table, for datasets such as YAGO and WIKI that contain relatively regular time intervals, a temporal gap-guided decoder can capture this time interval pattern well enough to make accurate predictions. At the same time, it does not degrade performance even for time-interval insensitive datasets. Our causal enhancement module, under the independent constraint of emphasizing causal features and confounding features, eliminates the influence of the fast bridge through causal intervention, forcing the model to learn the intrinsic causes of the events. It is worth noting that our causal enhancement module can be seen as a flexible component that can be easily used in several GNN-based reasoning frameworks.

A.3 Parameter Sensitivity Analysis

In the CEGRL-TKGR, λ_1 and λ_2 jointly affect the disentanglement intensity of causal and confounding features, and λ_3 controls the intensity of causal intervention. We study the sensitivity of parameters in different benchmark datasets, as depicted

in Fig. 3. Specifically, one parameter is fixed at 0.5 and the other parameter varies in $[0,1]$ with a step size of 0.1. The model is relatively stable in most parameter selection cases, but on noisy datasets, the model has higher requirements for hyper-parameters, and extreme values will degrade the performance of the model. The best range for λ_1 , λ_2 is about 0.5 to 0.7. λ_3 should be a relatively small value, ranging from 0.3 to 0.6.

A.4 Performance on Noisy Temporal Knowledge Graphs

To explore whether the proposed CEGRL-TKGR can alleviate noise and surface patterns, we randomly replace a certain percentage of positive triples in the training set of each TKG dataset in form of noisy TKGs. Taking YAGO and WIKI datasets as examples, we test the performance of CEGRL-TKGR and CEGRL-TKGR w/o CE under different noise deviations, respectively. The experimental results are shown in Fig. 4.

From the experimental results, we can draw the following conclusion: when the noise in the dataset increases, the performance of models lacking the recognition of causal features and confounding features will deteriorate sharply, and the performance of MRR and Hits@1 will decrease, which indicates that the CEGRL-TKGR w/o CE is easy to capture data bias and make wrong predictions based on it. In contrast, CEGRL-TKGR uses the causal enhancement module to effectively reduce the impact of confounding features and shows more stable performance on these two noisy TKG datasets. The performance degradations on MRR and Hits@1 are significantly smaller than those without the causal module.

Table 4: The ablation study of our model on the six benchmark datasets. "w/o" means "without".

Model	ICEWS14		ICEWS18		ICEWS05-15		YAGO		WIKI		GDELT	
	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10
CEGRL-TKGR w/o TD	42.21	62.43	32.67	52.68	48.13	68.33	84.71	90.56	78.54	84.37	19.93	34.50
CEGRL-TKGR w/o CE	41.89	61.65	32.62	52.54	48.03	68.20	81.93	88.39	79.04	84.79	19.66	33.71
CEGRL-TKGR	42.74	62.68	32.90	52.95	48.35	68.47	86.25	91.70	79.66	85.59	20.11	34.51

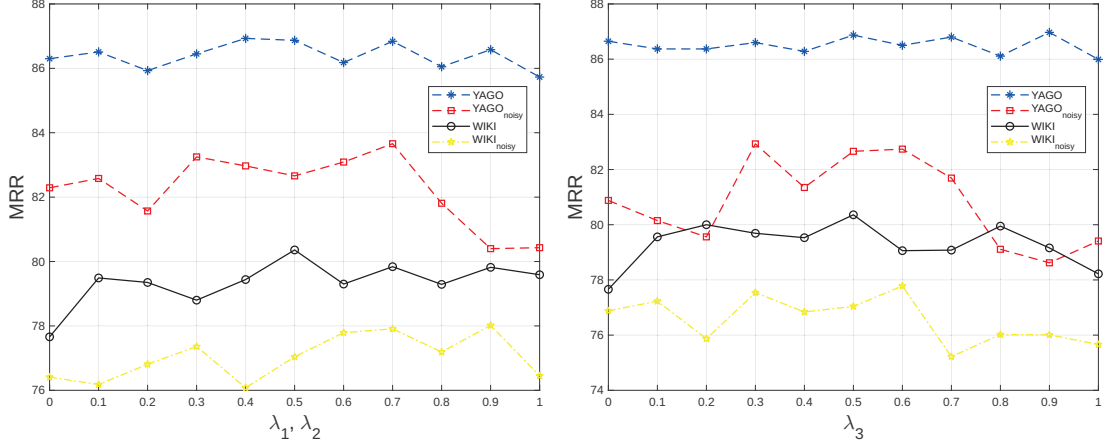
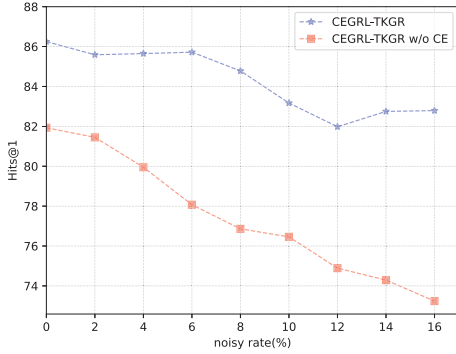
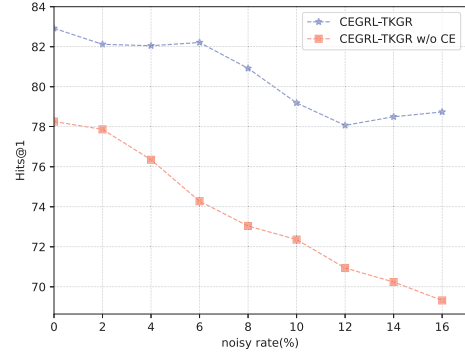


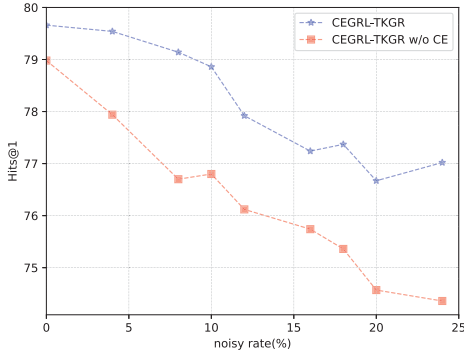
Figure 3: The parameters sensitivity analysis of loss coefficients λ_1, λ_2 and λ_3 .



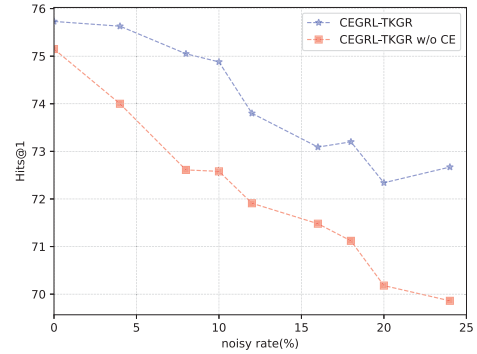
(a) MRR results on the YAGO dataset.



(b) Hits@1 results on the YAGO dataset.



(c) MRR results on the WIKI dataset.



(d) Hits@1 results on the WIKI dataset.

Figure 4: The performance of CEGRL-TKGR and CEGRL-TKGR w/o CE on the noisy YAGO and WIKI datasets, respectively.

Reasoning Knowledge Filter for Logical Table-to-Text Generation

Yu Bai^{1,4}, Baoqiang Liu¹, Shuang Xue², Fang Cai³, Na Ye^{1,4}, Guiping Zhang^{1,4}

¹School of Computer Science, Shenyang Aerospace University

²School of Foreign Language, Shenyang Aerospace University

³Department of Statistics, Stanford University

⁴National and Local Joint Engineering Laboratory for Multilingual Collaborative Translation Technology

Correspondence: {baiyu, liubaoqiang}@sau.edu.cn

Abstract

Logical table-to-text generation (LT2T) seeks to produce logically faithful textual descriptions base on tables. Current end-to-end LT2T models, which use descriptions directly as learning objectives, frequently face challenges in maintaining logical faithfulness due to the lack of a reasoning knowledge. Recent research have introduced reasoning knowledge generated by models for LT2T task, but the noise along with it limited its performance. We therefore propose a framework *reasoning knowledge filter* that leverages the collaboration between large language models and smaller models to filter data points with high-quality reasoning knowledge. This framework aims to provide highly matched table, description and reasoning knowledge triplets for LT2T. The results obtained on LogicNLG database demonstrate that the efficiencies of the method in this paper has achieved optimal performance with a reduced amount of data. Specifically, it enhances SP-Acc by 1.4 points and NLI-Acc by 0.7 points compared to the current state-of-the-art model.

1 Introduction

The objective of the table-to-text generation task is to generate a sentence or a paragraph based on a table. In recent years, it has been an important research direction in natural language processing (Lebret et al., 2016; Wiseman et al., 2017; Parikh et al., 2020; Cheng et al., 2021).

However, previous work is still limited to surface-level descriptions that simply use language to describe the basic facts about the tables (Parikh et al., 2020; Wiseman et al., 2017; Novikova et al., 2017; Lebret et al., 2016). As shown in Figure 1, a surface-level description of the table might be “The attendance for the game held at the Wachovia Center on December 2nd was 19,227”. Chen et al. (2020a) argue that new research should go beyond surface-level descriptions. To this end, they proposed a new task called logical table-to-text gen-

2008 - 09 tampa bay lightning season

date	opponent	location	...	attendance
december 2	...	wachovia center	...	19227
december 4	...	st pete times forum	...	15598
december 6	...	st pete times forum	...	17154
...
december 18	...	st pete times forum	...	16333
...
december 27	...	st pete times forum	...	18226

description : 4 game were played at the St Pete Time Forum.

Figure 1: Example for logicNLG dataset.

eration (LT2T) with a dataset named LogicNLG. In LT2T, the descriptions generated from tables are no longer surface-level, instead, they are required to be logically faithful to the tables. The facts included in such descriptions require multi-step reasoning based on the table. For example, the description in Figure 1 requires the following steps of reasoning: (1) Select all rows with the “location” column value as “st pete times forum”. (2) Count the resulting rows to get “4”.

Since LogicNLG only provides tables and descriptions, most research focuses on modifying model architectures or training objectives to ultimately achieve a model that can outperform end-to-end training (Chen et al., 2021; Nan et al., 2022; Zhao et al., 2023b). These models need to implicitly learn how to generate logically faithful descriptions by reasoning on tables.

A recent study RKT (Liu et al., 2024) suggests that incorporating explicit reasoning knowledge into LogicNLG can enhance the performance of existing generation models. Based on the idea of transfer learning, RKT generates the corresponding reasoning knowledge for each LogicNLG data by learning from out-of-domain datasets. For example, RKT generate a reasoning knowledge for the description in Figure 1 : “select the rows whose location record fuzzily matches to St Pete Times Forum. the number of such rows is 4”.

We have observed that this reasoning knowledge

vehicles & animals			
country	date	label	format
united kingdom	...	parlophone	lp
united kingdom	...	parlophone	cd
united kingdom	...	parlophone	cd digipak
united states	...	astralwerks	cd
australia	...	capitol records	cd

description : Australia and United State made their Label using the same Format.

reasoning knowledge : select the rows whose country record fuzzily matches to Australia. for the format records of these rows, all of them fuzzily match to Cd. there is only one such row in the table. the label record of this unique row is Capitol Records.

Figure 2: The incorrect reasoning knowledge generated by RKT.

is not manually annotated but generated by models, which introduces a significant amount of noisy data. As shown in Figure 2, the reasoning knowledge generated by RKT for description is incorrect, it may have a negative impact on model training process.

We hypothesize that if we can filter out some of the noisy data and train the model to use only the remaining clean data, we can further improve the performance of existing generation models. However, relying on human to label nearly 30,000 such pieces of data could take more time and effort. With powerful reasoning capabilities, Large Language Models (LLMs) has the potential to replace humans in the annotation process(Zhao et al., 2023a). But with large token size of serialized datas, LLMs may have higher economic costs.

To filter reasoning knowledge and minimize costs, we propose a framework *reasoning knowledge filter* (RKF) based on knowledge distillation. RKF first splits the LogicNLG dataset, which contains reasoning knowledge introduced by RKT(Liu et al., 2024), into two parts of different sizes. Then, GPT-4o is tasked with adding correctness labels to the reasoning knowledge in the smaller part. Subsequently, the data with correctness labels is used to train a reasoning knowledge correctness classification model. Finally, this classification model is employed to determine the correctness of the reasoning knowledge in the remaining larger part of the data. The results obtained on LogicNLG dataset demonstrate that our method further improves the performance of existing models, with an increase of 1.4 points in SP-Acc and 0.7 points in NLI-Acc over the current state-of-the-art model.

Overall, our contributions are as follows:

- We proposed a framework *reasoning knowledge filter* to filter out noisy data from the

existing reasoning knowledge in LogicNLG, successfully reducing the training bias in existing generation models.

- We provided a updated version of the LogicNLG dataset, which offers a higher degree of alignment between reasoning knowledge, tables, and descriptions compared to provided in RKT.
- We achieved optimal performance using less data than the original LogicNLG training set.

2 Related Work

With the recognition of pre-trained language models (PLMs) by researchers, most recent works have adopted the approach of end-to-end fine-tuning of PLMs to accomplish table-to-text generation. Kale and Rastogi (2020) obtained notable results across various datasets by solely fine-tuning T5; Wang et al. (2022a) changed the attention and position encoding base on T5; and An et al. (2022) proposed optimizing the loss function during the fine-tuning of PLMs using contrastive learning. For LT2T, R2D2 modified the input and loss function used during the fine-tuning of T5(Nan et al., 2022).

Notably, in the data-to-text generation field, some researchers have been dedicated to integrating PLMs with traditional natural language generation generation method content planning and surface realisation(Holmes-Higgin, 1994). This type of model, which uses PLMs for content planning and surface realization, is referred to as a pipeline model. The generation of descriptions using a pipeline model typically involves two steps: (1) Content Planning: selecting and ordering key information from the input data, and (2) Surface Realization: generating descriptions based on the key information. PlanGen(Su et al., 2021) is a representative of such pipeline models.

The reasoning capabilities of LLMs have garnered significant attention in recent years. Chain-of-Thought (CoT) is a crucial technique for further unlocking the reasoning capabilities of LLMs. Widely recognized research on CoT includes few-shot CoT(Wei et al., 2022), zero-shot CoT(Kojima et al., 2022), auto-CoT(Zhang et al., 2022) and Self-consistency(Wang et al., 2022b). Our work partially draws on few-shot CoT and self-consistency research.

Knowledge Distillation (KD)(Xu et al., 2024) in LLMs is a technique used to transfer knowledge

from a large, complex model to a smaller, simpler model. This process aims to retain the performance of the larger model while reducing computational requirements. KD methods are mainly used for white-box classification models or for training smaller models to replicate the behavior of black-box model APIs, such as ChatGPT. Many KD efforts achieve text classification tasks by imitating the output distribution of the teacher model(Song et al., 2020; Liang et al.; Zhang et al., 2023).

3 Method

This section begins with some preliminary information for our work (3.1), followed by *reasoning knowledge filter* framework (3.2).

3.1 Preliminaries

3.1.1 LogicNLG Datasets

The LogicNLG dataset is divided into three parts: train, dev, and test. We define these parts as follows:

$$DT = \{(T, S)_i\}_{i=1}^{|DT|} \quad (1)$$

$$DV = \{(T, S)_i\}_{i=1}^{|DV|} \quad (2)$$

$$DS = \{(T, S)_i\}_{i=1}^{|DS|} \quad (3)$$

in which T is a structured table, S is a description, and $|DT||DV||DS|$ is the size of dataset.

3.1.2 LogicNLG Datasets With Reasoning Knowledge

RKT introduces explicit reasoning knowledge to the LogicNLG train set, expanding the original (T, S) pairs into triplets. We can redefine DT as

$$DTRK = \{(T, S, RK)_i\}_{i=1}^{|DTRK|} \quad (4)$$

where RK is reasoning knowledge.

Reasoning knowledge is a formal process of reasoning out descriptions from tables(Liu et al., 2024). It based on seven categories of logical operation functions: count, superlative, comparative, aggregation, majority, unique, and ordinal (details of the logical operation functions are provided in the AppendixA). High-quality reasoning knowledge can be seen as a graph structure. As shown in Figure 3, the bottom-up execution order in graph structures provides a more intuitive explanation of the formal process of inferring descriptions from tables.

reasoning knowledge : select the rows whose location record fuzzily matches to St Pete Times Forum. the number of such rows is 4.

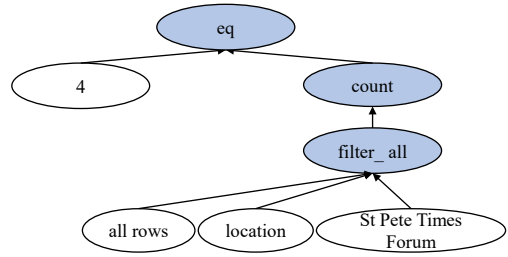


Figure 3: The correspondence between reasoning knowledge and graph structures for example in Figure 1.

3.1.3 Table Serialization

In table-to-text generation, tables are typically transformed into a sequence format before being fed into the model. Current research suggests various methods for table serialization. Early research tended to serialize tables using XML formats(An et al., 2022). With the advent of pre-trained language models, it has become common to use natural language templates for table serialization(Chen et al., 2020a). For LLMs, directly adding special tokens between cells and rows is also an effective serialization choice. Moreover, Chen et al. (2020a) highlighted that serializing the entire table can negatively impact model performance. Consequently, we adhere to Chen et al. (2020a) approach and choose specific sub-columns of the table for serialization.

3.2 Reasoning Knowledge Filter

To illustrate the dataset’s transformation, the process of *reasoning knowledge filter* is depicted in Figure 4. The numbers enclosed in brackets within the figure correspond to the equations provided in this paper. Now, we will separately introduce the implementation details.

3.2.1 Dataset Partition

As mentioned in Section 1, we had planned to use LLMs to replace human annotation because manually annotating the correctness of reasoning knowledge incurs significant time and financial costs. Therefore, we initially planned to use LLMs to completely replace human annotation. However, while LLMs can significantly reduce time costs, they do not lower economic costs. According to our estimates, with even the most token-efficient method to serialize the tables, the entire training set would still need over 30,000,000 prompt tokens and 45,000,000 completion tokens (for each

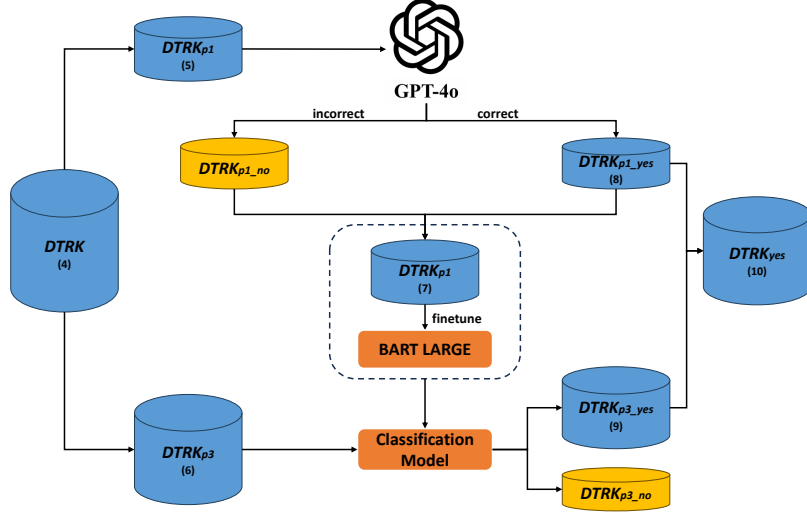


Figure 4: Reasoning knowledge filter process.

data point, we assume that the prompt requires a minimum of 1,024 tokens, and three completions require 1,536 tokens). To further reduce economic costs, we decided to partition $DTRK$ in a 1:3 ratio. The partitioned data is defined as follows:

$$DTRK_{p1} = \{(T, S, RK)_i\}_{i=1}^{|DTRK_{p1}|} \quad (5)$$

$$DTRK_{p3} = \{(T, S, RK)_i\}_{i=1}^{|DTRK_{p3}|} \quad (6)$$

3.2.2 Annotation of Reasoning Knowledge Correctness Based on LLMs

It is widely acknowledged that LLMs possess powerful multi-step reasoning capabilities. In scenarios where data is scarce, numerous studies have attempted to use LLMs to generate training data (Long et al., 2024). We believe that employing LLMs to assess the correctness of reasoning knowledge is feasible.

The recently released GPT-4o has demonstrated outstanding performance in multi-step reasoning capabilities, particularly excelling in handling complex logical reasoning tasks. It has shown significant advantages in accurately generating and verifying intricate reasoning chains. Therefore, we have chosen GPT-4o to assess the correctness of reasoning knowledge in $DTRK_{p1}$.

Figure 5 illustrates the entire process of using GPT-4o to assess the correctness of reasoning knowledge. Each prompt can be divided into two parts: the instruction description and the serialized data. The instruction description remains constant in each prompt. We prompt GPT-4o to perform step-by-step reasoning based on the reasoning knowledge to determine whether the facts

included in the description can be derived. To quickly pinpoint the judgment results of GPT-4o, we also require GPT-4o to return a definitive response, specifically either “*The result is yes*” or “*The result is no*”. In the completions generated by GPT-4o, there remains a small portion of data that does not include the two specified markers. We consider this subset of data as containing incorrect reasoning knowledge. In serialized data, we choose to add special tokens between cells and rows when table serialization. We distinguish cells and rows with “#” and newlines, respectively. Based on our tests, LLMs perform well with each serialization scheme mentioned in Section 3.1.3. However, the scheme involving the addition of special characters is the most token-efficient. Due to the possibility of errors in single-instance inferences generated by LLMs, we require GPT-4o to randomly generate three completions for each data point to maximize the collection of accurate reasoning knowledge. The conclusions derived from these three completions may differ. Inspired by research on the self-consistency of CoT, we select the conclusion that appears most frequently as the final result.

After GPT-4o completes the correctness assessment of reasoning knowledge for all $DTRK_{p1}$ data, the definition of $DTRK_{p1}$ will be updated as follows:

$$DTRK_{p1} = \{(T, S, RK, FLAG)_i\}_{i=1}^{|DTRK_{p1}|} \quad (7)$$

where the $FLAG$ is used to indicate the correctness of the reasoning knowledge. The candidate values for $FLAG$ are “*YES*” or “*NO*”. Further-

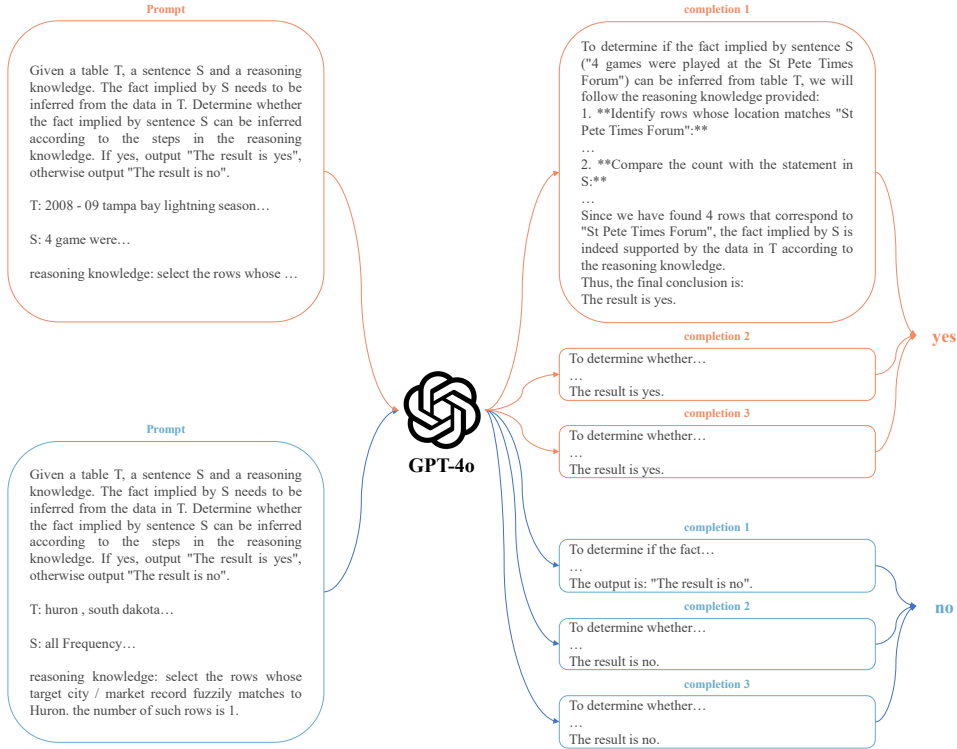


Figure 5: Annotation of reasoning knowledge correctness based on LLMs. Detailed examples will be presented in the Appendix A.

more, we define the subset of $DTRK_{p1}$ data where the $FLAG$ is set to “YES” as follows:

$$DTRK_{p1_yes} = \{(T, S, RK)_i\}_{i=1}^{|DTRK_{p1_yes}|} \quad (8)$$

3.2.3 Reasoning Knowledge Correctness Classification Model

In previous section, we annotated $DTRK_{p1}$ using GPT-4o. Now, we need to complete the annotation for $DTRK_{p3}$. Knowledge distillation (Xu et al., 2024), as a technique where smaller models learn from the capabilities of LLMs, is widely utilized by researchers. Inspired by knowledge distillation, we decided to use a smaller text classification model to learn the prediction distribution of GPT-4o on $DTRK_{p1}$. Subsequently, we will employ this classification model to annotate the reasoning knowledge correctness for dataset $DTRK_{p3}$.

We selected the BART-large (Lewis, 2019) model as the foundation for our classification model. By jointly training the pre-trained weights of BART-large and the randomly initialized weights of the linear classification layer, we can assess the correctness of reasoning knowledge. We use $DTRK_{p1}$ to train the classification model, requiring the classification model to determine the correctness of

reasoning knowledge based on the input table, description, and reasoning knowledge. Figure 8 in appendix illustrates the inputs and outputs during the training of the classification model. Note that, following previous table-to-text research (Chen et al., 2020a, 2021; Nan et al., 2022; Liu et al., 2024), we use natural language templates to achieve table serialization during training.

After completing the training, we use the classification model to assess the correctness of all reasoning knowledge in $DTRK_{p3}$. Using the same input format as during training, we provide the classification model with a table, a description, and a reasoning knowledge, and require it to determine the correctness of the reasoning knowledge. We merge the data points that the classification model deems as correct reasoning knowledge into a set, which can be defined as:

$$DTRK_{p3_yes} = \{(T, S, RK)_i\}_{i=1}^{|DTRK_{p3_yes}|} \quad (9)$$

3.2.4 Merge Data

Through Sections 3.2.2 and 3.2.3, we successfully filtered all data split according to a 1:3 ratio in Section 3.2.1. The data marked as “YES” for the correctness of reasoning knowledge are the ones

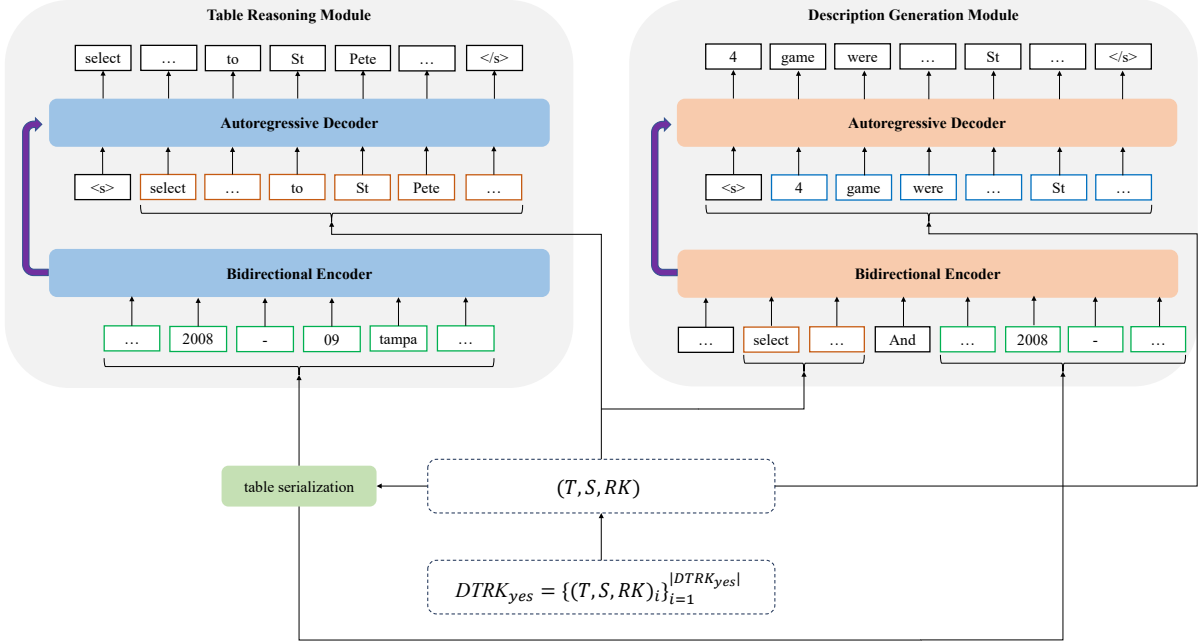


Figure 6: Training details of the generation model.

we need. Now, we need to merge the two parts of the data.

We merge $DTRK_{p1_yes}$ and $DTRK_{p3_yes}$ together and define them as follows:

$$DTRK_{yes} = \{(T, S, RK)_i\}_{i=1}^{|DTRK_{yes}|} \quad (10)$$

where $|DTRK_{yes}|$ equals $|DTRK_{p1_yes}|$ plus $|DTRK_{p3_yes}|$.

3.2.5 Generation Model

$DTRK_{yes}$ and DT differ in two aspects: first, the former has a smaller size; second, the former provides an additional piece of reasoning knowledge for each data point compared to the latter. This reasoning knowledge should be an intermediate process that the model can generate on its own, with the aim of better generating descriptions.

Following the work of Liu et al. (2024), we decompose the LT2T task into two subtasks: table reasoning and description generation. We implemented a pipeline-based generation model using the BART-large model as the foundation to accomplish these two subtasks. Figure 6 illustrates the training details of the generation model using the data from Figure 1 as an example. The generation model is divided into two independent modules: table reasoning module and description generation module.

The training objective of the table reasoning module is to generate a reasoning knowledge based

on table. We expect the table reasoning module to learn the experience knowledge embedded in reasoning knowledge, thereby enabling it to perform logical reasoning based on tabular data.

The training objective of the description generation module is to generate a description based on table and reasoning knowledge. Since the table reasoning module has already completed the reasoning based on the table, the description generation module’s task now shifts to generate a summarizing description based on the provided reasoning knowledge and table.

This pipeline generation model decomposes the originally complex task of table-to-text generation into two relatively simpler subtasks: reasoning and generation. Each module focuses solely on its respective subtask, which facilitates the model’s ability to uncover logical relationships within the table and produce accurate and logically faithful descriptions.

The prediction process of this generation model is as follows: (1) Input a serialized table into the table reasoning module, and the table reasoning module generates a reasoning knowledge based on the table information. (2) The reasoning knowledge generated by the table reasoning module is appended to the serialized table. (3) The concatenated sequence is then input into the description generation module, which combines the table and the reasoning knowledge to generate a description

that is logically faithful to the facts presented in the table.

4 Experiments

In this section, we describe our evaluation methodology, including the evaluation metrics and comparison baselines, evaluation and ablation study to thoroughly assess our model.

4.1 Evaluation Metrics

The evaluation metrics for LT2T are categorized into two types: surface-level matching and logical fidelity.

Similar to LT2T, we employ BLEU-1/2/3 (Papineni et al., 2002) as our evaluation metric, which assesses the consistency between the model’s output and the reference by using n-gram matching.

Following previous work (Chen et al., 2020a; Nan et al., 2022; Zhao et al., 2023b), we assessed our method using two evaluation metrics: SP-Acc, based on semantic parsing, and NLI-Acc, based on entailment scores. During evaluation, SP-Acc converts the predicted description into a logical form (Chen et al., 2020a) and executes it, while NLI-Acc computes the entailment score between the predicted description and the table. Both metrics determine the proportion of descriptions that meet specific criteria, thereby measuring whether a description is logically faithful to the table.

4.2 Baselines

Our method will be compared with the following studies:

GPT-Coarse-to-Fine, a method proposed by Chen et al. (2020a) when introducing LT2T. This method involves copying the reference description and masking key entity information, then appending it to the front of the reference description as training data for the model. This approach partially alleviates the issue of the model generating descriptions that are not logically faithful.

DCVED, an encoder-decoder for the LT2T, which employ the causal intervention method to mitigate spurious correlations (Chen et al., 2021).

R2D2, which addresses the issue of unfaithful data-to-text generation by replacing key information in the description with elements from the same column or by sampling from the model’s prediction distribution. It also introduces a new unlikelihood loss function, training the system to act as both a generator and a faithfulness discriminator (Nan et al., 2022).

LoFT, which employs a model to transform descriptions into logic forms (Chen et al., 2020b) and subsequently trains the model to generate descriptions based on both the tables and the logic forms. By incorporating candidate logic form synthesizers, LoFT improves the fidelity and diversity of the model’s predictions (Zhao et al., 2023b).

HISTALIGN, which enhances the context dependence of language models by introducing a novel Cache-LM training objective to ensure proper cache alignment. This allows the model to effectively utilize historical signals, thereby improving the coherence and faithfulness of the generated descriptions (Wan et al., 2023).

RKT, a framework that introduces explicit reasoning knowledge into LogicNLG, mitigating the issue of erroneous information often generated by end-to-end models that directly infer descriptions from tables (Liu et al., 2024).

4.3 Automatic Evaluation

Table 1 presents the results of our experiment. It can be observed that our method surpasses previous approaches in terms of logical fidelity evaluation metrics. Our method improves SP-Acc by up to 1.4 points and NLI-Acc by 0.7 points. The enhancement in these two metrics demonstrates that our table reasoning module effectively generates reliable reasoning knowledge from tables, and the description generation module uses this reasoning knowledge to guide the generation of more logically faithful descriptions.

Regarding BLEU scores, our method is competitive with previous approaches. However, it appears that further research is necessary. BLEU evaluates the n-gram token matching between reference descriptions and predicted descriptions, without assessing the correctness of the descriptions. However, our method shows significant improvement in logical fidelity metrics, indicating that the descriptions we generated are indeed more accurate. By observing the prediction results, we believe that the low BLEU score was attributed to our generated descriptions being logically faithful but not matching the reference descriptions. This situation may have occurred because the table reasoning module generated reasoning knowledge that did not match the reference descriptions. We provide an example of this situation in Figure 7. The reference description indicates that Pierre Lamine has more points than Shinji Someya, whereas the predicted description states that Mark Cockerell had the highest number

model	bleu1	bleu2	bleu3	SP-Acc	NLI-Acc
GPT-Coarse-to-Fine (sm)(Chen et al., 2020a)	46.6	26.8	13.3	42.7	72.2
GPT-Coarse-to-Fine (med)(Chen et al., 2020a)	49.0	28.3	14.6	45.3	76.4
DCVED(Chen et al., 2021)	49.5	28.6	15.3	43.9	76.9
R2D2(Nan et al., 2022)	51.8	32.4	18.6	50.8	85.6
LOFT(Zhao et al., 2023b)	48.1	27.7	14.9	57.7	86.9
HISTALIGN(Wan et al., 2023)	56.7	37.6	26.3	53.1	85.7
RKT(Liu et al., 2024)	54.8	34.1	19.7	59.6	88.1
Ours	55.2	34.3	20.0	61.0	88.8

Table 1: Performance results on the LogicNLG test set.

1976 world junior figure skating championships

...	name	...	point	...
...	mark cockerel	...	172.42	...
...	takashi mura	...	165.70	...
...	brian pockar	...	166.62	...
...	pierre lamine	...	150.50	...
...	shinji someya	...	150.34	...

Reference description : Pierre Lamine has a mere 0.16 more Point than Shinji Someya.

Prediction reasoning knowledge : select the row whose points record of all rows is maximum. the name record of this row is Mark Cockerell.

Prediction description : Mark Cockerell had the highest number of Point in the 1976 World Junior Figure Skating Championship.

Figure 7: Prediction example in LogicNLG test set.

model	SP-Acc	NLI-Acc
Previous best	59.6	88.1
ours (L+C)	61.0	88.8
ours (C)	60.4	88.5

Table 2: Ablation study result. (L+C) indicates that the filtering process utilized both the GPT-4o model and the classification model. (C) indicates that the filtering process used only the classification model.

of points.

5 Ablation Study

In Section 3.2.2 and 3.2.3, we utilized the GPT-4o model and a reasoning knowledge correctness classification model to filter out data from *DTRK* that contained correct reasoning knowledge. To further validate the effectiveness of the classification model, we applied it alone to filter dataset *DTRK*, resulting in a new dataset defined as follows:

$$DTRK_{cyes} = \{(T, S, RK)_i\}_{i=1}^{|DTRK_{cyes}|} \quad (11)$$

Subsequently, we retrained the generation model using *DTRK*_{cyes} according to the methodology

outlined in Section 3.2.5.

As shown in Table 2, significant performance improvements were observed even when the classification model was used to filter *DTRK*. This demonstrates the utility of our classification model base on knowledge distillation.

6 Conclusion

In this paper, we propose a framework *reasoning knowledge filter* based on large language models and knowledge distillation. This framework successfully filters out a dataset that is smaller in size compared to the original LogicNLG dataset but has a higher match quality among table, description, and reasoning knowledge triplets. Through this method, we are able to significantly enhance the performance and logical faithfulness of the generation model.

7 Limitations

Although our method improved logical faithfulness, it still falls short of human performance, which indirectly highlights the difficulty of the task. Meanwhile, three-quarters of *DTRK* was filtered by a smaller classification model, which learned the output distribution of GPT-4o through knowledge distillation. However, it is foreseeable that its filtering performance is not as effective as that of GPT-4o.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (U1908216).

References

Chenxin An, Jiangtao Feng, Kai Lv, Lingpeng Kong, Xipeng Qiu, and Xuanjing Huang. 2022. Cont: Contrastive neural text generation. *Advances in Neural Information Processing Systems*, 35:2197–2210.

- Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020a. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7929–7942.
- Wenqing Chen, Jidong Tian, Yitian Li, Hao He, and Yao-hui Jin. 2021. De-confounded variational encoder-decoder for logical table-to-text generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5532–5542.
- Zhiyu Chen, Wenhu Chen, Hanwen Zha, Xiyu Zhou, Yunkai Zhang, Sairam Sundaresan, and William Yang Wang. 2020b. Logic2text: High-fidelity natural language generation from logical forms. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2096–2111.
- Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. 2021. Hitab: A hierarchical table dataset for question answering and natural language generation. *arXiv preprint arXiv:2108.06712*.
- Paul Holmes-Higgin. 1994. Text generation—using discourse strategies and focus constraints to generate natural language text by kathleen r. mckeown, cambridge university press, 1992, pp 246, £ 13.95, isbn 0-521-43802-0. *The Knowledge Engineering Review*, 9(4):421–422.
- Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 97–102.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.
- Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- M Lewis. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- Kevin J Liang, Weituo Hao, Dinghan Shen, Yufan Zhou, Weizhu Chen, Changyou Chen, and Lawrence Carin. Mixkd: Towards efficient distillation of large-scale language models. In *International Conference on Learning Representations*.
- Baoqiang Liu, Yu Bai, Fang Cai, Shuang Xue, Na Ye, and XinYuan Ye. 2024. Reasoning knowledge transfer for logical table-to-text generation. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.
- Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On llms-driven synthetic data generation, curation, and evaluation: A survey. *arXiv preprint arXiv:2406.15126*.
- Linyong Nan, Lorenzo Jaime Flores, Yilun Zhao, Yixin Liu, Luke Benson, Weijin Zou, and Dragomir Radev. 2022. R2d2: Robust data-to-text with replacement detection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6903–6917.
- Jekaterina Novikova, Ondrej Dusek, and Verena Rieser. 2017. The e2e dataset: New challenges for end-to-end generation. In *18th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 201–206. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. Totto: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186.
- Kaitao Song, Hao Sun, Xu Tan, Tao Qin, Jianfeng Lu, Hongzhi Liu, and Tie-Yan Liu. 2020. Lightpaff: A two-stage distillation framework for pre-training and fine-tuning. *arXiv preprint arXiv:2004.12817*.
- Yixuan Su, David Vandyke, Sihui Wang, Yimai Fang, and Nigel Collier. 2021. Plan-then-generate: Controlled data-to-text generation via planning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 895–909.
- David Wan, Shiyue Zhang, and Mohit Bansal. 2023. Hystalign: Improving context dependency in language generation by aligning with history. *arXiv preprint arXiv:2305.04782*.
- Fei Wang, Zhewei Xu, Pedro Szekely, and Muhao Chen. 2022a. Robust (controlled) table-to-text generation with structure-aware equivariance learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5037–5048.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022b. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.
- Sam Wiseman, Stuart M Shieber, and Alexander M Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263.
- Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*.
- Rongzhi Zhang, Jiaming Shen, Tianqi Liu, Jialu Liu, Michael Bendersky, Marc Najork, and Chao Zhang. 2023. Do not blindly imitate the teacher: Using perturbed loss for knowledge distillation. *arXiv e-prints*, pages arXiv–2305.
- Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023a. A survey of large language models. *arXiv preprint arXiv:2303.18223*.
- Yilun Zhao, Zhenting Qi, Linyong Nan, Lorenzo Jaime Flores, and Dragomir Radev. 2023b. Loft: Enhancing faithfulness and diversity for table-to-text generation via logic form control. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 554–561.

A Appendix

Table 3 presents all logical operation functions related to reasoning knowledge. Figure 9 and 10 illustrate how data is annotated using GPT-4o.

Name	Arguments	Output	Description
count	view	number	returns the number of rows in the view
only	view	bool	returns whether there is exactly one row in the view
hop	row, header string	object	returns the value under the header column of the row
and	bool, bool	bool	returns the boolean operation result of two arguments
max/min/avg/sum	view, header string	number	returns the max/min/average/sum of the values under the header column
nth_max/nth_min	view, header string	number	returns the n-th max/n-th min of the values under the header column
argmax/argmin	view, header string	row	returns the row with the max/min value in header column
nth_argmax/nth_argmin	view, header string	row	returns the row with the n-th max/min value in header column
eq/not_eq	object, object	bool	returns if the two arguments are equal
round_eq	object, object	bool	returns if the two arguments are roughly equal under certain tolerance
greater/less	object, object	bool	returns if argument 1 is greater/less than argument 2
diff	object, object	object	returns the difference between two arguments
filter_eq/not_eq	view, header string, object	view	returns the subview whose values under the header column is equal/not equal to argument 3
filter_greater/less	view, header string, object	view	returns the subview whose values under the header column is greater/less than argument 3
filter_greater_eq/less_eq	view, header string, object	view	returns the subview whose values under the header column is greater/less or equal than argument 3
filter_all	view, header string	view	returns the view itself for the case of describing the whole table
all_eq/not_eq	view, header string, object	bool	returns whether all the values under the header column are equal/not equal to argument 3
all_greater/less	view, header string, object	bool	returns whether all the values under the header column are greater/less than argument 3
all_greater_eq/less_eq	view, header string, object	bool	returns whether all the values under the header column are greater/less or equal to argument 3
most_eq/not_eq	view, header string, object	bool	returns whether most of the values under the header column are equal/not equal to argument 3
most_greater/less	view, header string, object	bool	returns whether most of the values under the header column are greater/less than argument 3
most_greater_eq/less_eq	view, header string, object	bool	returns whether most of the values under the header column are greater/less or equal to argument 3

Table 3: Logical Operation Functions, same as in logic2text(Chen et al., 2020b).

input	<p>Given a table T, a sentence S and a reasoning knowledge. The fact implied by S needs to be inferred from the data in T. Determine whether the fact implied by sentence S can be inferred according to the steps in the reasoning knowledge.</p> <p>S:only Gary Player represented South Africa at the 2001 Open Championship.</p> <p>reasoning knowledge: select the rows whose country record fuzzily matches to South Africa. there is only one such row in the table. the player record of this unique row is Gary Player.</p> <p>T: The caption is "2001 open championship". In row 1, the player is Justin Leonard, the country is United States. In row 2, the player is Nick Faldo, the country is England. In row 3, the player is Tom Lehman, the country is United States. In row 4, the player is John Daly, the country is United States. In row 5, the player is Seve Ballesteros, the country is Spain. In row 6, the player is Bob Charles, the country is New Zealand. In row 7, the player is Tony Jacklin, the country is England. In row 8, the player is Tom Watson, the country is United States. In row 9, the player is Gary Player, the country is South Africa.</p>
output	1(YES)
input	<p>Given a table T, a sentence S and a reasoning knowledge. The fact implied by S needs to be inferred from the data in T. Determine whether the fact implied by sentence S can be inferred according to the steps in the reasoning knowledge.</p> <p>S:only 1 Mass Suicide occured in the united state, in California in 1997.</p> <p>reasoning knowledge:select the rows whose type record fuzzily matches to Mass Suicide. among these rows, select the row whose date record of all rows is 1st minimum. the location record of this row is California.</p> <p>T: The caption is "list of battles and other violent events by death toll". In row 1, the type is Mass Suicide, Murder, the location is Japan, the date is 1945. In row 2, the type is Mass Suicide, the location is Mariana Islands, the date is 1944. In row 3, the type is Mass Human Sacrifice, the location is Templo Mayor, Tenochtitlan, the date is 1487. In row 4, the type is Mass Suicide, the location is Judea, the date is 73. In row 5, the type is Mass Suicide, Murder, the location is Guyana, the date is 1978. In row 6, the type is Mass Suicide, the location is Uganda, the date is 2000, March. In row 7, the type is Mass Suicide, the location is Switzerland, Canada, the date is 1994. In row 8, the type is Mass Suicide, the location is California, the date is 1997. In row 9, the type is Mass Suicide, the location is France, the date is 1995, December 23.</p>
output	0(NO)

Figure 8: Train data for reasoning knowledge correctness classification model.

Prompt
<p>Given a table T, a sentence S and a reasoning knowledge. The fact implied by S needs to be inferred from the data in T. Determine whether the fact implied by sentence S can be inferred according to the steps in the reasoning knowledge. If yes, output "The result is yes", otherwise output "The result is no".</p> <p>T: 2008 - 09 tampa bay lightning season ...</p> <p>row number#location 1#wachovia center 2#st pete times forum 3#st pete times forum 4#td banknorth garden 5#hsbc arena 6#bell centre 7#scotiabank place 8#st pete times forum 9#philips arena 10#mellon arena 11#bankatlantic center 12#st pete times forum ...</p> <p>S: 4 game were played at the St Pete Time Forum. reasoning knowledge: select the rows whose location record fuzzily matches to St Pete Times Forum. the number of such rows is 4.</p>
Completion 1
<p>To determine if the fact implied by sentence S ("4 games were played at the St Pete Times Forum") can be inferred from table T, we will follow the reasoning knowledge provided:</p> <ol style="list-style-type: none"> Identify rows whose location matches "St Pete Times Forum": <ul style="list-style-type: none"> Looking at the table T, we can see the following rows with the location "st pete times forum": <ul style="list-style-type: none"> Row 2 Row 3 Row 8 Row 12 We count the matches: <ul style="list-style-type: none"> There are 4 rows that match the location "st pete times forum". Compare the count with the statement in S: <ul style="list-style-type: none"> Sentence S states that "4 games were played at the St Pete Times Forum." The reasoning knowledge states that we should find 4 rows that match. <p>Since we have found 4 rows that correspond to "St Pete Times Forum", the fact implied by S is indeed supported by the data in T according to the reasoning knowledge. Thus, the final conclusion is: The result is yes.</p>
Completion 2
<p>To determine whether the fact implied by sentence S can be inferred from the data in table T, we will follow the reasoning knowledge provided.</p> <ol style="list-style-type: none"> The reasoning knowledge states to "select the rows whose location record fuzzily matches to St Pete Times Forum." We need to count the number of rows in table T that match "St Pete Times Forum." <p>Looking at the data in T:</p> <ul style="list-style-type: none"> Row 2: st pete times forum Row 3: st pete times forum Row 8: st pete times forum Row 12: st pete times forum <p>From the analysis, we see that there are 4 rows that match "St Pete Times Forum".</p> <p>Now, let's evaluate the fact implied by S: "4 games were played at the St Pete Time Forum." Since we have identified 4 rows that match the location "St Pete Times Forum," this supports the implication stated in S.</p> <p>According to the reasoning knowledge, since the number of rows that match is indeed 4, we can conclude that the implication is valid.</p> <p>Thus, the output is: The result is yes.</p>
Completion 3
<p>To determine whether the fact implied by sentence S can be inferred from the data in table T using the reasoning knowledge provided, we can follow these steps:</p> <ol style="list-style-type: none"> Identify the target location from S: The sentence states that 4 games were played at the St Pete Times Forum. Apply the reasoning knowledge: According to the reasoning knowledge, we need to select the rows from T whose location records fuzzily match "St Pete Times Forum". Examine the data in table T: Looking at the table, we can see the locations listed: <ul style="list-style-type: none"> Row 2: st pete times forum Row 3: st pete times forum Row 8: st pete times forum Row 12: st pete times forum Count the matches: There are 4 rows where the location matches "St Pete Times Forum" (rows 2, 3, 8, and 12). <p>Since the reasoning knowledge states that we need to find 4 rows that fuzzily match "St Pete Times Forum", and we have found exactly 4 such rows, we can conclude that the fact implied by sentence S can indeed be inferred from the data in T.</p> <p>Therefore, the final output is: The result is yes.</p>

Figure 9: GPT-4o Annotation Detail.

<p>Prompt</p> <p>Given a table T, a sentence S and a reasoning knowledge. The fact implied by S needs to be inferred from the data in T. Determine whether the fact implied by sentence S can be inferred according to the steps in the reasoning knowledge. If yes, output "The result is yes", otherwise output "The result is no".</p> <p>T: huron , south dakota ...</p> <p>row number#frequency#target city / market#city of license 1#88.7 fm#huron#huron 2#90.5 fm#huron#huron 3#93.3 fm#huron / mitchell#wessington springs 4#105.1 fm#huron#huron 5#89.7 fm#huron#rocklin , california ...</p> <p>S: all Frequency Target the City Of Huron. reasoning knowledge: select the rows whose target city / market record fuzzily matches to Huron. the number of such rows is 1.</p>
<p>Completion 1</p> <p>To determine if the fact implied by sentence S ("all Frequency Target the City Of Huron") can be inferred from the table T according to the provided reasoning knowledge, let's break down what needs to happen.</p> <p>1. **reasoning knowledge Analysis**: The reasoning knowledge states to select rows whose "target city / market" record fuzzily matches "Huron" and the number of such rows must be 1.</p> <p>2. **Table Analysis**: Let's examine the rows in table T: - Row 1: target city / market is "huron" (matches) - Row 2: target city / market is "huron" (matches) - Row 3: target city / market is "huron / mitchell" (fuzzily could match "huron") - Row 4: target city / market is "huron" (matches) - Row 5: target city / market is "huron" (matches) Thus, we have the following evaluation: - Row 1: matches - Row 2: matches - Row 3: fuzzily matches (due to the presence of "huron") - Row 4: matches - Row 5: matches</p> <p>3. **Counting Matches**: The total number of rows that match or fuzzily match "Huron" is 5 (Rows 1, 2, 3, 4, and 5).</p> <p>4. **Conclusion Based on reasoning knowledge**: The reasoning knowledge requires that the number of such rows must be exactly 1. Since we found 5 rows that match or fuzzily match "Huron", we do not satisfy the condition of having exactly 1 row.</p> <p>Therefore, following the steps outlined in the reasoning knowledge, we find that the fact implied by sentence S cannot be inferred from the data in table T. The output is: "The result is no".</p>
<p>Completion 2</p> <p>To determine whether the fact implied by sentence S ("all Frequency Target the City Of Huron") can be inferred from the data in table T according to the specified reasoning knowledge, we can follow the steps outlined in the rule.</p> <p>1. The rule states to select the rows whose "target city / market" record fuzzily matches to "Huron".</p> <p>2. In table T, we will look for rows where the "target city / market" includes "Huron": - Row 1: target city / market = "huron" (matches) - Row 2: target city / market = "huron" (matches) - Row 3: target city / market = "huron / mitchell" (matches fuzzily, as it includes "Huron") - Row 4: target city / market = "huron" (matches) - Row 5: target city / market = "huron" (matches fuzzily, as it is part of the broader match)</p> <p>3. Counting the matched rows, we have: - Row 1: matches - Row 2: matches - Row 3: matches (fuzzy) - Row 4: matches - Row 5: matches (fuzzy) - Total matched rows = 5</p> <p>4. The reasoning knowledge also states that the expected number of such rows is 1. Since we found 5 rows that match, this contradicts the rule.</p> <p>Based on this reasoning, the fact implied by sentence S cannot be inferred because the number of matching rows does not meet the condition specified in the reasoning knowledge. Thus, the output is: The result is no.</p>
<p>Completion 3</p> <p>To determine whether the fact implied by sentence S ("all Frequency Target the City Of Huron") can be inferred from the data in table T according to the reasoning knowledge provided, we follow these steps:</p> <p>1. **Examine the reasoning knowledge**: The reasoning knowledge states that we need to select the rows whose target city / market record fuzzily matches "Huron" and that the number of such rows should be 1.</p> <p>2. **Check the Table**: Let's review the rows in table T: - Row 1: Target city / market is "huron" (matches) - Row 2: Target city / market is "huron" (matches) - Row 3: Target city / market is "huron / mitchell" (fuzzily matches) - Row 4: Target city / market is "huron" (matches) - Row 5: Target city / market is "huron" (fuzzily matches, but does not exactly match Huron)</p> <p>3. **Count Matching Rows**: - Rows that match "Huron" (fuzzily or exactly): Row 1, Row 2, Row 3, Row 4, Row 5. - Total matches: 5 rows.</p> <p>4. **Compare with the reasoning knowledge**: The reasoning knowledge indicates that there should be exactly 1 row that matches. Since we found 5 rows that match "Huron", this does not meet the criteria of the reasoning knowledge.</p> <p>Given that the condition of having exactly 1 matching row is not fulfilled, we conclude the following: The result is no.</p>

Figure 10: GPT-4o Annotation Detail.

From Chain to Tree: Refining Chain-like Rules into Tree-like Rules on Knowledge Graphs

Wangtao Sun^{1,2}, Shizhu He^{1,2*}, Jun Zhao^{1,2}, Kang Liu^{1,2,3*}

¹Institute of Automation, Chinese Academy of Sciences, Beijing, China

²School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China

³Shanghai Artificial Intelligence Laboratory

Abstract

With good explainability and controllability, rule-based methods play an important role in the task of Knowledge Graph Completion (KGC). However, existing studies primarily focused on learning chain-like rules, whose chain-like structure limits their expressive power. Consequently, chain-like rules often exhibit lower Standard Confidence, and are prone to the incorrect grounding values during reasoning, thus producing erroneous reasoning results. In this paper, we propose the concept of tree-like rules on knowledge graphs to expand the scope of the application and improve the reasoning ability of rule-based methods. To achieve this, we formalize the problem of tree-like rule refinement and propose an effective framework for refining chain-like rules into tree-like rules. Experimental evaluations on four public datasets demonstrate that the proposed framework can seamlessly adapt to various chain-like rule induction methods and the refined tree-like rules consistently exhibit higher Standard Confidence and achieve better performances than the original chain-like rules on link prediction tasks. Furthermore, we illustrate that the improvements brought by tree-like rules are positively correlated with the density of the knowledge graphs. The data and code of this paper can be available at <https://github.com/forangel2014/tree-rule>.

1 Introduction

Knowledge Graph Completion (KGC) (Taskar et al., 2003; Chen et al., 2022; Wang et al., 2023) is a fundamental and important task in Natural Language Processing. For KGC, rule-based methods play a pivotal role, which focus on first learning symbolic and interpretable rules, and then leverage them for effective reasoning within Knowledge Graphs (KGs). In these rule-based methods, the development of comprehensive and high-quality

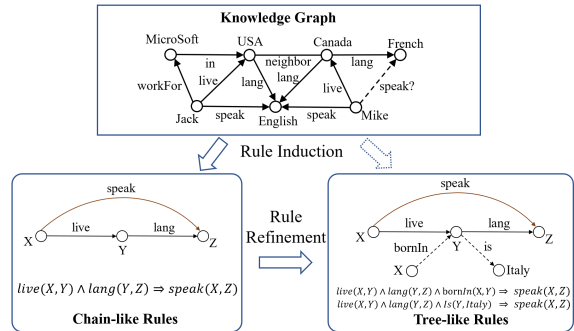


Figure 1: An example of an inaccurate chain-like rule and the refined tree-like rule. Although the chain-like rule (bottom-left) can predict most cases correctly, due to its chain structure, it has limited expressive power. The refined tree-like rule (bottom-right) leverages the information in the KG that originally ignored to improve the chain-like rule.

rule sets is essential for the success of KGC (Nandi et al., 2023; Meilicke et al., 2024).

To this end, previous works have proposed various types of methods to induce rules from the KGs, like symbol-based (Galárraga et al., 2013), embedding-based (Omran et al., 2018; Qu et al., 2021; Cheng et al., 2023), Differentiable-ILP-based (Yang et al., 2017; Yang and Song, 2020), and RL-based (Meilicke et al., 2024). However, these existing rule induction methods only consider the chain-like rules in KGs (Galárraga et al., 2013; Yang et al., 2017; Omran et al., 2018; Meilicke et al., 2024; Qu et al., 2021; Cheng et al., 2023). chain-like rules are a special case of the Horn Clauses (Russell and Norvig, 2016), which is equivalent to a multi-hop reasoning process (Yang and Song, 2020). For example, in the left-bottom subfigure in Figure 1, $live(X, Y) \wedge lang(Y, Z) \Rightarrow speak(X, Z)$ is a typical chain-like rule. Since there is only one path leading the query variable X to the target variable Z , there are limited constraints for the semantics in the reasoning path of

*corresponding author

the chain-like rules. Consequently, chain-like rules are often prone to the incorrect grounding values during reasoning, thus producing erroneous reasoning results. For example, in Figure 1, this rule may reason erroneous facts when X is just a *traveler*, or Y is a *country* with more than one official language and X just *speaks* one of them.

Therefore, this paper introduces the concept of *tree-like rules*, which is a more general form of rules. Apart from a direct path from the query variable X to the target variable Z , the tree-like rule body also contains some branch atoms to constrain the grounding values of the rule. These branch atoms can be seen as “hanging” triplets on the path, which further shape and narrow down the possible grounding values of the rule. The path and the branch atoms stretched together into a tree structure in the KG. For example, in the right-bottom subfigure in Figure 1, the tree-like rule may look like $\text{live}(X, Y) \wedge \text{lang}(Y, Z) \wedge \text{bornIn}(X, Y) \Rightarrow \text{speak}(X, Z)$ or $\text{live}(X, Y) \wedge \text{lang}(Y, Z) \wedge \text{is}(Y, \text{Italy}) \Rightarrow \text{speak}(X, Z)$. The added atoms $\text{bornIn}(X, Y)$ and $\text{is}(Y, \text{Italy})$ yield new constraints for the grounding values of variable Y . With these added constraints, tree-like rules are expected to possess higher quality (Standard Confidence) and avoid the wrong predictions that X *speaks* Z in the cases mentioned before. As a result, tree-like rules possess higher Standard Confidence than chain-like rules and will exhibit better reasoning performance on the task of KGC.

To refine chain-like rules into tree-like rules, the key challenge is to tackle the large combinatorial space of the rule body, i.e. searching and selecting the probable branch atoms. The branch atoms are supposed to exclude the incorrect groundings while still including the correct groundings of the rule. To refine chain-like rules into tree-like rules by adding branch atoms, our proposed framework first transforms the optimization problem of the Standard Confidence of the entire rule to that of the **best branch atom selection** in a specific variable in the rule body. For the best branch atom selection problem, our framework proposes a three-step pipeline: Forward Reasoning, Backward Reasoning, and Candidate Atom Selections. Through this pipeline, our framework effectively refines the original chain-like rules into high-quality tree-like rules.

To verify the effectiveness of tree-like rules refined by our framework, we conduct experiments on four widely used benchmark KGs with three dif-

ferent sources of chain-like rules. The experimental results show that tree-like rules continuously exhibit higher Standard Confidence, and outperform chain-like rules on the link prediction tasks for different sources of chain-like rules on different KGs. With further analysis, we also find that the improvements brought by tree-like rules are positively correlated with the density of the knowledge graphs, showing that tree-like rules have greater advantages in KGs with complex topological structures.

In summary, the contributions of this paper are as follows:

- This paper proposes the concept of tree-like rules for the task of rule induction on KGs. An effective framework is proposed for refining chain-like rules induced from any existing method into tree-like rules.
- The paper conducts experiments on four open-accessed datasets and the results show our refined tree-like rules from different chain-like rules consistently have higher Standard Confidence, and outperform on KG reasoning task than original chain-like rules. Further analysis finds that the improvements brought by tree-like rules are positively correlated with the density of the knowledge graphs.

2 Problem Formulation

In the scope of First-Order Logic (FOL), the rule (or Horn Clause) is formalized as $\sigma \Rightarrow \varphi$. Here, the left part σ is called “rule body”, which serves as the premise, when it is satisfied by some groundings, then the right part, “rule head” φ , will be grounded as the conclusion.

For the evaluation of the quality of such a rule, as we are adding constraints to refine it to be more precise, we adopt the widely-used metric, Standard Confidence (sc). It can be defined as:

$$sc = \frac{\#(S_\sigma \cap S_\varphi)}{\#S_\sigma} \quad (1)$$

where $\#(\cdot)$ stands for the count of possible groundings. This metric can also be easily understood if we take the *rule* as a binary classifier in machine learning: S_σ stands for the situation that the classifier output “positive”, and S_φ stands for the situations that are “true”. Therefore, sc corresponds to the *precision*, being a key metric to describe how much we can trust the rule.

Based on the definition and objective above, our task can be stated as follows. A given chain-like

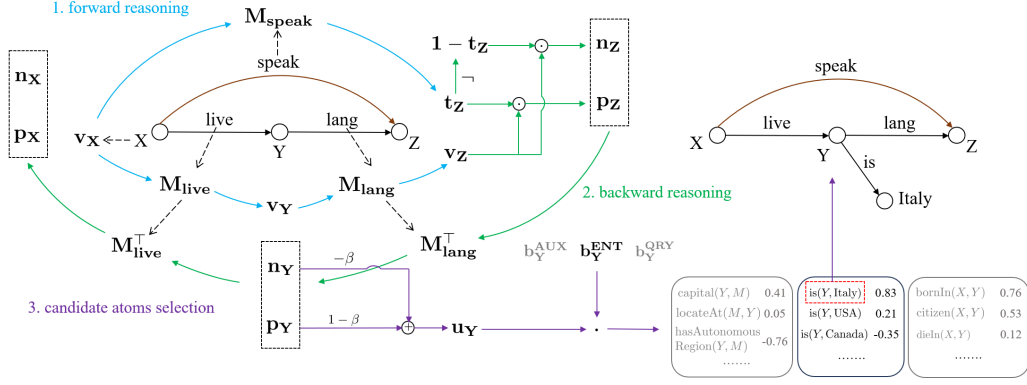


Figure 2: The framework of our proposed method. In the **Forward Reasoning** stage, the Query variable X is first grounded with b randomly sampled entities and by forward reasoning, we obtain the grounding values of Y and Z . In the **Backward Reasoning** stage, we then abductively obtain the positive groundings and negative groundings of each variable in the rule body. Finally in the **Candidate Atoms Selection** stage, three types of candidate branch atoms are then selected according to their inner product scores with the variable representation.

rule \mathcal{R} of length n can be represented as:

$$\mathcal{R} : r_0(x_0, x_1) \wedge \dots \wedge r_{n-1}(x_{n-1}, x_n) \Rightarrow r(x_0, x_n)$$

To obtain tree-like rules from it, we aim to find branch atoms $b(x_i)$ for the variable x_i while being aware of the objective in Eq 1. Finally, a refined tree-like rule has the following format:

$$\begin{aligned} \mathcal{R}^* : & r_0(x_0, x_1) \wedge \dots \wedge r_{i-1}(x_{i-1}, x_i) \wedge b(x_i) \\ & \wedge \dots \wedge r_{n-1}(x_{n-1}, x_n) \Rightarrow r(x_0, x_n) \end{aligned}$$

Please note that when we represent branch atoms as $b(x_i)$, we omit other variables and constants that may appear within the predicate, to emphasize that this branch is a constraint on the variable x_i .

3 Method

Given the chain-like rule, this paper propose to first grounded the rules with entities in the KG, and then find the top branch atoms that could eliminate the undesirable groundings, to make the groundings satisfying rule body be as close as to those satisfying rule head.

Given a KG $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}, \mathcal{R}, \mathcal{T}$ stands for the entity set, relation set, and triplet set, respectively, to reason with the rule on this KG, this paper adopts the matrix representation of entities and relations for reasoning rules.

For an entity $e \in \mathcal{E}$, let $\mathbf{v}_e \in \{0, 1\}^{1 \times |\mathcal{E}|}$ be the one-hot encoding of entity e , i.e. only the i th element is 1 if e is the i th entity in \mathcal{E} . Based on the encoding of single entity, for a variable x in the rule, if the set C_x denote the entities that can

be ground to x , then the variable grounding of x , $\mathbf{v}_x \in \{0, 1\}^{1 \times |\mathcal{E}|}$, is defined as $\mathbf{v}_x = \sum_{e \in C_x} \mathbf{v}_e$.

For an relation $r \in \mathcal{R}$, let $\mathbf{M}_r \in \{0, 1\}^{|\mathcal{E}| \times |\mathcal{E}|}$ be the binary adjacency matrix of relation r , i.e. $\mathbf{M}_r[i, j] = 1$ iff $(e_i, r, e_j) \in \mathcal{T}$. With the definition above, a reasoning hop $(h, r, t?)$ can be modeled by the following matrix multiplication:

$$\mathbf{v}_t = \mathbf{v}_h \mathbf{M}_r \quad (2)$$

To obtain \mathcal{R}^* from \mathcal{R} , we need to find a branch atom $b(x_i)$ (correspond to a constraint vector \mathbf{b}_{x_i}) for the ground value of variable x_i , to best match the reasoning results between rule body and rule head, based on Eq 1, we have:

$$\begin{aligned} J_{\mathcal{R}} = & \left\| \mathbf{v}_{x_0} \mathbf{M}_r \odot \left[\left(\mathbf{v}_{x_0} \prod_{j=0}^{i-1} \mathbf{M}_{r_j} \odot \mathbf{b}_{x_i} \right) \prod_{j=i}^{n-1} \mathbf{M}_{r_j} \right] \right\| \\ & / \left\| \left(\mathbf{v}_{x_0} \prod_{j=0}^{i-1} \mathbf{M}_{r_j} \odot \mathbf{b}_{x_i} \right) \prod_{j=i}^{n-1} \mathbf{M}_{r_j} \right\| \end{aligned} \quad (3)$$

where $\|\mathbf{v}\|$ stands for the 1-norm of the vector \mathbf{v} .

However, it is hard to directly find branch atoms by maximizing Eq 3. Thus we approximately transform it into the best branch atom selection problem:

$$J'_{\mathcal{R}} = S \left(\mathbf{v}_{x_0} \mathbf{M}_r \prod_{j=n-1}^i \mathbf{M}_{r_j}^\top, \mathbf{v}_{x_0} \prod_{j=0}^{i-1} \mathbf{M}_{r_j} \odot \mathbf{b}_{x_i} \right) \quad (4)$$

where $S(\mathbf{a}, \mathbf{b}) = (1 - \beta) \|\mathbf{a} \odot \mathbf{b}\| - \beta \|(1 - \mathbf{a}) \odot \mathbf{b}\|$ is a similarity metric balancing the excluding of the incorrect groundings and the including of the correct groundings. Here, the \odot denotes the element-wise product, and $\beta \in (0, 1)$ balances the degree of including

positive groundings and excluding negative groundings. The colors in this equation correspond to the colors in Figure 2.

To find the best branch atoms to maximize Eq 4, as shown in Figure 2, we propose a three-step framework:

- **§3.1 Forward Reasoning:** we first sample a batch of b entities to ground the query variable x_0 (i.e. X in Figure 2). Then a forward reasoning process transfers the groundings of x_0 to the target variable x_n (i.e. Z in Figure 2) through both the rule body and rule head. At target variable x_n , the positive groundings (\mathbf{p}_{x_n} , the entities correctly predicted by rule body) and negative groundings (\mathbf{n}_{x_n} , the entities incorrectly predicted by the rule body) are obtained.
- **§3.2 Backward Reasoning:** we then abductively obtain the positive groundings and negative groundings of each variable in the rule body, by sequentially multiplying the transpose of the relation matrix with the current grounding vectors.
- **§3.3 Candidate Atoms Selection:** the variable representation is a weighted sum of its positive groundings and negative groundings, which is a trade-off of including positive groundings and excluding negative groundings when adding branch atoms. Finally, we consider three types of branch atoms that are to be added to the rule bodies, the candidate branch atoms are then selected according to their inner product scores with the variable representation.

3.1 Forward Reasoning

Let us first sample a batch of b entities to ground x_0 , and we concatenate their encoding to get the initial variable grounding $\mathbf{v}_{x_0} \in \{0, 1\}^{b \times |\mathcal{E}|}$. As most rules only cover a small part of entities on the entire KG, directly sample b entities from the entity set \mathcal{E} may involve many “inactive tracks” (line of all 0s in matrix) to the reasoning process, we choose to sample from the entities that at least satisfy the first relation r_0 . This can be done by sum up the columns of \mathbf{M}_{r_0} to find the candidate entities:

$$\mathbf{v}_{\text{cand}} = \sum_j \mathbf{M}_{r_0}^{i,j} \quad (5)$$

Then, by randomly keeping b entities from \mathbf{v}_{cand} and concatenating their one-hot encoding, we can obtain the initial variable grounding \mathbf{v}_{x_0} . Based on the initial variable grounding \mathbf{v}_{x_0} , the forward process of rule body can be modeled as a

series of matrix multiplication:

$$\mathbf{v}_{x_{i+1}} = \mathbf{v}_{x_i} \mathbf{M}_{r_i}, i = 0, 1, \dots, n-1 \quad (6)$$

where \mathbf{v}_{x_i} is the variable grounding of x_i , and finally the reasoning result of this rule is \mathbf{v}_{x_n} . Similarly, we can obtain the true grounding of x_n by applying rule head: $\mathbf{t}_{x_n} = \mathbf{v}_{x_0} \mathbf{M}_r$. The \mathbf{v}_{x_n} and \mathbf{t}_{x_n} are vectorized groundings of S_σ and S_φ introduced in Section 2.

3.2 Backward Reasoning

After obtaining the reasoning results from both the rule body (\mathbf{v}_{x_n}) and rule head (\mathbf{t}_{x_n}) of x_n , we then discriminate the positive (the groundings that thought to ground the x_n by rule body, and proved to satisfy the rule head as well) and negative (the groundings that thought to ground the x_n by rule body, but proved not to satisfy the rule head) groundings by performing element-wise production:

$$\begin{aligned} \mathbf{p}_{x_n} &= \mathbf{v}_{x_n} \odot \mathbf{t}_{x_n} \\ \mathbf{n}_{x_n} &= \mathbf{v}_{x_n} \odot (1 - \mathbf{t}_{x_n}) \end{aligned} \quad (7)$$

where the i -th element in $\mathbf{p}_{x_n}/\mathbf{n}_{x_n}$ represents how many entity e_i appear as positive/negative result in rule \mathcal{R} .

Notice that the encoding of r^{-1} (inverse relation of r) is \mathbf{M}_r^\top . Then we can abductively obtain the positive and negative groundings at all variables in rule \mathcal{R} by backward reasoning:

$$\begin{aligned} \mathbf{p}_{x_i} &= \left(\mathbf{p}_{x_{i+1}} \mathbf{M}_{r_{i+1}}^\top \right) \odot \mathbf{v}_{x_i} \\ \mathbf{n}_{x_i} &= \left(\mathbf{n}_{x_{i+1}} \mathbf{M}_{r_{i+1}}^\top \right) \odot \mathbf{v}_{x_i} \end{aligned} \quad (8)$$

By performing such a backward reasoning process, we can have the knowledge that at each variable of the rule, which entities are desired (positive) and which are undesired (negative).

3.3 Candidate Atoms Selection

After obtaining the positive and negative groundings of each variable of the rule, we then propose to refine the chain-like rules by evaluating the candidate branches on each variable. The branch atoms we add at a specific variable of rule \mathcal{R} aims to include positive groundings while excluding the negative ones as much as possible. So we define the representation of each variable by linearly combining \mathbf{p}_{x_i} and \mathbf{n}_{x_i} :

$$\mathbf{u}_{x_i} = (1 - \beta) \cdot \mathbf{p}_{x_i} - \beta \cdot \mathbf{n}_{x_i} \quad (9)$$

where $\beta \in (0, 1)$ is a weight balance the degree of including positive groundings and excluding negative groundings. $\beta \rightarrow 0$ tend to include all positive groundings, while $\beta \rightarrow 1$ tend to exclude all negative groundings.

Now that we have obtained the vectorized representation of each variable in the rule body, we thus consider applying a binary mask \mathbf{b}_{x_i} (the constraint brought by branch atom $b(x_i)$) to the \mathbf{u}_{x_i} . This paper considers three types of branch atoms that are to be added to the rule body. As shown in Figure 2, suppose that we add branches to constrain the groundings of variable Y in the rule body:

- **AUX**. This type of branch atom yields a one-hop result from an auxiliary variable M , constraining Y to the entities that satisfy a certain relation. i.e. $b(Y) \Leftrightarrow r(M, Y)$. In this way, $\mathbf{b}_Y = \mathbf{1}^\top \mathbf{M}_r$, where $\mathbf{1}$ denotes a $|\mathcal{E}| \times 1$ column vector with all 1, which corresponds to the auxiliary M . For example, $b(Y) \Leftrightarrow \text{capital}(Y, M)$.
- **ENT**. This type of branch atom grounds a variable in the rule body to a unique entity. i.e. $b(Y) \Leftrightarrow \text{is}(e, Y)$. In this way, $\mathbf{b}_Y = \text{one-hot}(e)$. For example, $b(Y) \Leftrightarrow \text{Is}(Y, \text{Italy})$.
- **QRY**. This type of branch atom yields a one-hop result from the query variable X (i.e. x_0 in §2), constraining Y to the entities that have an additional relation to X . i.e. $b(Y) \Leftrightarrow r(X, Y)$. In this way, $\mathbf{b}_Y = \mathbf{v}_X \mathbf{M}_r$. For example, $b(Y) \Leftrightarrow \text{bornIn}(X, Y)$.

For each KG, the candidate constraint vectors of **AUX** and **ENT** can be obtained through preprocessing and stay fixed during the whole refinement process. For each rule, we can obtain the constraint vectors of **QRY** after the b initial groundings are sampled.

After obtaining the variable representations and constraint vectors of candidate atoms, the score of adding a branch atom $b(x_i)$ is defined by the inner product:

$$\text{score}(b(x_i)) = \mathbf{u}_{x_i} \mathbf{b}_{x_i}^\top \quad (10)$$

In the implementation, \mathbf{u}_{x_i} is multiplied with each type of candidate atoms and we greedily select the branch atoms with the top k scores for each variable appearing in the rule body.

4 Experiments

In this section, we conduct a series of experiments to evaluate and compare the refined tree-like rules

with original chain-like rules in the following two aspects:

- **Standard Confidence** (§4.5). We adopt the Standard Confidence as the direct metric to evaluate if the refined tree-like rules have better “quality” than the original chain-like rules.
- **Link Prediction** (§4.6). To further verify the effectiveness of the refined tree-like rules, we compare two types of rules on the task of Link Prediction and validate if tree-like rules conduct better reasoning than chain-like rules.

4.1 Datasets

We employ four commonly used Knowledge Graphs and their corresponding link prediction benchmarks: FB15k-237 (Toutanova and Chen, 2015), WN18RR (Dettmers et al., 2018), UMLS (Kok and Domingos, 2007), YAGO3-10 (Suchanek et al., 2007) for the evaluations.

4.2 Chain-like Rules

We adopt the following methods to mine chain-like rules to serve as the original chain-like rules for evaluation and refinement:

- **BBFS** We propose a bi-directional breadth-first search method to mine all chain-like rules within length n in KG as a basic search-based rule induction method.
- **AMIE** (Galárraga et al., 2013) AMIE is a classic symbol-based rule mining system. It learns chain-like rules by adding dangling atoms to the rule body sequentially while evaluating their coverage and confidence.
- **AnyBurl** AnyBurl (Meilicke et al., 2024) is a novel RL-based rule induction method and is currently one of the best symbolic rule reasoning methods competing with SOTA embedding reasoning approaches.

4.3 Implementations

For the tree-like rule refinement process, as it involves many multiplications of large and sparse matrices, we adopt the *torch.sparse* library to help us store and operate such matrices. For the evaluation process, to fairly evaluate the chain-like rules mined by each method and the tree-like rules refined by us, we adopt the toolkit of AnyBurl (Meilicke et al., 2024) to apply the learned rules to KGs and evaluate the link prediction results with the metrics MRR and Hit@n. As AnyBurl only originally supported chain-like rules, we modified the

Avg. sc Rule	FB15k-237			WN18RR		
	BBFS	AMIE	AnyBurl	BBFS	AMIE	AnyBurl
CHAIN	12.85	30.71	26.84	5.93	28.05	8.66
TREE(AUX)	21.24	41.42	27.38	15.85	42.12	-
TREE(ENT)	61.34	76.47	61.12	88.14	93.63	87.30
TREE(QRY)	40.88	56.66	47.57	61.71	59.98	62.88
TREE	35.96	56.63	43.04	55.06	73.64	85.43

Avg. sc Rule	UMLS			YAGO3-10		
	BBFS	AMIE	AnyBurl	BBFS	AMIE	AnyBurl
CHAIN	14.79	19.57	16.08	8.22	19.76	16.01
TREE(AUX)	19.86	33.29	26.05	15.56	27.60	19.09
TREE(ENT)	32.00	43.92	37.86	65.26	71.24	64.86
TREE(QRY)	25.65	39.51	32.19	41.08	44.48	47.23
TREE	24.48	38.24	30.93	37.17	49.18	42.38

Table 1: The average Standard Confidence of different rules on FB15k-237. CHAIN denotes the original chain-like rules mined by each method. AUX, ENT, and QRY denote the three kinds of branch atoms in §3.3. TREE denotes the refined tree-like rules. - denotes there are no such type of rules refined.

AnyBurl toolkit to make it compatible with tree-like rules.

4.4 Setups

For each rule, we sample $b = 100$ entities to ground the query variable X and conduct the forward & backward reasoning process. We set β to the Standard Confidence sc of the original chain-like rule \mathcal{R} . For each variable, branch atoms with top $k = 5$ scores are selected to refine the rule. The chain-like rules are all within length $n = 3$. The random seed is fixed to 37.

4.5 Standard Confidence

We first verify if the refined tree-like rules are actually more precise, i.e. have higher Standard Confidence than the original chain-like rules. As shown in Table 1, The refined tree-like rules (TREE) consistently have significant higher average Standard Confidence than the original chain-like rules (CHAIN). These results verify that our proposed refinement method effectively refine optimize the initial objective in Eq 1. Moreover, we can observe that among the tree types of proposed candidate branch atoms, the ranking of their Standard Confidence is $ENT > QRY > AUX$, indicating that their constraining strength as branch atoms weakens in this order, which aligns with our intuition.

4.6 Link Prediction

To further verify the effective of the refined tree-like rules in the KG reasoning, we evaluated the link prediction performances of chain-like rules from all induction methods and their refined tree-like rules on all four different KGs. As shown in Table 2, the refined tree-like rules consistently outperform the original chain-like rules induced by different methods on different KGs. On the FB15k-237 and UMLS datasets, the refinement of chain-like rules into tree-like rules exhibits a performance gain of more than 2% in most cases. Notably, on the UMLS dataset, tree-like rules demonstrate a significant out-performance compared to Anyburl chain-like rules, with an impressive 7.79% improvement in MRR. These results lead us to the conclusion that our framework effectively refines chain-like rules from different rule-mining methods into higher-quality tree-like rules on different knowledge graphs, thereby yielding superior reasoning outcomes.

4.7 Performance Analysis

From Table 2, we can also observe that the improvements (delta values) vary across different knowledge graphs. We explain this based on the topological structure and density of different knowledge graphs. Intuitively, the sparser the knowledge graph, the simpler the abstract structure it implies (tending towards simple chain-like rules), while the denser the knowledge graph, the more complex

Dataset		FB15k-237				WN18RR			
Rule		MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10
BBFS	CHAIN	24.32	18.38	25.97	36.65	39.29	37.94	40.08	42.02
	TREE	27.32	21.05	29.60	39.39	40.04	38.77	41.00	42.41
	Δ	+3.00	+2.67	+3.63	+2.74	+0.75	+0.83	+0.92	+0.39
AMIE	CHAIN	22.60	17.25	24.27	33.78	36.21	36.06	36.31	36.47
	TREE	25.70	20.20	27.93	36.56	36.24	36.08	36.37	36.50
	Δ	+3.10	+2.95	+3.66	+2.78	+0.03	+0.02	+0.06	+0.03
AnyBurl	CHAIN	32.74	23.94	35.75	50.98	48.42	44.22	50.99	56.03
	TREE	35.05	26.52	38.34	52.42	48.98	45.27	51.16	55.83
	Δ	+2.31	+2.58	+2.59	+1.44	+0.56	+1.05	+0.17	-0.20
Dataset		UMLS				YAGO3-10			
Rule		MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10
BBFS	CHAIN	75.13	65.17	82.29	91.33	53.47	47.56	58.34	63.32
	TREE	77.84	68.20	85.45	92.98	54.68	49.18	59.42	63.72
	Δ	+2.71	+3.03	+3.16	+1.65	+1.21	+1.62	+1.08	+0.40
AMIE	CHAIN	39.17	32.16	45.10	50.83	52.07	46.68	57.08	60.74
	TREE	42.08	35.94	46.97	51.90	53.02	48.08	57.62	60.74
	Δ	+2.91	+3.78	+1.87	+1.07	+0.95	+1.40	+0.54	+0.00
AnyBurl	CHAIN	69.64	55.85	79.60	92.25	63.07	57.34	67.30	72.10
	TREE	77.43	66.73	85.94	94.67	63.38	57.82	67.36	72.38
	Δ	+7.79	+10.88	+6.34	+2.42	+0.06	+0.48	+0.06	+0.28

Table 2: The link prediction performance of different rules on four KGs. CHAIN denotes the original chain-like rules mined by each method. TREE denotes the refined tree-like rules. Δ denotes the improvements.

KG	FB15k-237	WN18RR	UMLS	YAGO3-10
density	2.59e-03	1.06e-04	2.20e-01	1.42e-04
avg Δ MRR	2.80	0.61	4.47	0.74

Table 3: The edge density and the average Δ MRR brought by tree-like rules of each KG.

the abstract structure it implies (tree-like rules will have an advantage in reasoning).

From a qualitative perspective, WN18RR and YAGO3-10, with smaller deltas, are subsets of WN18 (which is also a subset of WordNet (Fellbaum, 2010)) and YAGO3, respectively. FB15k-237, with a moderate delta, is a larger subset of FB15k and is relatively denser. These three knowledge graphs have suffered varying degrees of information loss as they are sampled from the whole knowledge graphs. UMLS, on the other hand, is an unfiltered and complete knowledge graph, and therefore retains the most complete information.

From a quantitative perspective, we calculated the density of the four knowledge graphs used in the experiments using edge density (number of edges / the number of possible edges) and the average Δ MRR brought by tree-like rules, and the results are shown in Table 3. We can see that in

knowledge graphs with higher density, the reasoning gain brought by tree-like rules is greater, and the Pearson correlation coefficient between them is 0.844. These verify that the density of the used knowledge graph and the improvement brought by tree-like rules are positively correlated.

4.8 Case Study

To better compare and present the tree-like rules refined from chain-like rules, we provide two specific examples from YAGO3-10 in Figure 3. It can be observed that the original chain-like rules, constrained by the semantic expressiveness of their chain structure, tend to produce a large number of factually incorrect groundings, resulting in lower standard confidence. In contrast, the tree-like rules refined by our method address the semantic gaps of the original rules in various ways, thereby achieving higher standard confidence.

5 Related Work

Rule induction over knowledge graphs is a classical yet challenging task. Inductive Logic Programming (ILP) seeks to induce the symbolic pattern behind the knowledge graphs. It faces the combinatorial

Examples of Rule Refinement

Chain-like Rule 1

Standard Confidence: 0.39

$\text{isLocatedIn}(X, Y) \Leftarrow \text{hasCapital}(X, Y)$

Refined Tree-like Rules:

1. **Standard Confidence:** 1.00
 $\text{isLocatedIn}(X, Y) \Leftarrow \text{hasCapital}(X, Y), \text{hasOfficialLanguage}(Y, M)$
2. **Standard Confidence:** 0.50
 $\text{isLocatedIn}(X, Y) \Leftarrow \text{hasCapital}(X, Y), \text{hasWonPrize}(Y, M)$
3. **Standard Confidence:** 1.00
 $\text{isLocatedIn}(X, Y) \Leftarrow \text{hasCapital}(X, Y), \text{is}(\text{Gangtok}, Y)$

Chain-like Rule 2

Standard Confidence: 0.11

$\text{worksAt}(X, Y) \Leftarrow \text{hasAcademicAdvisor}(X, A), \text{graduatedFrom}(A, Y)$

Refined Tree-like Rules:

1. **Standard Confidence:** 0.17
 $\text{worksAt}(X, Y) \Leftarrow \text{hasAcademicAdvisor}(X, A), \text{graduatedFrom}(A, Y), \text{influences}(A, X)$
2. **Standard Confidence:** 0.33
 $\text{worksAt}(X, Y) \Leftarrow \text{hasAcademicAdvisor}(X, A), \text{graduatedFrom}(A, Y), \text{is}(\text{University_of_Cambridge}, Y)$
3. **Standard Confidence:** 0.20
 $\text{worksAt}(X, Y) \Leftarrow \text{hasAcademicAdvisor}(X, A), \text{graduatedFrom}(A, Y), \text{owns}(Y, M)$

Figure 3: Examples of the refinement of chain-like rules into tree-like rules from YAGO3-10, along with their respective Standard Confidences.

space of searching predicates and binding variables. Traditional ILP methods including AMIE (Galárraga et al., 2013), AMIE+ (Galárraga et al., 2015) and RLvLR (Omran et al., 2018) used search-based methods to induce chain-like rules. Recently, many works have studied the way of inducing chain-like rules in a differentiable approach, named Differentiable ILP, like NeuralLP (Yang et al., 2017), NLM (Dong et al., 2019), DLM (Zimmer et al., 2021), NLIL (Yang and Song, 2020).

However, only a few works considered the defect of expressive power chain-like rules. NLIL (Yang and Song, 2020) induced conjunctions of chain-like rules like $\text{Car}(X) \Leftarrow \text{Wheel}(Y_1) \wedge \text{Of}(Y_1, X) \wedge \text{Window}(Y_2) \wedge \text{Of}(Y_2, X)$, but they are only tree-like rules with branches at the target variable X . TyRule (Wu et al., 2022) proposed to learn typed rules with type predicate type _{n} (x_i)

adding to each variable, but it needs extra type information of the knowledge graphs. In this paper, we propose the concept of tree-like rules and an effective framework for refining chain-like rules into tree-like rules to reach better reasoning ability.

6 Conclusion

This paper introduces the concept of tree-like rules and presents an effective framework for refining chain-like rules into tree-like rules. To verify the effectiveness of the tree-like rules refined by our framework, this paper carried out experiments to show that the refined tree-like rules consistently exhibit higher Standard Confidence and outperform the original chain-like rules on KG reasoning tasks. Further analysis illustrates that the improvements brought by the tree-like rules are positively correlated with the the density of the KGs.

7 Acknowledgments

This work was supported by the National Key R&D Program of China (No. 2022ZD0160503) and Beijing Natural Science Foundation (L243006). This work was also supported by the National Natural Science Foundation of China (No.62376270).

References

- Pei Chen, Wenlin Yao, Hongming Zhang, Xiaoman Pan, Dian Yu, Dong Yu, and Jianshu Chen. 2022. [Zerokbc: A comprehensive benchmark for zero-shot knowledge base completion](#). In *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1–6.
- Kewei Cheng, Nesreen K Ahmed, and Yizhou Sun. 2023. Neural compositional rule learning for knowledge graph reasoning. *arXiv preprint arXiv:2303.03581*.
- Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. 32(1).
- Honghua Dong, Jiayuan Mao, Tian Lin, Chong Wang, Lihong Li, and Denny Zhou. 2019. Neural logic machines. *arXiv preprint arXiv:1904.11694*.
- Christiane Fellbaum. 2010. Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer.
- Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M Suchanek. 2015. Fast rule mining in ontological knowledge bases with amie+. *The VLDB Journal*, 24(6):707–730.
- Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. 2013. [AMIE: association rule mining under incomplete evidence in ontological knowledge bases](#). pages 413–422.
- Stanley Kok and Pedro Domingos. 2007. Statistical predicate invention. pages 433–440.
- Christian Meilicke, Melisachew Wudage Chekol, Patrick Betz, Manuel Fink, and Heiner Stuckeschmidt. 2024. Anytime bottom-up rule learning for large-scale knowledge graph completion. *The VLDB Journal*, 33(1):131–161.
- Ananjan Nandi, Navdeep Kaur, Parag Singla, et al. 2023. Simple augmentations of logical rules for neuro-symbolic knowledge graph completion. pages 256–269.
- Pouya Ghiasnezhad Omran, Kewen Wang, and Zhe Wang. 2018. Scalable rule learning via learning representation. pages 2149–2155.
- Meng Qu, Junkun Chen, Louis-Pascal A. C. Xhonneux, Yoshua Bengio, and Jian Tang. 2021. [Rnnlogic: Learning logic rules for reasoning on knowledge graphs](#).
- Stuart J Russell and Peter Norvig. 2016. *Artificial intelligence: a modern approach*. Pearson.
- Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. pages 697–706.
- Ben Taskar, Ming-Fai Wong, Pieter Abbeel, and Daphne Koller. 2003. Link prediction in relational data. *Advances in neural information processing systems*, 16.
- Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. pages 57–66.
- Jiapu Wang, Boyue Wang, Meikang Qiu, Shirui Pan, Bo Xiong, Heng Liu, Linhao Luo, Tengfei Liu, Yongli Hu, Baocai Yin, et al. 2023. A survey on temporal knowledge graph completion: Taxonomy, progress, and prospects. *arXiv preprint arXiv:2308.02457*.
- Hong Wu, Zhe Wang, Kewen Wang, and Yi-Dong Shen. 2022. Learning typed rules over knowledge graphs. 19(1):494–503.
- Fan Yang, Zhilin Yang, and William W Cohen. 2017. Differentiable learning of logical rules for knowledge base reasoning. *Advances in neural information processing systems*, 30.
- Yuan Yang and Le Song. 2020. [Learn to explain efficiently via neural logic inductive learning](#).
- Matthieu Zimmer, Xuening Feng, Claire Glanois, Zhao-hui Jiang, Jianyi Zhang, Paul Weng, Li Dong, Hao Jianye, and Liu Wulong. 2021. Differentiable logic machines. *arXiv preprint arXiv:2102.11529*.

LAB-KG: A Retrieval-Augmented Generation Method with Knowledge Graphs for Medical Lab Test Interpretation

Rui Guo^{1,2}, Barry Devereux², Greg Farnan¹, Niall McLaughlin²,

¹Cirdan, ²Queen’s University Belfast

Correspondence: rui.guo@cirdan.com

Abstract

Laboratory tests generate structured numerical data, which a clinician must interpret to justify diagnoses and help patients understand the outcomes of the tests. LLMs have the potential to assist with the generation of interpretative comments, but legitimate concerns remain about the accuracy and reliability of the generation process. This work introduces LAB-KG, which conditions the generation process of an LLM on information retrieved from a knowledge graph of relevant patient conditions and lab test results. This helps to ground the text-generation process in accurate medical knowledge and enables generated text to be traced back to the knowledge graph. Given a dataset of laboratory test results and associated interpretive comments, we show how an LLM can build a KG of the relationships between laboratory test results, reference ranges, patient conditions and demographic information. We further show that the interpretive comments produced by an LLM conditioned on information retrieved from the KG are of higher quality than those from a standard RAG method. Finally, we show how our KG approach can improve the interpretability of the LLM generated text.

1 Introduction

Artificial Intelligence (AI) has become increasingly influential in the medical field, offering transformative potential in various applications such as medical data summarisation (Van Veen et al., 2024) and diagnostics (Tu et al., 2024). The data generated in clinical care, from Electronic Health Records (EHRs) to laboratory tests, present both an opportunity and a challenge. In principle, using such data efficiently and intelligently has the potential to create efficiencies for healthcare professionals which allow them to improve patient experiences and outcomes. Laboratory diagnostics generate substantial amounts of structured numerical data, which can be difficult for patients and clinicians to inter-

pret effectively. AI models have the potential to provide interpretative comments and personalised explanations of laboratory results, improving the laboratory-clinical interface, and improving patient understanding (Padoan and Plebani, 2022a,b).

However, there are critical considerations when using AI models in the medical domain, including issues such as hallucinations, inaccuracies, and non-determinism. These issues can lead to incorrect or harmful results in healthcare (Cadamuro et al., 2023; Stevenson et al., 2024), and the errors can often be difficult to identify during model evaluation and to characterize *a priori*. These problems call for approaches to improve the reliability and accuracy of AI systems in medicine.

Integrating Knowledge Graphs (KGs) with LLMs through Retrieval-Augmented Generation (RAG) can be a promising strategy. KGs provide structured, interconnected data that can ground LLM outputs in factual information, reducing hallucinations, and improving the accuracy of AI-generated content (Yan et al., 2024; Gilbert et al., 2024). By combining the LLM’s generative capabilities with the KG’s factual grounding, AI systems can be more reliable and explainable.

In this work, we aim to improve laboratory test interpretation generation by combining RAG with a Knowledge Graph, referred to as the LAB-KG approach. Traditional RAG methods rely on embedding similarity between the user’s query and a set of documents or knowledge base entries. They retrieve relevant information to condition the language model’s generation process. However, the reasoning behind the generated interpretations often remains a black box. Our LAB-KG approach uses both the internal knowledge of LLMs and lab test examples to build a knowledge graph that explicitly captures the relevance between each test result and the patient’s condition. This allows for more explainable and transparent interpretation generation.

Our contributions are threefold:

1. **Knowledge Graph Construction with Limited Examples:** We present a novel approach for building a Knowledge Graph (KG) utilising the internal knowledge of Large Language Models (LLMs) and a limited set of laboratory test examples, capturing the relationships between test results and medical conditions.
2. **Improved Performance over Retrieval-Augmented Generation (RAG):** Our KG-based approach demonstrates better performance compared to traditional Retrieval-Augmented Generation methods. By structurally representing knowledge, the system can more accurately interpret and retrieve relevant conditions from new patient test results.
3. **Explainable System:** The proposed KG approach offers greater interpretability than standard RAG methods. The explicit structure of the KG allows for the tracing of errors in generated reports back to specific nodes and relationships within the graph.

2 Previous Work

The application of AI to the task of laboratory test interpretation is an area of growing interest. Traditional methods of providing interpretative comments on laboratory reports have been recognised as essential to improving the laboratory-clinical interface (Plebani, 2009).

Several studies have been applied to use AI and natural language processing models to interpret laboratory test results. Cadamuro et al. (2023) evaluated the performance of ChatGPT and other AI models in understanding laboratory medicine test results. Whilst the AI models could recognise laboratory tests and detect deviations from reference intervals, their interpretations were often superficial and incorrect. The models sometimes failed to differentiate between slight and severe deviations and did not provide meaningful suggestions for follow-up diagnostics.

Stevenson et al. (2024) evaluated the thyroid function test result interpretation by biochemist, ChatGPT, and Google Bard. The AI tools correctly interpreted only a fraction of the cases, showing the limitations of current AI models in complex medical interpretation tasks.

Abusoglu et al. (2024) assessed the performance of various chatbots as assistants for problem-

solving in clinical laboratories. Their study showed that AI applications had good performance in identifying cases and responding to questions related to preanalytical, analytical, and postanalytical errors. However, the chatbots' accuracy varied, and there were concerns about their reliability and safety in clinical settings.

An early work by Patil et al. (2013) introduced a Concept Graph Engine (CG-Engine) that generates patient-specific personalised disease rankings based on laboratory test data, using the Unified Medical Language System (UMLS) as a medical knowledge base. The CG-Engine constructs a concept graph connecting laboratory tests to diseases and computes weights based on relation types, semantic types, and other attributes. While their approach utilises a knowledge base to connect lab tests and conditions, it relies on pre-existing medical ontologies that may differ from the actual data terminology.

Despite these advancements, a major challenge with LLMs in the medical domain is their tendency to produce hallucinations and inaccurate information. Retrieval-Augmented Generation (RAG) techniques have been proposed to mitigate these issues, where LLMs are augmented with external knowledge sources to ground their outputs in factual data. Zakka et al. (2024) developed *Almanac*, an LLM framework augmented with retrieval capabilities from curated medical resources for medical guidelines and treatment recommendations. Their results showed significant performance improvements compared to standard LLM pipelines.

In the domain of laboratory test interpretation, He et al. (2023) built a dataset by collecting and annotating interpretations of textual lab results from health articles. They evaluated transformer-based language models for recognizing key terms and mapped them to concepts in major controlled terminologies.

In healthcare generally, integrating LLMs with Knowledge Graphs can improve the reliability and accuracy of AI models. Gilbert et al. (2024) discussed the potential of combining LLMs with KGs as medical information curators. By providing a structured representation of medical knowledge, KGs can help LLMs generate more accurate and verifiable outputs, reducing the risk of misinformation and enhancing patient safety.

Our work builds upon these approaches by constructing a knowledge graph that combines LLM internal knowledge with examples to associate lab

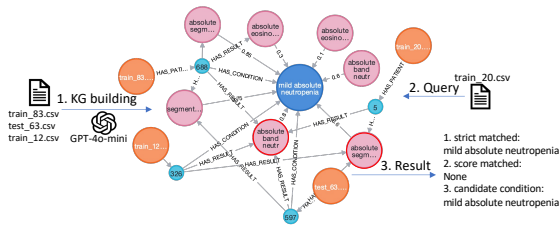


Figure 1: Overview of LAB-KG. GPT-4o-mini helps to find the relationship between the test result and condition. When a new patient’s test results are input into the system, they are compared with the LAB-KG to identify relevant conditions using strict set matching and confidence score matching. This process enables the generation of accurate and explainable interpretations of laboratory test results.

tests with conditions. This allows for improved accuracy in lab test interpretation generation and provides explainability through the graph structure.

3 Method

Given a set of patient full blood test csv files, we build a LAB-KG with the help of GPT-4o-mini to find the relationships between the test results and conditions. A condition in our context refers to a specific medical finding or diagnosis derived from laboratory test results. For instance, “Mild normochromic normocytic anaemia” indicates a type of anemia characterized by red blood cells that are of normal size (normocytic) and normal hemoglobin content (normochromic). Clinicians use those conditions to determine the appropriate follow-up and management for the patient. The new patient test result is compared with the LAB-KG to find the relevant condition. An overview of this process is in Figure 1. An example of a transcribed report is shown in Table 1.

3.1 KG-RAG Approach

We propose an approach combining both the internal knowledge of a large language model and limited examples to build a knowledge graph. A laboratory test is a medical procedure using a sample of blood, urine, or other tissues to assess a patient’s health. Interpreting lab test results can be complex due to the subtle variations that may indicate different medical conditions. Our knowledge graph (KG) represents relationships between lab tests, conditions, and patients and can be queried to generate interpretations for new patients. The relation between lab tests and conditions are built as in the method described below.

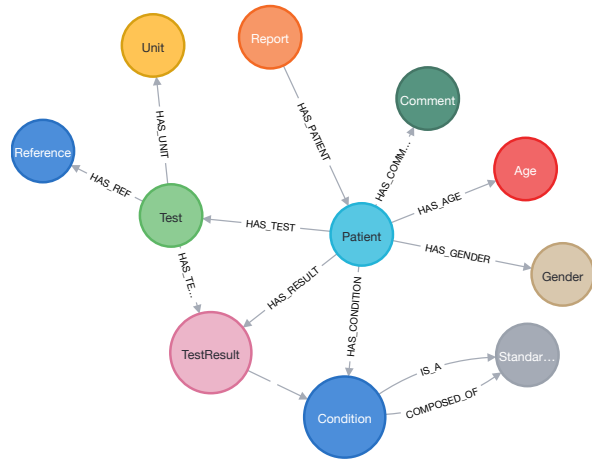


Figure 2: The schema of the lab-kg.

3.1.1 Graph Building

The knowledge graph is constructed to model the relationships between lab tests, patients, conditions, and results. The schema of the graph is shown in Figure 2. The key nodes and relationships are summarised in Table 4 in appendix.

Reference Ranges A reference range is the set of values considered normal for a healthy individual for a specific test, serving as a benchmark to interpret individual test results. The reference ranges for some tests are sometimes missing, and to address this issue, we aggregated all the reports’ reference. We ask LLM to infer the correct reference range by providing all the related reference ranges for that test and asking the LLM to use its internal medical knowledge for the patient.

Test Names and Test Result Test names are the standardised identifiers used to represent specific laboratory tests. The test names in our dataset are standardised by curating a list of test names and manually mapping different variations to a standard name. Each *TestResult* node represents the result of a specific test. If the reference range is provided, the test result will be labelled with a suffix indicating its status (e.g., *Normal*, *Abnormal (High)*, *Abnormal (Low)*, *Borderline (High)*, *Borderline (Low)*).

Condition Extraction The most important task for LLM is extracting the conditions from the comments and determining the relevance of each test result to the conditions mentioned in the patient comments. We prompted the LLM to split the comment into several conditions and establish potential *CONTRIBUTES_TO* relationships between each test result node and condition node. This effectively

Category	Test Name	Result	Unit	Ref Start	Ref End	norm	Ab flag
Info	Age	9					
Blood	Haemoglobin	11.30	g/dL	11.5	15.5	-0.05	Low
Blood	Hematocrit	33.9	%	35	45	-0.11	Low
Blood	Red cell count	4.71	x10 ⁶ /uL	4	5.2	0.59	
Blood	MCV	72.0	fL	78	96	-0.33	Low
Blood	MCH	24.0	pg	26	32	-0.33	Low
Blood	MCHC	33.4	g/dL	31	36	0.48	
Blood	RDW	14.2	%	11.5	14.5	0.9	
Blood	Platelet Count	292	x10 ³ /uL	170	450	0.44	
Blood	T.L.C	8.2	x10 ³ /uL	5	13	0.4	
WBC Diff	Basophils	1	%	0			
...	...						
WBC Diff	Monocytes (Absolute)	1.1	x10 ³ /uL	0.2	1	1.12	High
Comments	Mild microcytic hypochromic anaemia. Platelets are adequate. Mild absolute monocytosis.						

Table 1: Patient report example (Abridged). Ab flag: abnormality flag.

builds a rule set based on the examples and the LLM’s knowledge. For example, “Mild normocytic normochromic anemia with mild anisocytosis” can be split into two conditions: “Mild normocytic normochromic anemia” and “mild anisocytosis.” We only ask LLM to infer that *CONTRIBUTES_TO* relationship from the abnormal conditions to test results, and omit the conditions such as “normal blood picture” or “follow up is recommended”, which cannot be mapped to a set specific test result.

Knowledge Aggregation We added an aggregation stage where we asked the LLM to assign weights to each relationship between a test result and a condition identified by the LLM. First, we added a *StandardTerm* node to group different conditions with potential semantic similarity. This grouping is based on querying each condition name using the BioPortal API for standardised terms, prioritizing matches in ontologies such as SNOMED CT, LOINC, and MEDDRA. In this way, we can group conditions under the same standard term, such as “mild anaemia” and “moderate anaemia” both being under the standard term “anaemia.” Then, by providing all the *CONTRIBUTES_TO* in the KG between a condition group and related *TestResult*, we aim for the LLM to use these examples to indicate the importance of each test result for a particular condition group by assigning a weight to each *CONTRIBUTES_TO* relationship. This weight-assigning stage uses the aggregation *CONTRIBUTES_TO* from a condition with

the frequency of each test result and the patient age/gender distribution.

3.2 Graph Retrieval Process

The KG is queried to find candidate conditions for a new patient. We tested three methods to find relevant conditions: an example-based match, a confidence score ranking, and their combination.

We first identify abnormal test results for a new patient and retrieve the connected *Condition* nodes for any abnormal tests in that patient, creating a list of candidate conditions. The connected patients and their related test results for each potential condition are retrieved from there.

Not all test results connected to a condition are critical; some might be false positives or less relevant. To filter less important conditions, we use two methods to select potential conditions.

Strict Match For each condition, we compare the test results of the new patient to those of the retrieved patients. Suppose the test results of the new patient cover all the test results of one patient in the training dataset connected to that condition (here, *Borderline* and *Abnormal* are treated the same). We consider it a “strict match” for that condition. An example is illustrated in Figure 3, with the condition “mild normochromic normocytic anemia.” The new patient (with id 938) matches most of the test results of an existing patient (with id 100) but lacks “RBC count Abnormal (Low).” In this case, the new patient will not be assigned to this condition based on strict match. Note that there are other test results related to the condition without

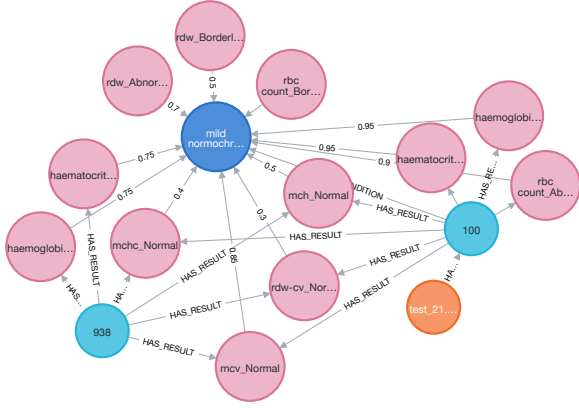


Figure 3: The condition query process for a new patient. The patient with ID 938 lacks “RBC count Abnormal (Low)” compared to the example patient with ID 100 using strict test result matching.

Test result	Weight
haematocrit_Abnormal (Low)	0.95
haemoglobin_Abnormal (Low)	0.95
rbc count_Abnormal (Low)	0.9
mcv_Normal	0.85
haemoglobin_Borderline (Low)	0.75
haematocrit_Borderline (Low)	0.75
rdw_Abnormal (High)	0.7
rbc count_Borderline (Low)	0.6
rdw_Borderline (High)	0.5
mch_Normal	0.5
mchc_Normal	0.4
rdw-cv_Normal	0.3

Table 2: The weight for “mild normochromic normocytic anemia” assigned by LLM

any patient connected, due to additional borderline connections added with slightly lower weights than abnormal, or because there are existing patients in the same condition group with those test results.

Confidence score-based match We utilise the weight assigned on the *CONTRIBUTES_TO* relationship to calculate each condition’s confidence score by normalizing the weights connected to that patient for each condition. We sum the weights of the test results in the patient connected to one condition, and divide that sum by the total weight of all test results linked to that condition. A detailed example of a confidence score match is in the Appendix. The threshold to filter the confidence score is decided by the performance of the training data, as explained in section 4.

After the candidate conditions were retrieved

from graph retrieval, we added an optional finalising stage using LLM to refine the conditions given the candidate conditions, merging potential duplicates or selecting the most specific condition rather than a broader one.

A key advantage of our LAB-KG approach is its inherent explainability addressing the limitations of traditional AI models in laboratory test interpretation. When generating interpretations for a new patient, clinicians can examine the specific test results leading to each suggested condition, along with the associated weights and confidence scores. This allows the clinicians to understand which conditions are being suggested and the rationale behind them. For instance, if a condition is identified, clinicians can review the exact match of test results between the new patient and existing examples and the weights of individual test results contribute to the overall confidence score.

Explainability Example The Knowledge Graph (KG) provides a transparent means to explain why each condition is retrieved, allowing us to identify and correct errors by examining the relationships between conditions and test results. As an illustrative example, consider the case of a patient diagnosed with “mild microcytosis,” depicted in Figure 4. Initially, the KG connected both low Mean Corpuscular Hemoglobin (MCH) and low Mean Corpuscular Volume (MCV) to “mild microcytosis,” even though low MCV alone is sufficient to diagnose microcytosis. When querying a new patient (ID 283) who exhibited low MCV but not low MCH, the system failed to retrieve “mild microcytosis” because the KG’s connections implied that both low MCH and low MCV were required for retrieval. Upon reviewing the definition of “mild microcytosis,” we corrected the KG by removing the redundant connection between low MCH and “mild microcytosis.” After this, the system successfully retrieved “mild microcytosis” for the patient, demonstrating how the explainability provided by the KG facilitates refinement and improves retrieval accuracy.

4 Implementation and Evaluation

To the best of our knowledge, there are very few publicly accessible datasets providing detailed laboratory test reports along with associated clinical interpretations. We utilised a dataset from Mendelley Data (Abdelmaksoud et al., 2022), which includes 260 clinical laboratory test reports issued by

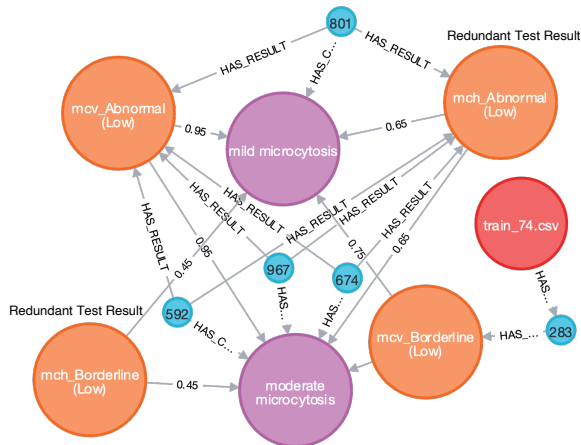


Figure 4: Illustration of explainability in the KG. Initially, “mild microcytosis” was connected to both “low MCH” and “low MCV.” When querying patient ID 283, “mild microcytosis” was not retrieved because the KG incorrectly required both “low MCH” and “low MCV” for retrieval. After removing the unnecessary connection to “low MCH,” “mild microcytosis” was successfully retrieved for the patient.

24 Egypt laboratories covering several test types. Among these, blood tests constitute the majority. We used GPT-4o¹ to transcribe all the blood test reports from PDF to CSV format. After removing duplicates, we obtained 47 unique blood test reports.

We used the Cypher query language in Python to build the KG in Neo4j Community Edition. GPT-4o-mini was used as the default LLM. Once the graph was built on training examples, it included 38 patient examples, 37 conditions, 78 test results, 459 nodes and 2287 relationships.

Our evaluation metrics include MEDCON (Yim et al., 2023), BLEU-3 (Papineni et al., 2002), ROUGE-1/2/L (Lin, 2004), BERTScore (Zhang et al., 2019), METEOR (Banerjee and Lavie, 2005), recall, precision, and F1 score. The LLM helps to preprocess the condition result, aligning semantically equivalent conditions (e.g., “mild anemia” vs. “anemia”) between the generated and target, so that the extracted sets of conditions are comparable. The calculation of recall and precision itself remains a standard statistical comparison after the conditions have been extracted and aligned by the LLM. The F1 score is based on recall and precision. We use BLEU-3 instead of BLEU-4 because the results can be very short, such as “normal blood picture,” which BLEU-4 would omit.

¹<https://platform.openai.com/docs/models>

MEDCON is a metric for evaluating medical condition extraction from generated texts, considering semantic similarity and clinical relevance. The KG and KG with *CONTRIBUTES_TO* relationships inferred without examples (referred to KG * below) are compared for those metrics, together with other methods listed below.

In all our experiments, we performed five-fold cross-validation, with test data sizes of 10, 10, 9, 9, and 9 in each fold. We use MEDCON to select the threshold for the confidence score in each fold and from a list of values ranging from 0.1 to 0.9, with step 0.05. The best threshold values are stable across folds (0.55, 0.6, 0.55, 0.5, 0.5 for KG and 0.45, 0.35, 0.35, 0.35, 0.35 for KG *). The median values 0.55 and 0.35 are selected as the final threshold values for all the folds in KG and KG * respectively.

We compared the performance of the LAB-KG approach with several baseline methods, including:

1. Prompt Engineering

A detailed prompt was designed to output different conditions given the patient report, which is a textual representation of each CSV file.

2. Text Embedding-Based Retrieval

This method relies purely on text embeddings to retrieve relevant interpretations. The eight most similar examples are provided to the LLM for few-shot learning (we selected eight by testing numbers from 1 to 8). The query is the document of the new patient without the comment row. HuggingFace’s all-MiniLM-L6-v2 model embeds the text. We tested different document components in the retrieval and generation stages, including:

- (A) Using all the rows.
- (B) Using only the abnormal rows.
- (C) Adding the normalised value as a column.

An exhaustive search of all possible component combinations in the retrieval/generation stages is infeasible, so we tested four configurations using the same components for both retrieval and generation stages, and two configurations using different elements, totalling six results, as described in Table 3. The input examples include the above components and

the final comment, which the LLM may use as context to align its knowledge to the format and content of the example output comment.

3. LAB-KG Built with Examples (KG)

We evaluated the results of the strict match, the confidence score match, and the combination of the strict and confidence score match. We also tested the effect of using the LLM to finalise the result.

4. Finding the relationship between *TestResult* and *Condition* without examples (KG *)

An approach using the LLM’s internal knowledge only to infer the *CONTRIBUTES_TO* relationships between test result nodes and condition nodes. We aim to assess the LLM’s ability to find these relationships without examples. Based on the KG built with examples, the *CONTRIBUTES_TO* relationships are removed first. Then, for each condition, the LLM is provided with all possible test results to find the relationships and assign weights based on its own knowledge. New test result nodes can be created in this case.

5. Random Forest

This traditional machine learning classifier was trained to predict the conditions given the patient data. Two kinds of inputs are tested: one with test results categorised as inputs (e.g., *Haemoglobin Abnormal (High)*), and another using the numerical test values directly. The conditions are classified, and adjectives such as “mild” and “moderate” are removed to reduce the possible classes to predict.

To determine whether using the LLM to evaluate the results is reliable, the correlation between F1 and each metric is shown in Table 5 in appendix. The F1 score has the highest correlation with MEDCON (0.95) and Bert score (0.94), and the correlation for the KG without examples is MEDCON (0.97) and Bert score (0.91). This validates the LLM’s alignment between the generated and target results.

The results are presented in Table 3. The results show that combining the LLM’s internal knowledge and examples can most effectively utilise the LLM and data, with an F1 score of 0.76, higher than the best KG * result of 0.71. The RAG approach has a best F1 score of 0.56, much lower

than the best KG retrieval approach. When using the strict match, because it is based on the occurrence of test results in the examples, KG AND KG * show little difference. The combination of result of strict match and confidence score based match achieved higher score than separate result for KG, however, the combination of result for KG * is worse. The finalisation step does not make the result much different for the F1 score.

A detailed example about the difference in the result using KG and KG * in the appendix. All the LLM generated interpretation and the calculated metrics can be downloaded at <https://docs.google.com/spreadsheets/d/10YTnKbLUs9UAVGACH3wcNt-erMBLpJI>

5 Conclusion

In this paper, we integrate the knowledge graph with RAG and LLM to improve the interpretation of laboratory test results with limited examples, providing an explainable framework clinicians can understand.

The evaluation demonstrated that the LAB-KG method outperforms LLM prompt engineering, text embedding-based retrieval, and random forests. The combination of strict matching and confidence score-based matching with KG allows us to retrieve the most clinically relevant interpretations. The KG with the relationship between condition and test result built without examples also performs well, especially in the strict match case, demonstrating its accurate internal knowledge.

We observed that in some cases, the relevance inferred by the LLM without examples was better than when examples were provided. This suggests the potential to combine the LLM’s internal knowledge more effectively with examples to optimise performance. When multiple conditions are present in the example, the LLM sometimes struggles to differentiate the test results associated with each condition. Providing separate conditions in examples or generating synthetic data could help mitigate this issue.

The strength of our findings may be limited with only 47 blood test reports. Expanding the dataset and applying LAB-KG to other laboratory tests are essential steps for validating LAB-KG. The LLM’s internal knowledge may not be up to date and limited, and integrating our knowledge graph with external medical ontologies like SNOMED CT is for future exploration.

category		medcon	bleu	bert score	meteor	rouge1	rouge2	rougeL	recall	precision	f1
zero shot	PE only	0.28	0.16	0.5	0.25	0.31	0.15	0.3	0.45	0.44	0.38
RAG	Q=G=A	0.39	0.29	0.6	0.44	0.47	0.32	0.45	0.56	0.46	0.48
	Q=G=A,C	0.38	0.2	0.59	0.39	0.42	0.23	0.39	0.56	0.42	0.46
	Q=G=B	0.49	0.29	0.64	0.47	0.5	0.32	0.48	0.68	0.5	0.56
	Q=G=B,C	0.47	0.28	0.63	0.46	0.48	0.3	0.47	0.67	0.51	0.56
	Q=A,C G=B,C	0.46	0.29	0.64	0.45	0.49	0.3	0.47	0.67	0.52	0.56
	Q=B G=B,C	0.48	0.29	0.64	0.47	0.49	0.31	0.48	0.67	0.51	0.56
KG retrieval	strict	0.68	0.37	0.71	0.56	0.56	0.43	0.53	0.78	0.65	0.67
	score	0.67	0.29	0.67	0.5	0.51	0.39	0.45	0.73	0.65	0.66
	strict + score	0.75	0.36	0.73	0.58	0.58	0.47	0.53	0.88	0.73	0.76
KG * retrieval	strict	0.67	0.3	0.7	0.51	0.54	0.38	0.5	0.78	0.65	0.68
	score	0.58	0.23	0.63	0.45	0.45	0.35	0.41	0.74	0.58	0.62
	strict + score	0.6	0.25	0.65	0.49	0.47	0.36	0.44	0.81	0.59	0.66
KG retrieval + finalise	strict	0.64	0.41	0.73	0.56	0.61	0.46	0.56	0.73	0.7	0.67
	score	0.59	0.28	0.67	0.49	0.53	0.35	0.46	0.69	0.68	0.65
	strict + score	0.74	0.4	0.78	0.58	0.68	0.45	0.56	0.82	0.75	0.75
KG * retrieval + finalise	strict	0.66	0.4	0.74	0.56	0.63	0.42	0.57	0.74	0.73	0.71
	score	0.51	0.27	0.66	0.46	0.53	0.34	0.47	0.68	0.55	0.57
	strict + score	0.55	0.3	0.68	0.51	0.54	0.36	0.46	0.7	0.64	0.65
random forest	input = categories				N/A				0.46	0.53	0.49
	input = values				N/A				0.28	0.38	0.32

Table 3: Evaluation results for different methods. PE: prompt engineering. RAG: A = using all the rows; B = using abnormal rows; C = adding normalised value as a column. The KG achieves the best result with strict match + confidence score, using the KG built with examples, with a F1 score 0.76. The KG * has a similar performance for the strict match, but worse with the confidence score and combination of strict match and confidence score. The best result for RAG is an F1 score of 0.56, which is higher than the zero-shot and random forest results.

References

- Esraa Abdelmaksoud, Ahmed Gadallah, and Ahmed Asad. 2022. [Clinical laboratory test reports](#).
- Sedat Abusoglu, Muhittin Serdar, Ali Unlu, and Gulsum Abusoglu. 2024. Comparison of three chatbots as an assistant for problem-solving in clinical laboratory. *Clinical Chemistry and Laboratory Medicine (CCLM)*, 62(7):1362–1366.
- Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.
- Janne Cadamuro, Federico Cabitza, Zeljko Debeljak, Sander De Bruyne, Glynis Frans, Salomon Martin Perez, Habib Ozdemir, Alexander Tolios, Anna Carobene, and Andrea Padoan. 2023. Potentials and pitfalls of chatgpt and natural-language artificial intelligence models for the understanding of laboratory medicine test results. an assessment by the european federation of clinical chemistry and laboratory medicine (eflm) working group on artificial intelligence (wg-ai). *Clinical Chemistry and Laboratory Medicine (CCLM)*, 61(7):1158–1166.
- Stephen Gilbert, Jakob Nikolas Kather, and Aidan Hogan. 2024. Augmented non-hallucinating large language models as medical information curators. *NPJ Digital Medicine*, 7(1):100.
- Zhe He, Shubo Tian, Arslan Erdengasileng, Karim Hanna, Yang Gong, Zhan Zhang, Xiao Luo, and Mia Liza A Lustria. 2023. Annotation and information extraction of consumer-friendly health articles for enhancing laboratory test reporting. In *AMIA Annual Symposium Proceedings*, volume 2023, page 407. American Medical Informatics Association.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Andrea Padoan and Mario Plebani. 2022a. Artificial intelligence: is it the right time for clinical laboratories? *Clin Chem Lab Med*, 60(12):1859–1861.
- Andrea Padoan and Mario Plebani. 2022b. Flowing through laboratory clinical data: the role of artificial intelligence and big data. *Clinical Chemistry and Laboratory Medicine (CCLM)*, 60(12):1875–1880.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Meru A Patil, Sandip Bhaumik, Soubhik Paul, Swaruppananda Bissoyi, Raj Roy, and Seungwoo Ryu. 2013. Estimating personalized risk ranking using laboratory test and medical knowledge (umls). In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1274–1277. IEEE.
- Mario Plebani. 2009. Interpretative commenting: a tool for improving the laboratory–clinical interface. *Clinica Chimica Acta*, 404(1):46–51.
- Emma Stevenson, Chelsey Walsh, and Luke Hibberd. 2024. Can artificial intelligence replace biochemists? a study comparing interpretation of thyroid function test results by chatgpt and google bard to practising biochemists. *Annals of Clinical Biochemistry*, 61(2):143–149.
- Tao Tu, Anil Palepu, Mike Schaekermann, Khaled Saab, Jan Freyberg, Ryutaro Tanno, Amy Wang, Brenna Li, Mohamed Amin, Nenad Tomasev, et al. 2024. Towards conversational diagnostic ai. *arXiv preprint arXiv:2401.05654*.
- Dave Van Veen, Cara Van Uden, Louis Blanke-meier, Jean-Benoit Delbrouck, Asad Aali, Christian Bluethgen, Anuj Pareek, Malgorzata Polacin, Eduardo Pontes Reis, Anna Seehofnerová, et al. 2024. Adapted large language models can outperform medical experts in clinical text summarization. *Nature medicine*, 30(4):1134–1142.
- Youfu Yan, Yu Hou, Yongkang Xiao, Rui Zhang, and Qianwen Wang. 2024. Knownet: Guided health information seeking from llms via knowledge graph integration. *IEEE Transactions on Visualization and Computer Graphics*.
- Wen-wai Yim, Yujuan Fu, Asma Ben Abacha, Neal Snider, Thomas Lin, and Meliha Yetisgen. 2023. Acibench: a novel ambient clinical intelligence dataset for benchmarking automatic visit note generation. *Scientific Data*, 10(1):586.
- Cyril Zakka, Rohan Shad, Akash Chaurasia, Alex R Dalal, Jennifer L Kim, Michael Moor, Robyn Fong, Curran Phillips, Kevin Alexander, Euan Ashley, et al. 2024. Almanac—retrieval-augmented language models for clinical medicine. *NEJM AI*, 1(2):AIoa2300068.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

A Appendix

A.1 The KG Schema

Start Node Type	Relationship Type	End Node Type
TestResult	CONTRIBUTES_TO	Condition
Condition	COMPOSED_OF	StandardTerm
Report	HAS_PATIENT	Patient
Patient	HAS_TEST	Test
Patient	HAS_COMMENT	Comment
Patient	HAS_CONDITION	Condition
Patient	HAS_AGE	Age
Patient	HAS_GENDER	Gender
Patient	HAS_RESULT	TestResult
Condition	IS_A	StandardTerm
Test	HAS_REF	Reference
Test	HAS_UNIT	Unit
Test	HAS_TEST_RESULT	TestResult

Table 4: The schema of the lab-kg

A.2 An Query Example

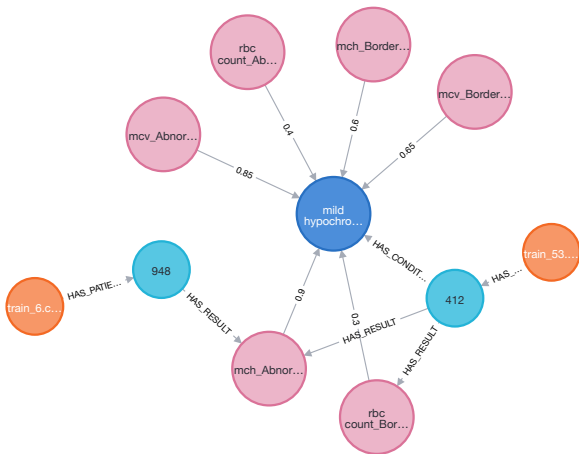


Figure 5: The “mild hypochromia” query process for a new patient. The query patient with ID 948 lacks “RBC count Borderline Abnormal (Low)” compared to the example patient with ID 412 using strict test result matching. The RBC count is not necessary for “mild hypochromia”.

A.3 Confidence Score Calculation

In this example, the weights associated with the condition “mild normochromic normocytic anaemia” are shown in Table 2.

The maximum possible score is the sum of the highest weights for each test result:

$$\begin{aligned} \text{Max Score} &= 0.95 + 0.95 + 0.9 + 0.85 \\ &\quad + 0.7 + 0.5 + 0.4 + 0.3 \\ &= 5.55 \end{aligned}$$

We sum the weights of the matching test results for the patient’s test results, omitting borderline weights if there is a corresponding abnormal weight with a higher value. In this case, the confidence score is:

$$\begin{aligned} \text{Patient Score} &= 0.75 + 0.75 + 0.85 \\ &\quad + 0.5 + 0.4 + 0.3 \\ &= 3.55 \end{aligned}$$

The normalised confidence score is:

$$\text{Confidence Score} = \frac{3.55}{5.55} = 0.64$$

A.4 KG and KG * Comparison

To compare the differences between the results from *CONTRIBUTES_TO* relationships built by the LLM with and without examples, we compared the generated results and highlighted the most significant differences below:

1. KG with relevance built with example wins

KG * cannot generate “Mild absolute neutropenia” in many cases, which is in the target output. The reason is that the LLM only assigns weight to *neutrophils percentage_Abnormal (Low)*, *absolute neutrophils count_Borderline (Low)*, and *neutrophils percentage_Borderline (Low)*. However, the absolute neutrophil count comprises *absolute segmented neutrophil count* and *absolute band neutrophil count*, which are present in the examples but not recognised by the LLM’s internal knowledge without examples.

2. KG with Relevance Built without Example Wins

The KG built with examples cannot use strict match for the condition “mild hypochromia.” Because some example patients (12.5% across all the patients with that condition) have a test result *RBC count_Abnormal (High)* associated with that condition, the LLM connects that test result to the condition with a low weight (0.4). However, in the

strict match, the new patient only has the test result *MCV_Abnormal (Low)*, which is not present in the example patient who has both *MCV_Abnormal (Low)* and *RBC count_Abnormal (High)*. The internal LLM did not connect *RBC count_Abnormal* to that condition, and it can retrieve that condition by strict match. The LLM’s decision is affected by the noise of the dataset, which causes this misidentification. The graph for this query is Figure 5 in appendix.

A.5 Correlation between F1 Score and Other Metrics

metrics	medcon	bleu	bert_score	meteor	rouge1	rouge2	rougeL
corr	0.95	0.75	0.94	0.91	0.91	0.9	0.85
corr *	0.97	0.66	0.91	0.87	0.84	0.9	0.74

Table 5: The correlation between F1 score and each metric, for the result built by 1. KG (corr) and 2. KG * (corr *).

Bridging Language and Scenes through Explicit 3-D Model Construction

Tiansi Dong^{1,2} and Writwick Das¹ and Rafet Sifa¹

¹Hybrid Intelligence, Fraunhofer IAIS / Schloss Birlinghoven, 1, 53757 Sankt Augustin, Germany

²Computer Laboratory, University of Cambridge / 15 JJ Thomson Ave, Cambridge, UK

{tiansi.dong, writwick.das, rafet.sifa}@iais.fraunhofer.de

td540@cam.ac.uk

Abstract

We introduce the methodology of explicit model construction to bridge linguistic descriptions and scene perception and demonstrate that in Visual Question-Answering (VQA) using MC4VQA (Model Construction for Visual Question-Answering), a method developed by us. Given a question about a scene, our MC4VQA first recognizes objects utilizing pre-trained deep learning systems. Then, it constructs an explicit 3-D layout by repeatedly reducing the difference between the input scene image and the image rendered from the current 3-D spatial environment. This novel “iterative rendering” process endows MC4VQA the capability of acquiring spatial attributes without training data. MC4VQA outperforms NS-VQA (the SOTA system) by reaching 99.94% accuracy on the benchmark CLEVR datasets, and is more robust than NS-VQA on new testing datasets. With newly created testing data, NS-VQA’s performance dropped to 97.60%, while MC4VQA still kept the 99.0% accuracy. This work sets a new SOTA performance of VQA on the benchmark CLEVR datasets, and shapes a new method that may solve the out-of-distribution problem. The source code and data sets are available for public access <https://github.com/writzx/mc4vqa/>.

1 Introduction

The success of LLMs is witnessed by its capability of human-like question-answering (Biever, 2023), but, they remain as black-box systems, data hungry, and do not work well for out-of-distribution data in real application (Goyal and Bengio, 2022). Spatial semantics bridges spatial descriptions and visual perception and is the first semantics that human babies acquire. It is used as a reference for the understanding of other semantics (Regier, 1997; Bellmund et al., 2018). It plays a fundamental role in computational linguistics and cognitive modelling (Tversky,

2019). Visual question answering (VQA) is a challenging task that involves answering questions about an image in natural language (Agrawal et al., 2016; Wu et al., 2016). For example, given an image of a dice and the question "What is the shape of the object?", a VQA system should be able to generate the answer “cube”. VQA is a challenging task because it requires the model to understand both the visual and spatial content of the image and the meaning of the question (Agrawal et al., 2016; Zou and Xie, 2020). A VQA system must be able to reason about spatial relations, such as the distance between objects, the relative positions of objects, and the orientation of objects. The state-of-the-art (SOTA) VQA system is Neural-Symbolic VQA (NS-VQA) (Yi et al., 2019). NS-VQA achieves a near-perfect accuracy of 99.8% on the CLEVR dataset (Johnson et al., 2016), which is a challenging dataset of images and questions that test a VQA system’s ability to reason about spatial relations.

NS-VQA combines deep representation learning for visual recognition and language understanding with symbolic program execution for reasoning. NS-VQA generates executable programs as the meaning of the question, and apply for the learned visual and spatial attributes to produce the answer. NS-VQA learns spatial attributes about an input image by supervised deep learning. Therefore, it does not have an explicit 3-D spatial layout of the input image. This weakens the explainability and reliability, makes the system data-hungry and performs well only when training and testing data share the same or very similar distribution (Goyal and Bengio, 2022; Gigerenzer, 2022).

On the other hand, sufficient empirical experiments in psychological research advocates the model theory for spatial reasoning (Johnson-Laird and Byrne, 1991; Knauff et al., 2003; Goodwin and Johnson-Laird, 2005; Knauff, 2009, 2013), whose standard process is a sequence of *model construction*, *model inspection*,

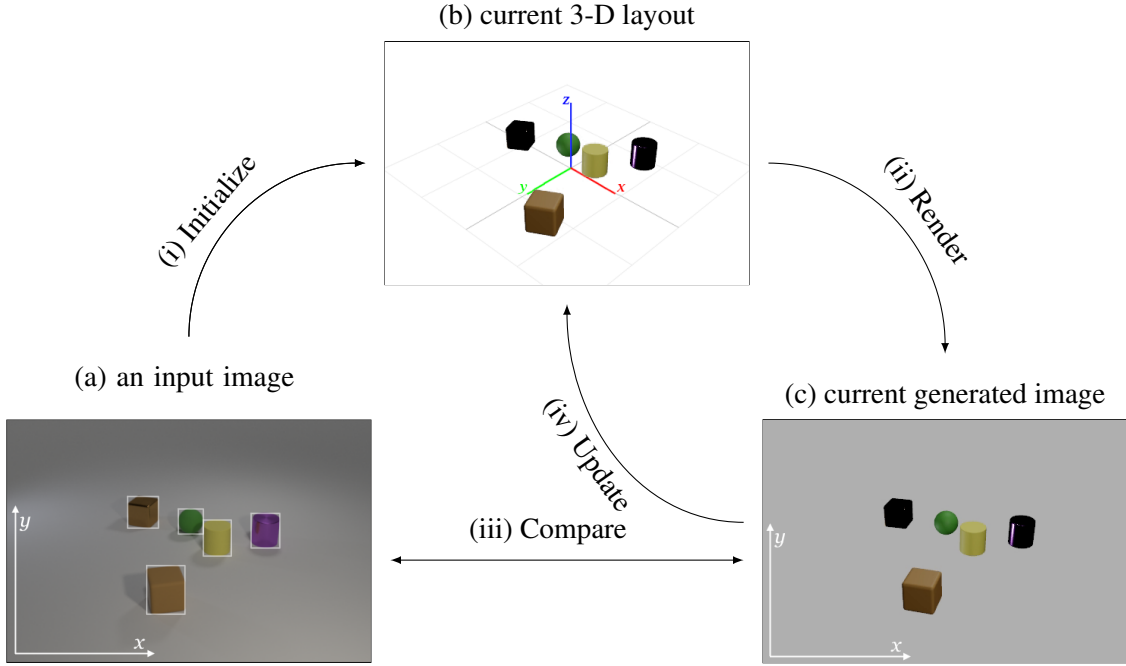


Figure 1: Overview of the MCIR process: (a) An input 2-D image; (i) Initializing a 3-D model of a scene with the colors, shapes and materials of the objects detected in the 2-D input image; (b) Reconstruction of a 3-D spatial layout of the input image; (ii) Perform perspective projection on the 3-D model to generate a 2-D image and realistic 2-D coordinates of the objects; (c) A projected 2-D image generated using the current 3-D spatial layout; (iii) Compare the projected coordinates of the objects with the bounding boxes to calculate their distances from their original 2D locations; (iv) Update the positions of the objects in 3-D layout to reduce the difference calculated in (iii).

tion, and model variation (Johnson-Laird and Byrne, 1991). The preferred mental model theory argues that people construct a preferred and simplified model in mind, in a deterministic manner, while ignoring other possible models (Ragni and Knauff, 2013; Knauff, 2013) – The construction of the first model shall not be a stochastic process that *produces one model this time and another the next time* (Ragni and Knauff, 2013, p.563-564), the next model will be revised following the principle of minimal changes from the current one (Harman, 1986; Gärdenfors, 1988; Gärdenfors, 1990; Knauff et al., 2013), and generated by a local transformation of the current model.

Inspired from the model theory, here, we move one step ahead of NS-VQA, by replacing its supervised learning component of spatial attribute with a 3-D spatial reconstruction component, and developed the process of “Model Construction by Iteration Render” (MCIR). As illustrated in Figure 1, the MCIR process first initialises a 3-D spatial layout for all recognised objects, Figure 1(i), followed by the loop of Render-and-Update, Figure 1(ii,iv). The Render operation projects a 3-D layout into a 2-D image, Figure 1(c); the Update operation is carried out to reduce the difference between the

original input image and the current rendered image. The result of the Comparison operation is always greater than or equal to zero.

We compare MC4VQA with NS-VQA in two experiments. The first experiment is performed using the original CLEVR dataset. MC4VQA achieved an accuracy of 99.94%. This outperforms all state-of-the-art methods, including NS-VQA. The aim of the second experiment is to examine whether traditional supervised learning endows neural-networks the ability to acquire 3-D spatial attributes from 2D images. We developed a new testing dataset, which contains 4000 images, generated by the CLEVR image generator from four different camera perspectives. Each scene is generated using a randomly selected camera configuration. NS-VQA had an overall accuracy of 98.39%. In contrast, our proposed method maintained another near-perfect accuracy at 99.8%. The success of MC4VQA not only demonstrates the power of the method of model construction and inspection for the acquisition of spatial knowledge (advocated in the psychological literature), but also shows the limitation of supervised deep learning – lacking the ability of generalisation of training patterns (Goyal and Bengio, 2022).

The contributions of MC4VQA are listed as follows: (1) it is the first VQA system that explicitly reconstructs 3-D spatial layout to bridge spatial linguistic descriptions and visual perception; (2) MC4VQA can be further developed by integrating more features of mental model theory in psychology, or used in psychological experiments; (3) Source code and new datasets are publicly accessible. The rest of the paper is structured as follows: Section 2 reviews a number of related works; Section 3 formalises the task of VQA by explicitly re-constructing 3-D spatial layout; Section 4 presents the detail of MC4VQA; Section 5 reports experiment results of MC4VQA, which greatly outperforms the SOTA performance, and demonstrates the power of the model construction method in new testing data; Section 5 concludes the paper, and lists a number of future research topics.

2 Related Work

A convergent opinion from linguistics, neuroscience, and psychology is that the spatial domain is the first domain that human babies understand, and is the reference domain for the understanding of other domains (Lakoff and Johnson, 1980; Regier, 1997; Grady, 1997; Tversky, 2019). The next generation of language system shall be a brain- and AI-inspired understanding system that explicitly represents situations (McClelland et al., 2020). Our work focuses on the NS-VQA model, and promises a novel method to explicitly represent scene images by constructing 3-D geometric spatial models. NS-VQA uses an older object detection model based on Detectron (Girshick et al., 2018) and Mask R-CNN (He et al., 2018). Since then, newer models with improved accuracy and speed have been released, such as YOLO (Redmon et al., 2016; Jocher et al., 2023), which produces impressive results and can be used for real-time video processing.

YOLO YOLO (You Only Look Once) is a powerful object detection model which is known for its speed and accuracy (Redmon et al., 2016). The current version of YOLO (v8) (Jocher et al., 2023) is the state-of-the-art object detection model that utilizes Cross Stage Partial (CSP) (Wang et al., 2019) architecture, which was introduced in YOLOv4 (Bochkovskiy et al., 2020). Our MC4VQA uses YOLOv8 as its object detection model. YOLO offers several pretrained models, of which we chose "YOLOv8x-seg" which has great segmentation accuracy.

Question Parsing and Execution Several papers have used program search and neural networks to recover programs from domain specific language (Neelakantan et al., 2016; Balog et al., 2017), including semantic parsing methods (Berant et al., 2013; Liang et al., 2011) to map sentences to logical forms from a knowledge base. Prior knowledge of semantics of the program and execution context is important to correctly parse an arbitrary set of question tokens following the semantics. So, the model needs to learn based on a set of input questions and answer pairs. NS-VQA's question parser follows the work done by (Andreas et al., 2016; Rothe et al., 2017; Goldman et al., 2019). The parser implementation uses a Bi-LSTM parser to generate programs from sentences similar to CLEVR-IEP (Johnson et al., 2017). The execution engine is slightly different from IEP, in the sense that it uses symbolic reasoning based on object positions generated by its attribute network.

Neural-symbolic approach to VQA NS-VQA stands for "Neural-symbolic Visual Question Answering" (Yi et al., 2019). Traditional neural-network approaches often do not have competitive performance on challenging reasoning tasks on CLEVR dataset (Johnson et al., 2016). In contrast, NS-VQA achieves a near-perfect accuracy on the CLEVR dataset, by learning a symbolic program from the question, and executing the program on an implicit spatial model learned by supervised deep learning, ResNet34 (He et al., 2015). It remains unclear whether NS-VQA's ResNet34 really learns the way to acquire 3D spatial relations from 2D images. The symbolic program may only match similar pairwise relationships in the training scene images. Furthermore, supervised models for generating 3D scene representations are prone to bias due to the invariant camera configuration used by the CLEVR training images.

3 Motivation of VQA through Model Construction and Inspection

Ever since Tolman's rats experiments (Tolman, 1948) in the 1940s, sufficient evidence has been collected to show that animals and humans can construct comprehensive spatial models in mind of their environments through sensorimotor interaction (Spelke and Lee, 2012) and that this spatial model in mind structures our language (Lakoff and Johnson, 1980; Tversky and Lee, 1999; Tversky, 2019). This motivates us to move one step ahead of NS-VQA by replacing its supervised ResNet34 compo-

ment with a novel component that explicitly constructs 3D spatial layout, thus MC4VQA (Model Construction for VQA). This allows the symbolic program execution engine to more accurately identify objects and their spatial relationships in the scene. As being unsupervised, our method may improve the overall generalization of the scene construction, allowing to function on unknown camera configurations.

4 Formalising the task

In this section, we define the task of VQA through model construction and inspection. The input of MC4VQA consists of an image \mathcal{I} and a question \mathcal{Q} asking the content of this image, whose content can be described as a set of objects $\mathcal{I}_{O_1} \dots \mathcal{I}_{O_n}$ and a set of 2D locations \mathcal{L}_{O_i} of \mathcal{I}_{O_i} , line 1 in Algorithm 1. The process of model construction \mathbf{P} will construct a 3D spatial layout \mathcal{S} for \mathcal{I} . \mathcal{S} consists of a set of 3D objects O_i with their size and their 3D location information.

Let \mathcal{S}_0 be an initial 3D layout, line 2 in Algorithm 1, the construction process \mathbf{P} will update \mathcal{S}_i to \mathcal{S}_{i+1} , with the following procedure: \mathbf{P} will trigger an inspection function \mathbf{I} to take a photo of \mathcal{S}_i , so called “rendering”, let $\mathbf{I}(\mathcal{S}_i) = \mathcal{I}^{(i)}$. Then, a function \mathbf{M} will measure the difference between $\mathcal{I}^{(i)}$ and the original image \mathcal{I} . Finally, a function \mathbf{g} will apply a set of geometric operations on objects in \mathcal{S}_i . This transforms \mathcal{S}_i into \mathcal{S}_{i+1} , so that a photo of \mathcal{S}_{i+1} will be more similar to the original image, that is, $\mathbf{M}(\mathcal{I}^{(i+1)}, \mathcal{I}) < \mathbf{M}(\mathcal{I}^{(i)}, \mathcal{I})$. The construction process will stop, if $\mathbf{M}(\mathcal{I}^{(i+1)}, \mathcal{I})$ is less than a predefined threshold value ϵ . The final 3D layout \mathcal{S}_n will be inspected to answer the question \mathcal{Q} (Algorithm 1).

5 MC4VQA

MC4VQA has four components: an object detector (YOLOv8), a 3D model constructor (MCIR), a question parser (Bi-LSTM encoder), and a program executor.

Object Detection The YOLOv8 object detector is trained on the same 4000 CLEVR images used by NS-VQA. The input image is first passed to the object detector to generate object proposals. The object proposals are composed of the predicted object masks and the object bounding boxes, along with their class names. Object proposals with a score of less than 0.9 are discarded. The predicted class names are composed of the discrete attributes of the objects, e.g., the object size, colour, material, and

Algorithm 1: VQA by 3D model construction and inspection

Input: an image \mathcal{I} ;
Input: a question \mathcal{Q} about the content of \mathcal{I} ;
Output: an answer \mathcal{A} to \mathcal{Q} ;

- 1 recognise 3D objects $O_1 \dots O_n$ in \mathcal{I} ;
- 2 Initialise 3D spatial layout \mathcal{S}_c by placing all O_i at the same location;
- 3 $\mathcal{I}^{(c)} \leftarrow \mathbf{I}(\mathcal{S}_c)$;
- 4 **while** $\mathcal{I}^{(c)}$ not similar with \mathcal{I} **do**
- 5 update 3D locations and postures of objects O_i in \mathcal{S}_c , to increase the similarity to \mathcal{I} ; ▷ reduce the value $\mathbf{M}(\mathcal{I}_c) - \mathbf{M}(\mathcal{I})$
- 6 $\mathcal{I}^{(c)} \leftarrow \mathbf{I}(\mathcal{S}_c)$; ▷ $\mathcal{I}^{(c)}$ is a photo of \mathcal{S}_c
- 7 $\mathcal{A} \leftarrow$ answer \mathcal{Q} by inspecting 3D layout \mathcal{S}_c ;
- 8 **return** \mathcal{A}

shape. These attributes are used to construct the 3D scene and to answer the questions.

3D Model Construction The object proposals generated by the object detector are passed to MCIR, which processes the bounding boxes of the objects to compute more realistic box midpoints. The bounding boxes from the object detector do not take into account occlusion behind other objects, so it is important to correct them before generating the 3D scene.

After the approximately realistic midpoints are generated, they are passed to MCIR, which generates the 3D spatial model. This model is then passed to the question executor as the scene representation of the input image.

Question Parsing and Program Execution The question parser and the program executor used by MC4VQA are both directly taken from the NS-VQA implementation without any changes. The output format of MCIR is compatible with the input format of the program executor, so they integrate well with each other. The reconstructed 3-D representation is used to generate the answers.

6 Experiments

A series of experiments are conducted to compare the methods of model construction and of supervised learning for VQA.

Experiment I MC4VQA is implemented by replacing NS-VQA’s supervised learning model with a model of 3D scene construction

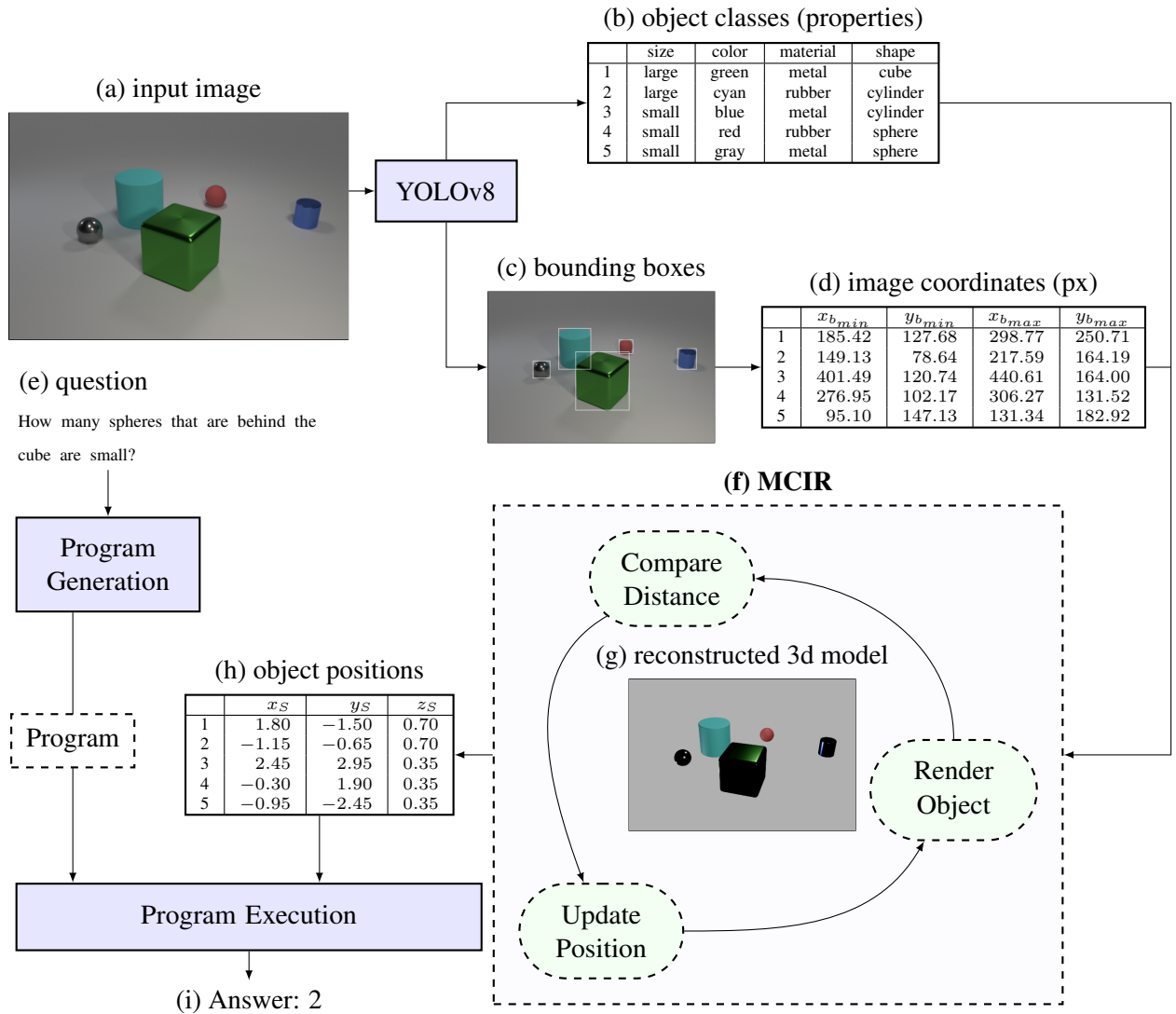


Figure 2: Overview of NS-VQA Extended with Iterative Rendering

to acquire spatial attributes, and share the same object detection model and the same model of question parsing and program execution.

We used three camera configurations to test the performance of MC4VQA as follows: (1) C1 was a random configuration to serve as a baseline; (2) C2 was chosen to simulate the camera direction that a human would likely choose when looking at the CLEVR images; (3) C3 was calculated based on the average of the first ten camera directions specified in the CLEVR scenes to represent a manually fine-tuned camera configuration.

YOLO for object proposals We trained a YOLOv8 object detector on the same 4000 CLEVR images. These are the same images used to train the object proposal model of NS-VQA in (Yi et al., 2019). Object proposals with a score of less than 0.9 were discarded. A predicted class name consists of 2 discrete attributes

of the object, such as the size, the colour, the material, and the shape. These attributes are used to construct the 3D scene and to answer the questions using the program executor. The training of the YOLOv8 model was run on resized image size of 480x480 for 100 epochs with a learning rate of 0.01.

Equipped with this YOLO model, NS-VQA (Yi et al., 2019) improves its overall accuracy from 99.8% to 99.93%, as listed in Table 1.

VQA through 3-D Model Construction MC4VQA uses YOLO object proposals to initialise a 3-D layout, then repeatedly optimizes this layout by reducing the difference between the objects in the input image and the objects in the 3-D scene generated by the rendering engine. Then, MC4VQA uses NS-VQA’s question parser to generate programs and apply them to the 3D layout to generate answers, whose correctness is validated by the ground

Methods	Count	Exist	Compare Number	Compare Attribute	Query Attribute	Overall
Humans	86.7	96.6	86.5	95.0	96.0	92.6
MDETR (Kamath et al., 2021)	99.3	99.9	99.4	99.9	99.9	99.7
NMN (Andreas et al., 2017)	52.5	72.7	79.3	79.0	78.0	72.1
N2NMN (Hu et al., 2017)	68.5	85.7	84.9	90.0	88.7	83.7
IEP (Johnson et al., 2017)	92.7	97.1	98.7	98.1	98.9	96.9
TbD (Mascharka et al., 2018)	97.6	99.4	99.2	99.5	99.6	99.1
RN (Santoro et al., 2017)	90.1	93.6	97.8	97.1	97.9	95.5
FiLM (Perez et al., 2017)	94.5	93.8	99.2	99.2	99.0	97.6
NS-CL (Mao et al., 2019)	98.2	99.0	98.8	99.3	99.1	98.9
MAC (Hudson and Manning, 2018)	97.2	99.4	99.5	99.3	99.5	98.9
OCCAM (Wang et al., 2021)	98.1	99.8	99.0	99.9	99.9	99.4
NS-VQA (Yi et al., 2019)	99.7	99.9	99.9	99.8	99.8	99.8
NS-VQA (YOLOv8)	99.87	99.96	99.93	99.93	99.95	99.93
MC4VQA [C1]	99.89	99.97	99.94	99.91	99.92	99.92
MC4VQA [C2]	99.92	99.98	99.93	99.94	99.95	99.94
MC4VQA [C3]	99.92	99.97	99.93	99.97	99.94	99.94

Table 1: NS-VQA outforms state-of-the-art methods on the CLEVR dataset. With introduction of the YOLO model the accuracy is improved. Integrating with iterative render further improves the accuracy to a near perfect 99.94%. Our model depends on the camera configuration of the system. C¹ is a random configuration to serve as a baseline. C² is chosen to simulate the camera direction that a human would likely choose when looking at the CLEVR images. C³ is calculated based on the average of the first ten camera directions specified in the CLEVR scenes to represent a manually fine-tuned camera configuration.

Methods	Count	Exist	Compare Number	Compare Attribute	Query Attribute	Overall
NS-VQA	97.86	99.03	99.22	98.53	98.21	98.39
MC4VQA	99.52	99.85	99.97	99.90	99.88	99.80

Table 2: NS-VQA (YOLOv8) with attribute net performs slightly worse at 98.39% than MC4VQA (YOLOv8) with MCIR, which still maintains near perfect accuracy at 99.80%

truth in the validation set. The performance is measured in terms of the accuracy.

Results and Analysis Experiment results show that MC4VQA reaches 99.94% overall accuracy on the benchmark CLEVR dataset without training data. This outperforms the SOTA NS-VQA (Yi et al., 2019) and the NS-VQAv8 (NS-VQA with YOLO model). Experiments also show that MC4VQA reaches the performance of NS-VQAv8 in each evaluation task, at least from one camera configuration. We conclude that MC4VQA successfully acquired spatial attributes by utilising the method of 3D model construction without training data.

Experiment results show that MC4VQA reaches 99.94% accuracy on the benchmark CLEVR dataset, without training data. This outperforms the SOTA NS-VQA (Yi et al., 2019) and the NS-VQAv8 (NS-VQA with YOLO model). Experiments also show that MC4VQA reaches the performance of NS-VQA at least from one camera configuration for rendering. We conclude that *by utilising the method of 3D model construction, MC4VQA successfully acquired spatial attributes without*

training data.

Experiment II In Experiment I, the testing and training data are from benchmark CLEVR dataset, sharing the same distribution. The second experiment compares the performances of the well-trained NS-VQA and MC4VQA on new test datasets.

Design of the experiment We generated 4000 CLEVER images with four different camera configuration, and 40000 questions, and fed them to the well-trained NS-VQA with YOLOv8 and MC4VQA.

Experiment Results show that the overall performance of NS-VQA drops from 99.93% to 98.39% and that the overall performance of MC4VQA slightly drops from 99.94% to 99.80%, Table 2. This suggests our method is more robust than NS-VQA.

Error Analysis We examined cases when NS-VQA made mistakes. In Figure 3, NS-VQA fails to locate the small gray cube accurately, resulting in an incorrect answer. MC4VQA overcomes this limitation by using corrected bounding boxes and a 3D spatial model to cor-

Algorithm 2: The simple MCIR Algorithm

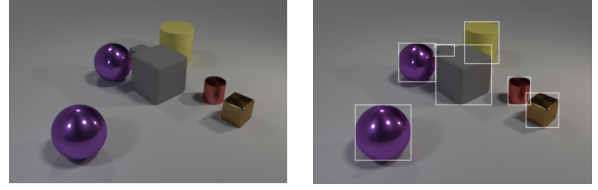
Input: object proposals from YOLO**Data:** o_{max} - total number of objects**Data:** j_{max} - maximum number of iterations

```
1  $o_i \leftarrow 1$ ; /*  $o_i$ : current object index */
2 while  $o_i \leq o_{max}$  do
3    $j \leftarrow 1$ ;
4    $O \leftarrow \text{objects}[o_i]$ ;
5    $C \leftarrow \text{box-midpoints}[o_i]$ ;
6    $S \leftarrow \text{initialize}(O)$ ;
7    $I \leftarrow \text{project}(S)$ ; /*  $I \sim (x_I, y_I)$ : 2-D
   image coordinates of  $O$  (current) */
8    $d \leftarrow |C - I|$ ; /*  $d$ : pixel distance */
9    $u_p \leftarrow 1$ ; /*  $u_p$ : previously used update
   value */
10  while  $d > d_{threshold}$  do
11     $u_i \leftarrow u_p$ ; /*  $u_i$ : index of update
   value */
12    while  $j \leq j_{max}$  do
13      /*  $U$ : set of available update
   values */
14      /*  $u_{max}$ : number of update
   values */
15       $u \leftarrow U[u_i \bmod u_{max}]$ ;
16       $S_c \leftarrow S + u$ ; /*  $S_c$ : candidate
   scene coordinate */
17       $I_c \leftarrow \text{project}(S_c)$ ; /*  $I_c$ :
   candidate image coordinate */
18       $d_c \leftarrow |C - I_c|$ ; /*  $d_c$ : new pixel
   distance */
19      if  $d_c < d$  then
20         $S \leftarrow S_c$ ;  $I \leftarrow I_c$ ;  $d \leftarrow d_c$ ;
21         $u_p \leftarrow u_i$ ;
22        break
23     $u_i \leftarrow u_i + 1$ 
24   $o_i \leftarrow o_i + 1$ 
```

rectly identify the cube’s location. NS-VQA made similar mistakes when there are objects very close to together each other. We hypothesize that the performance of NS-VQA drops if the questions are about closely situated objects. We report Experiment III as follows.

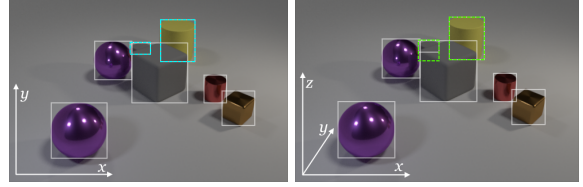
Experiment III We create a new testing dataset, in which some objects are very close to each other, and evaluate the performances of NS-VQA and MC4VQA.

Design of the experiment Two sets of CLEVR images were created, 1000 images for each, as follows.



(a) An input image, where two gray cubes are very closely located.

(b) Bounding boxes created by YOLO object detection model.



(c) 2D spatial attribute used by NS-VQA

(d) 3D spatial layout used by MC4VQA

Figure 3: (a) Given an input image and the question “what number of objects are behind the small brown metallic thing and in front of the yellow metta object?” (b) YOLO successfully identifies all objects with bounding boxes. In (c) NS-VQA uses 2D YOLO bounding boxes. In this case, the small gray cube is not calculated as being in front of the yellow cylinder. (d) MC4VQA used its constructed 3-D spatial layout, instead of 2D YOLO bounding boxes, and correctly calculated the small gray cube being in front of the yellow cylinder.

- In one set, there are two objects being very close to each other; (minimum distance between two objects is 0.1 units, as opposed to CLEVR default of 0.4 units)
- In another set, at least two objects are close, and all objects are less spread out in the scene. (maximum coordinates along the axes: 2.0 units, as opposed to CLEVR default of 3.0 units)

These two testing datasets were fed to NS-VQA and MC4VQA.

Results an analysis The performance of NS-VQA continued to decrease to below 98.0%. The performance of MC4VQA decreased slightly, and still reached 99.0% in both testing datasets, as listed in Tables 3 and 4, respectively.

Limitations of MC4VQA Our MCIR process optimises a 3D layout through reducing the difference between a rendered image and the input image. It does not have other spatial constraints, such as extended 3D objects cannot be partially overlapped. This limitation will cause MC4VQA to construct incorrect 3D layout. For example, Figure 4 illustrates a new testing image whose camera configuration is very near to the objects. This causes the effect of plac-

Methods	Count	Exist	Compare Number	Compare Attribute	Query Attribute	Overall
NS-VQA	96.54	98.48	98.97	99.47	97.44	97.90
MC4VQA	98.70	100.00	97.94	100.00	99.43	99.30

Table 3: NS-VQA vs MC4VQA when the objects are closer to each other.

Methods	Count	Exist	Compare Number	Compare Attribute	Query Attribute	Overall
NS-VQA	95.67	99.24	96.91	98.94	97.73	97.60
MC4VQA	98.70	100.00	98.97	99.47	98.58	99.00

Table 4: NS-VQA vs MC4VQA when the objects are close and less spread out.

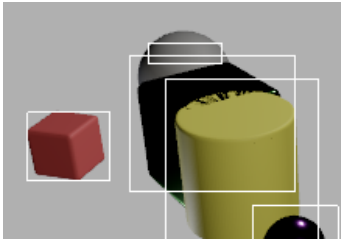


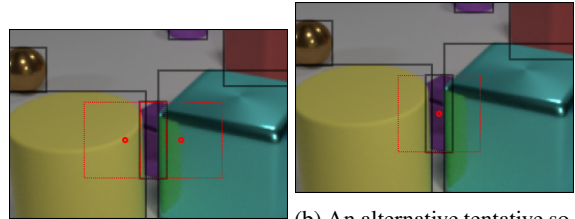
Figure 4: When objects are very close to each other in a 3D layout, they may be partially overlapped, as we see there is a yellowish black at the edge of the top surface of the yellow cylinder behind it.

ing large 3D objects in a relative small place. Without explicit spatial constraints, nearby 3D objects can be partially overlapped.

Another limitation of the MCIR system is using single camera configuration. Under certain situations, it might not be possible to figure out the precise location of an object in the 3D layout. For example, Figure 5(a) illustrates an image, in which a purple object is behind a big yellow cylinder and a green cuboid, only a very small part can be seen. Although this small part is sufficient to recognise what object class and what size it is, figuring out its precise location will be hard. Tentative solutions can be to set the bounding box as left (or right) as possible, Figure 5(a), or let the centre of the bounding box and the seen part be coincided, Figure 5(b). Each tentative solution can cause MC4VQA to give incorrect answers.

7 Conclusions and outlooks

Understanding surrounding environment is a fundamental ability for the survival of animals and humans, e.g., to escape from dangerous predators. It is a challenging research task in NLU and AI, and has various downstream applications, e.g., autonomous driving, service



(a) Left-most or right-most bounding-boxes can be used as tentative solutions. (b) An alternative tentative solution is to put the object to the centre of the bounding box.

Figure 5: A purple object is occluded by two big objects, whose location is hard to figure.

robots. VQA with the benchmark CLEVR dataset is a micro-world to explore this field, in which images are about layouts of synthesised geometric objects. Supervised neural networks to learn spatial attributes are very successful, with two conditions: (1) it needs a huge amount of training data; (2) the testing data shall have the same distribution as the training data. Both conditions are either expensive or unrealistic for real applications. We replace the method of supervised learning with the method of model construction to free the acquisition of spatial attributes from the imprisonment of data and go beyond the paradigm of supervised learning.

Our experiment results show that our new method is very promising – it does not need training data for acquiring spatial regions and achieves higher accuracy in answering questions about out-of-distribution scenes.

In this work, we implemented MCIR using a simple object-level loop to optimize object locations and used NS-VQA’s question parser and executor with the CLEVR validation questions. In the future, we will adopt a dual-camera configuration to figure out the locations of 3D objects precisely and will use the constructed 3D layout construction as the spatial semantics to interpret linguistic descriptions.

References

- Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. 2016. *Vqa: Visual question answering*. *Preprint*, arXiv:1505.00468.
- Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. *Learning to compose neural networks for question answering*. *Preprint*, arXiv:1601.01705.
- Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2017. *Neural module networks*. *Preprint*, arXiv:1511.02799.
- Matej Balog, Alexander L. Gaunt, Marc Brockschmidt, Sebastian Nowozin, and Daniel Tarlow. 2017. *Deepcoder: Learning to write programs*. *Preprint*, arXiv:1611.01989.
- Jacob L. S. Bellmund, Peter Gärdenfors, Edward I. Moser, and Christian F. Doeller. 2018. *Navigating cognition: Spatial codes for human thinking*. *Science*, 362(6415).
- Jonathan Berant, Andrew K. Chou, Roy Frostig, and Percy Liang. 2013. *Semantic parsing on freebase from question-answer pairs*. In *Conference on Empirical Methods in Natural Language Processing*.
- C. Bieber. 2023. The easy intelligence tests that AI chatbots fails. *Nature*, 619:686–689.
- Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. *Yolov4: Optimal speed and accuracy of object detection*. *Preprint*, arXiv:2004.10934.
- Peter Gärdenfors. 1990. The dynamics of belief systems: Foundations vs. coherence theories. *Revue Internationale de Philosophie*, 44(172 (1)):24–46.
- P. Gärdenfors. 1988. *Knowledge in Flux. Modelling the Dynamics of Epistemic States*. MIT Press.
- Gerd Gigerenzer. 2022. *How to Stay Smart in a Smart World: Why Human Intelligence Still Beats Algorithms*. The MIT Press.
- Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. 2018. Detectron. <https://github.com/facebookresearch/detectron>.
- Omer Goldman, Veronica Latcinnik, Udi Naveh, Amir Globerson, and Jonathan Berant. 2019. *Weakly-supervised semantic parsing with abstract examples*. *Preprint*, arXiv:1711.05240.
- Geoffrey Goodwin and Phil Johnson-Laird. 2005. Reasoning about relations. *Psychological review*, 112:468–93.
- Anirudh Goyal and Y. Bengio. 2022. *Inductive biases for deep learning of higher-level cognition*. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 478.
- J. Grady. 1997. *Foundations of Meaning: Primary Metaphors and Primary Scenes*. University Microfilms.
- Gilbert Harman. 1986. *Change in View: Principles of Reasoning*. Cambridge, MA, USA: MIT Press.
- Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2018. *Mask r-cnn*. *Preprint*, arXiv:1703.06870.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. *Deep residual learning for image recognition*. *Preprint*, arXiv:1512.03385.
- Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. 2017. *Learning to reason: End-to-end module networks for visual question answering*. *Preprint*, arXiv:1704.05526.
- Drew A. Hudson and Christopher D. Manning. 2018. *Compositional attention networks for machine reasoning*. *Preprint*, arXiv:1803.03067.
- Glenn Jocher, Ayush Chaurasia, and Jing Qiu. 2023. *YOLO by Ultralytics*.
- Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2016. *Clevr: A diagnostic dataset for compositional language and elementary visual reasoning*. *Preprint*, arXiv:1612.06890.
- Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2017. *Inferring and executing programs for visual reasoning*. *Preprint*, arXiv:1705.03633.
- P. N. Johnson-Laird and R. M. J. Byrne. 1991. *Deduction*. Lawrence Erlbaum Associates, Inc.
- Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. 2021. *Mdetr – modulated detection for end-to-end multi-modal understanding*. *Preprint*, arXiv:2104.12763.
- M. Knauff, T. Fangmeier, C. C. Ruff, and P. N. Johnson-Laird. 2003. Reasoning, models, and images: behavioral measures and cortical activity. *Journal of Cognitive Neuroscience*, 15(4):559–573.
- Markus Knauff. 2009. A neuro-cognitive theory of deductive relational reasoning with mental models and visual images. *Spatial Cognition & Computation*, 9(2):109–137.
- Markus Knauff. 2013. *Space to Reason*. MIT Press.
- Markus Knauff, Leandra Bucher, Antje Krumnack, and Jelica Nejasmic. 2013. Spatial belief revision. *Journal of Cognitive Psychology*, 25(2):147–156.
- G. Lakoff and M. Johnson. 1980. *Metaphors We Live By*. The University of Chicago Press, Chicago. Citation is based on the reprinted in 2003.
- Percy Liang, Michael I. Jordan, and Dan Klein. 2011. *Learning dependency-based compositional semantics*. *Preprint*, arXiv:1109.6841.

- Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. 2019. [The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision](#). *Preprint*, arXiv:1904.12584.
- David Mascharka, Philip Tran, Ryan Soklaski, and Arjun Majumdar. 2018. [Transparency by design: Closing the gap between performance and interpretability in visual reasoning](#). In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE.
- James McClelland, Felix Hill, Maja Rudolph, Jason Baldridge, and Hinrich Schütze. 2020. [Placing language in an integrated understanding system: Next steps toward human-level performance in neural language models](#). *Proceedings of the National Academy of Sciences of the United States of America*, 117.
- Arvind Neelakantan, Quoc V. Le, and Ilya Sutskever. 2016. [Neural programmer: Inducing latent programs with gradient descent](#). *Preprint*, arXiv:1511.04834.
- Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. 2017. [Film: Visual reasoning with a general conditioning layer](#). *Preprint*, arXiv:1709.07871.
- Marco Ragni and Markus Knauff. 2013. A theory and a computational model of spatial reasoning with preferred mental models. *Psychological review*, 120:561–588.
- Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. [You only look once: Unified, real-time object detection](#). *Preprint*, arXiv:1506.02640.
- T. Regier. 1997. *The Human Semantic Potential: Spatial Language and Constrained Connectionism*. The MIT Press, Cambridge, Massachusetts.
- Anselm Rothe, Brenden M. Lake, and Todd M. Gureckis. 2017. [Question asking as program generation](#). *Preprint*, arXiv:1711.06351.
- Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. 2017. [A simple neural network module for relational reasoning](#). *Preprint*, arXiv:1706.01427.
- Elizabeth Spelke and Sang Ah Lee. 2012. Core systems of geometry in animal minds. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 367:2784–93.
- E. C. Tolman. 1948. Cognitive Maps in Rats and Men. *The Psychological Review*, 55(4):189–208.
- B. Tversky and P. Lee. 1999. How space structures language. In C. Freksa, C. Habel, and K. F. Wender, editors, *Spatial Cognition*, volume 1404 of *LNAI*, pages 157–176. Springer-Verlag.
- Barbara Tversky. 2019. *Mind in Motion*. Basic Books, New York, USA.
- Chien-Yao Wang, Hong-Yuan Mark Liao, I-Hau Yeh, Yueh-Hua Wu, Ping-Yang Chen, and Jun-Wei Hsieh. 2019. [Cspnet: A new backbone that can enhance learning capability of cnn](#). *Preprint*, arXiv:1911.11929.
- Zhonghao Wang, Kai Wang, Mo Yu, Jinjun Xiong, Wen mei Hwu, Mark Hasegawa-Johnson, and Humphrey Shi. 2021. [Interpretable visual reasoning via induced symbolic space](#). *Preprint*, arXiv:2011.11603.
- Qi Wu, Damien Teney, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel. 2016. [Visual question answering: A survey of methods and datasets](#). *Preprint*, arXiv:1607.05910.
- Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B. Tenenbaum. 2019. [Neural-symbolic vqa: Disentangling reasoning from vision and language understanding](#). *Preprint*, arXiv:1810.02338.
- Yeyun Zou and Qiyu Xie. 2020. [A survey on VQA: Datasets and approaches](#). In *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*. IEEE.

VCRMNER: Visual Cue Refinement in Multimodal NER using CLIP Prompts

Yu Bai^{1,2}, Lianji Wang¹, Xiang Liu¹, Haifeng Chi¹, Guiping Zhang^{1,2}

¹School of Computer Science, Shenyang Aerospace University

²National and Local Joint Engineering Laboratory for Multilingual Collaborative Translation Technology

Correspondence: baiyu@sau.edu.cn

Abstract

With the continuous growth of multi-modal data on social media platforms, traditional Named Entity Recognition has rendered insufficient for handling contemporary data formats. Consequently, researchers proposed Multi-modal Named Entity Recognition (MNER). Existing studies focus on capturing the visual regions corresponding to entities to assist in entity recognition. However, these approaches still struggle to mitigate interference from visual regions that are irrelevant to the entities. To address this issue, we propose an innovative framework, Visual Cue Refinement in MNER (VCRMNER) using CLIP Prompts, to accurately capture visual cues (object-level visual regions) associated with entities. We leverage prompts to represent the semantic information of entity categories, which helps us assess visual cues and minimize interference from those irrelevant to the entities. Furthermore, we designed an interaction transformer that operates in two stages—first within each modality and then between modalities—to refine visual cues by learning from a frozen image encoder, thereby reducing differences between text and visual modalities. Comprehensive experiments were conducted on two public datasets, Twitter15 and Twitter17. The results and detailed analyses demonstrate that our method exhibits robust and competitive performance.

1 Introduction

Named Entity Recognition (NER) primarily identifies key entities (e.g., person, locations, organizations) within unstructured textual data sources (Li et al., 2020b). In the context of social media applications, NER technology is primarily used to analyze and track the dynamics of public opinion, major events, and other related information trends. As social networks evolve, the volume of multi-modal data on social media continues to grow, rendering traditional text-based NER methods insufficient for

this new form of data. Consequently, researchers have developed Multi-modal Named Entity Recognition (MNER) (Lu et al., 2018)(Moon et al., 2018). MNER integrates image and text data to identify named entities, effectively resolving the ambiguities present in traditional NER tasks (Lu et al., 2018). It has now become an important research direction in the field of information extraction.

In MNER tasks, textual content and images often exhibit low relevance (Sun et al., 2021)(Hu et al., 2017), with entities usually concentrating on specific visual regions (visual cues). Other regions might interfere with the accurate identification of named entities (Xu et al., 2022)(Zhang et al., 2023a). Early studies (Lu et al., 2018)(Moon et al., 2018)(Wu et al., 2020)(Jia et al., 2022) have explored the inherent correlations between images and text using attention mechanisms. However, this approach does not address the low correlation between images and text, and it is challenging to assess the effectiveness of implicit alignments. Subsequently, research (Sun et al., 2021)(Xu et al., 2022) focused on reducing the influence of irrelevant images on entity recognition by evaluation mechanisms to assess the correlation between entire images and their corresponding textual sentences. For instance, they utilized contrastive learning methods to assess image-text similarity, or they employed pre-trained models for this assessment. Additionally, this approach diminishes the significance of object-level visual regions. Recently, studies (Chen et al., 2022)(Yu et al., 2023)(Zheng et al., 2020) have attempted to exploit object-level visual regions using visual tools such as Mask R-CNN (He et al., 2017). The object-level visual information typically corresponds directly to visual objects with less noise, these visual regions can better assist in entity recognition. However, these visual tools are typically trained solely on visual datasets, which hampers their ability to accurately capture the visual regions pertinent to the entities. Consequently,

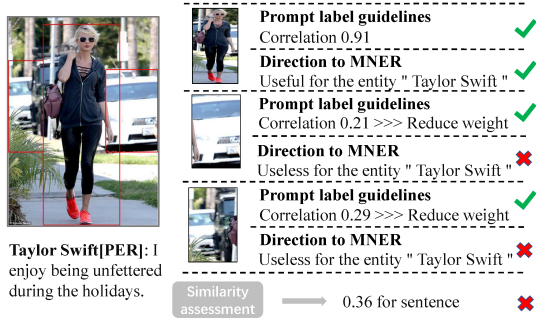


Figure 1: An example shows the problems of different MNER methods.

irrelevant visual regions can mislead the model’s judgments, resulting in error propagation.

As shown in Fig. 1, the similarity score between the input image and the sentence, as evaluated by similarity assessment, is merely 0.36. Despite the person in the image being "Taylor Swift" as mentioned in the text, the relevance of the visual region where this person is located has decreased. In the method that utilizes Mask R-CNN to detect object-level visual regions within the image, three pertinent object regions were identified. However, the "car" in the image does not contribute to the identification of the entity "Taylor Swift". If the significance of this visual region is not diminished, it could potentially mislead the identification process of the entity.

Employing semantic information of entity category words to evaluate the relevance of visual regions helps minimize the interference caused by visual regions that are unrelated to the entities. CLIP (Radford et al., 2021) as a pre-trained multi-modal model, bridges the modalities of images and text. When used as an evaluator, CLIP is capable of assessing the relevance between images and textual content. However, the training corpus for CLIP lacks category words for entity classifications, including person, location, organization, and miscellaneous. Consequently, it is crucial to guide visual region assessments with category words.

In this paper, we propose a new framework Visual Cue Refinement for MNER (VCRMNER) that uses prompts instead of category words to guide the evaluation of visual regions. Specifically, we characterize the types of entities using a series of prompts and compute the center of these prompts’ vector representations to represent the entity category words. The similarity between these category word representations and the representations of vi-

sual cues is assessed with CLIP to determine the relevance of these visual regions. Furthermore, we developed a two-stage modal fusion interactive transformer. First, attention is calculated within each modality separately. Then, the model fuses the modalities together. This balances the differences between text and visuals, avoiding excessive interference from visual representations. By refining visual cues from the frozen image encoder, our model reduces modal discrepancies effectively. Comprehensive experiments were conducted on two public datasets Twitter15 (Zhang et al., 2018) and Twitter17 (Lu et al., 2018). The results and detailed analyses confirm that our method provides robust and competitive performance. Our main contributions are summarized as follows:

- We proposed an innovative architecture VCRMNER that employs a transformer block that combines cross-attention and self-attention, interacting with a frozen visual encoder. This interaction reduces the semantic gap between modalities, thereby more effectively integrating information from different modalities to achieve MNER.
- We designed a prompt-guided visual cue evaluation module that supplements additional semantic information by using prompts to replace entity category words, thereby effectively reducing interference from visual noise unrelated to the entities.
- We conducted extensive experimental verification on two benchmarks, and the experimental results fully demonstrated that our method achieved sota.

2 Related Work

2.1 Prompt learning

The concept of prompt learning involves designing appropriate "prompts" to elicit the desired outputs from the model. The core of this approach lies the idea of not training the model directly for specific tasks, but rather constructing a form of input that enables the model to infer the correct answers based on existing knowledge (Liu et al., 2023). Recently, some studies (Huang et al., 2022) developed a prompt learning method for NER that uses category-specific words to optimize contrastive learning of label representations. This approach, however, is generally limited to the textual modality. In multimodal approaches, (Wang

et al., 2022) proposed a method that leverages the association between prompts and visual images to filter prompts containing entity semantics to assist in entity recognition. This method significantly mitigates the differences between the visual and textual modalities. In contrast, we propose using a specifically designed prompt-driven vision-language model as an evaluator, starting from raw data, to assess whether object-level visual regions (visual cues) are related to entity categories. This approach minimizes interference from irrelevant object-level visual regions and enhances entity recognition through precise visual cues.

2.2 Pretrain vision language model

With the continuous advancement of pre-trained models, significant progress has been made in the fields of computer vision and natural language processing. In this context, Unicoder (Li et al., 2020a) attempted to use a universal encoder to integrate visual and linguistic representations, drawing on the paradigm of cross-lingual pre-training models, inputting both visual and textual data into multi-layer Transformers for multi-task cross-modal pre-training, aimed at image-text retrieval tasks. CLIP (Radford et al., 2021), which uses large-scale image-text pairs and contrastive learning to predict the match between captions and images, thereby understanding the relevance between two different modalities. The CLIP model demonstrates strong generalization ability across various visual tasks without the need for specific task training.

These multimodal visual-language pre-trained models break down modality barriers, narrow the gap between modalities, and exhibit great potential in numerous downstream tasks. For the MNER task, images in image-text pairs often contain visual objects unrelated to entity types, and the existing multimodal visual-language models, with their capability to evaluate image-text associations, provide a technical foundation for the assessment of visual objects.

2.3 MNER

With the increasing amount of multi-modal data on social media platforms, MNER has attracted the attention of many researchers. Based on different image processing methods, we categorize MNER research into two main classes:

(1) Treating the entire image and merging through the interaction between image represen-

tations and text representations in vector space. For example, CNN-LSTM (Lu et al., 2018) introduces a modality attention module that diminishes irrelevant modality information while amplifying the primary modality, used for multi-modal representation. CoA (Zhang et al., 2018) introduces an adaptive co-attention architecture to integrate visual and textual information for MNER. UMT (Yu et al., 2020) has designed a unified multi-modal transformer framework that utilizes an entity span detection task to learn rich multi-modal representations. MAF (Xu et al., 2022) proposes a matching and alignment framework to mitigate the effects of mismatched text-image pairs and enhance the consistency of multi-modal representations. DebiasCL (Zhang et al., 2023a) employs implicit alignment between visual objects and textual entities, using debiasing-based contrastive learning to optimize the shared semantic space between text and images. These methods attempt to leverage entire image to enhance textual representations; however, they overlook the preference for object-level visual regions in the MNER task.

(2) Explicitly extracting object-level visual regions (visual cues) and facilitating interaction between representations of visual regions and textual representations. OCSGA (Wu et al., 2020) utilizes a dense co-attention mechanism to establish both intra-connections and inter-connections between textual entities and visual objects. UMGF (Zhang et al., 2021a) proposes a graph fusion method to learn various semantic relationships between words and multiple visual objects. BGAMNER (Chen et al., 2023) explores the matching relationships between visual regions and words through bidirectional image-text generation. The HamLearning (Liu et al., 2024) enhances text word representations by dynamically aligning image and text sequences and modeling the relationships among the mined visual regions, thereby achieving multi-level cross-modal learning. Although visual regions more accurately point to named entities, the visual regions in images are not always relevant to the entities.

Unlike the above methods, we propose an approach that uses prompts instead of entity category words to evaluate visual cues and dynamically achieves word alignment with visual cues and inter-modal fusion through an interactive transformer.

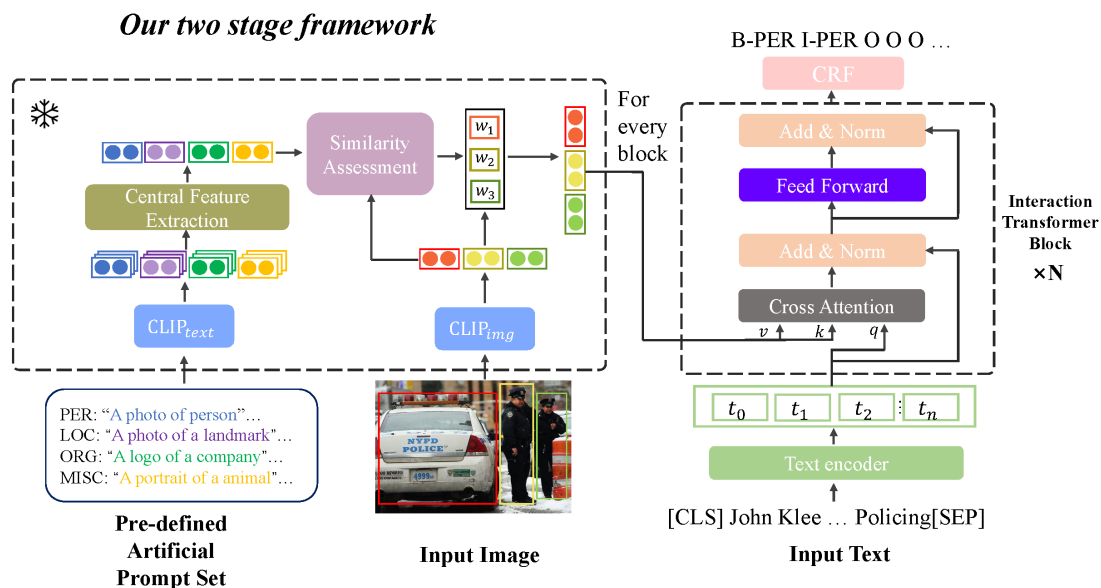


Figure 2: The overall architecture of the approach we proposed.

3 Our Method

Overview our method. Fig. 2 shows our model architecture, which is divided into two main stages. In the first stage, we design the corresponding ten text prompts for each entity category word and map each text prompt to the multi-modal vector space through the clip text encoder. Further, we find the center of the vector representation of the prompts corresponding to each entity category by averaging, and we use this center as the text vector representation of the entity category words. Subsequently, we map the object-level visual regions, obtained by Mask R-CNN, to the shared vector space of image and text via the clip. We then compare the visual representation with the prompts center to evaluate the correlation. In the second stage, we designed an interactive transformer aimed at refining the captured visual cues, enabling the model to more effectively learn the knowledge embedded in the visual modality. This module achieves sustained intra-modal and inter-modal interactions through self-attention and cross-attention mechanisms, thereby narrowing the semantic gap between the textual and visual modalities. Finally, we input the final vector representation obtained from the interaction transformer into the Conditional Random Field (CRF) (Wallach et al., 2004) to complete the sequence annotation decoding process. The following section describes the details of our method.

Formula definition. Given a sentence $S = (w_1, w_2, w_3, \dots, w_n)$, where n represents the total number of words in the sentence. I is the image corresponding to the sentence S . The goal of the MNER task is to combine the image I to label each word in the sentence S to obtain the sequence \hat{Y} . $\hat{y}_i \in \hat{Y}$, \hat{y}_i type is the label of the model predict, $Y = (y_1, y_2, y_3, \dots, y_n)$, y_i represents the type of the real label. According to the BIO notation, the MNER task pre-defines four types of entities $PER, LOC, ORG, MISC$.

Input Embedding: In this paper, we choose RoBERTa (Liu et al., 2019) as the text encoder. Before feeding a sentence S into RoBERTa, the RoBERTa tokenizer segments it into a sequence of word embeddings. Special tokens "[CLS]" and "[SEP]" are inserted at the beginning and end of the word embedding sequence, respectively. This process generates a token sequence $T = \{T_i\}_{i=1}^{N_t} \in \mathbb{R}^{N_t \times d}$, where N_t represents the number of tokens in the sentence, and d represents the dimension of the embeddings. The token sequence is then input into RoBERTa to obtain the vector representation $H_t \in \mathbb{R}^{N_t \times d}$, where h_0 is the vector representation of the entire sentence, and the others are the vector representations of the words in the sentence.

As a Transformer-based image recognition model, Vision Transformer (ViT) (Dosovitskiy et al., 2020) efficiently processes global features of images through the self-attention mechanism.

Therefore, we choose clip-vit as the model’s visual encoder. Given an image I , ViT segments the input image into multiple fixed-size image patches (e.g., 16x16 pixels) and linearly projects these image patches into a series of one-dimensional embedding vectors, represented as $V = \{V_i\}_{i=1}^{N_v} \in \mathbb{R}^{N_v \times d}$, where N_v represents the number of patches in the image, and d represents the embedding dimension.

3.1 Stage one. Prompt-guided visual cues extraction and evaluation

Prompts for design: When evaluating the relevance of object-level visual regions, directly using label words from named entity recognition (such as person, location, and organization) may lead to low recognition accuracy because these keywords do not directly correspond to the content in the CLIP model’s training set. To overcome this limitation, we introduce a prompt-based visual cue method that utilizes a set of carefully designed prompts as substitutes for entity category words, thereby harnessing CLIP’s capability to assess visual regions. We designed ten distinct prompts for each entity category using the large language model GPT-4, which generated and screened prompts based on the MNER task description and the definitions of entity category words. These prompts were then manually filtered to select those that accurately describe the image content and align with the entity type definitions. All prompts are available in the supplementary material.

Visual cues assessment: The predefined prompts are used in place of entity keywords and are encoded through a CLIP text encoder, which converts each prompt into a vector representation. Given $prompt_i^j$ has the following formula to map it to a multi-modal vector representation space:

$$p_i^j = clip_{text}(prompt_i^j) \in \mathbb{R}^d \quad (1)$$

where $p_i^j \in \mathbb{R}^d$ represents the vector embedding of the i -th prompt in the j -th category, with d denoting the dimensionality of the embedding space, and $clip_{text}$ is the CLIP model’s text encoder, responsible for mapping the input prompt into the shared multi-modal vector space.

After obtaining the vector representation of all prompts for each category, we obtain the entity category textual representation by calculating the average of these vectors:

$$c^j = \frac{1}{n} \sum_{i=1}^n p_i^j \quad (2)$$

where $c^j \in \mathbb{R}^d$ represents the centroid vector of the j -th entity category in the multi-modal embedding space, obtained by averaging the n prompt vectors p_i^j , with n denoting the total number of prompts in the category and d being the dimensionality of the embedding space.

It is noteworthy that, since the entity category words are fixed, the prompts we design will also be fixed. Therefore, the process of computing the center of the prompt is performed only once. After obtaining the vector representation of the prompts center, it is passed as a fixed parameter into the model. As the text prompts undergo only a single pass through the text encoder and are fixed, the model we propose does not lead to excessive computational growth.

Given an image, we follow the approach of (Zhang et al., 2021b) by utilizing the Visual toolkit to extract the top m most salient local visual objects. These visual cues $O = \{O_1, O_2, O_3, \dots, O_m\}$ are then resized to 224x224 pixels and mapped into the multimodal representation space using the CLIP visual encoder. The process is represented as follows:

$$v_i' = clip_{img}(o_i) \quad (3)$$

where o_i represents the visual objects. v_i' denotes the vector representation of the i -th object-level visual region in the multimodal vector space.

Subsequently, the text representation of entity category words c^j and the visual representation v_i' are normalized to eliminate the influence of the length of the vector on the similarity calculation so that the similarity is mainly affected by the direction of the vector and not by its length.

$$c^j = \frac{c^j}{D_c}, v_i = \frac{v_i}{D_v} \quad (4)$$

where D_c and D_v are the dimensions of c^j and v_i , and then we use softmax to increase the disparity in similarities.

$$w_sim_i^j = softmax(logit * v_i * c^{jT}) \quad (5)$$

where $w_sim_i^j$ represents the relevance between the i -th visual region and the j -th class entity, $logit$ is a parameter in clip which utilized to enhance the discriminative power between categories.

After obtaining the similarities between visual regions and various categories, we select the similarity $w_sim_i^j$ with the maximum relevance to the visual region as the weight to update the visual representation. At this point, visual regions that are

irrelevant or have low relevance are assigned lower weights.

$$cue_v = \text{cat}(v_i * \max(w_sim_i^j), 2) \quad (6)$$

where cue_v is the visual cues, and cat is the concatenate method that concatenates each visual representation along the second dimension. cue_v is the visual representation fed into the interaction transformer.

3.2 Stage two. Interaction transformer for visual cues refinement

To refine and effectively learn visual cues and achieve interactions and fusion between modalities, we have designed an interaction transformer architecture that interacts with the visual representations obtained from a frozen visual encoder. Within this framework, textual interacts with context through self-attention mechanisms to capture intra-modal semantic associations. The updated textual representations are then fed into alternating cross-attention as queries, thereby facilitating the extraction of associations between text and visual modalities.

In the intra-modal interaction process, self-attention is first used to learn the associative information within the modality. Given the input sentence S , we utilize text encoder to obtain the textual representation:

$$H_t^0 = \text{encoder}_t(S) \quad (7)$$

where $H_t^0 \in \mathbb{R}^{N_t \times d}$ is the textual representation. Subsequently, for intra-modal interactions, we employ the classical multi-head self-attention mechanism to update the textual representations. During this computation process, the updated textual context representations are calculated as follows:

$$Q = H_t^{l-1}W^q, K = H_t^{l-1}W^k, V = H_t^{l-1}W^v \quad (8)$$

$$I'_t = MA(Q, K, V) \quad (9)$$

where l denotes the number of the layers, I'_t represents the representation after intra-modal interactions, $MA(Q, K, V)$ is the multi-head attention mechanism, and Q, K, V are the query matrix, key matrix, and value matrix respectively. I'_t is directly input into the feedforward network:

$$I_t = LN(I'_t + FFN(I'_t)) \quad (10)$$

After completing intra-modal interactions, we employ a cross-attention mechanism to facilitate

inter-modal interactions and refine visual cues, thereby reducing the differences between modalities. With I_t serving as the Query in cross-attention calculations with the visual representations. Given the visual representation cue_v . The computation process is as follows:

$$Q = I_t^{l-1}W^q, K = cue_vW^k, V = cue_vW^v \quad (11)$$

$$I'_m = CA(Q, K, V) \quad (12)$$

$$I_m = LN(I'_m + FFN(I'_m)) \quad (13)$$

where I_m represents the representation after intra-modal interactions, $LN()$ is the layer norm, $CA()$ is the cross attention, Q, K, V are the query matrix, key matrix, and value matrix respectively.

3.3 Label prediction and Model training

A decoder is required to decode the final representations. Extensive research has demonstrated the superior performance of CRF in sequence labeling tasks. CRF is capable of extracting hierarchical information from the semantic space for sequence labeling and has achieved commendable results in numerous sequence labeling tasks. Consequently, we employ a CRF for the purpose of decoding.

$$P(y | G) = \frac{\prod_{i=1}^n E_i(y_{i-1}, y_i, G)}{\sum_{y' \in Y} \prod_{i=1}^n E_i(y'_{i-1}, y'_i, G)} \quad (14)$$

where G is the final vector representation output by the interactive transformer. We choose the maximum likelihood function to calculate the loss and train our model:

$$\mathcal{L} = - \sum_{j=1}^M (\log P(y^j | G^j)) \quad (15)$$

4 Main Result

4.1 Experimental Setup

Dataset. Our experimental tests are consistent with previous studies using two benchmarks: Twitter15 (Lu et al., 2018) and Twitter17 (Zhang et al., 2018). Table 2 shows the basic statistics of the two benchmarks.

Table 2: The basic statistics of twitter15 and twitter17.

Entity Type	twitter15			twitter17		
	Train	Dev	Test	Train	Dev	Test
Person	2217	552	1816	2943	626	621
Location	2091	522	1697	731	173	178
Organization	927	247	839	1674	375	395
Miscellaneous	931	220	720	701	150	157
Total	6166	1541	5072	6049	1324	1351
Num of sentence	4000	1000	3257	3273	723	723

Table 1: The proposed method was compared with several baselines on the Twitter15 and Twitter17 benchmark datasets.

Modality	Methods	twitter15			twitter17		
		P	R	F1	P	R	F1
TEXT	CNN-BLSTM-CRF	66.24	68.09	67.15	80.00	78.76	79.37
	HBiLSTM-CRF	70.30	68.05	69.17	82.69	78.16	80.37
	BERT-CRF	69.22	74.59	71.81	83.32	83.57	83.44
TEXT+IMAGE	AdapCoAtt(Zhang et al., 2018)	69.87	74.59	72.15	85.13	83.20	84.10
	OCSGA(Wu et al., 2020)	74.71	71.21	72.92	-	-	-
	RpBERT(Sun et al., 2021)	71.15	74.30	72.69	-	-	-
	UMT(Yu et al., 2020)	71.67	75.23	73.41	85.28	85.34	85.31
	UMGF(Zhang et al., 2021a)	74.49	75.21	74.85	86.54	84.50	85.51
	MEGA(Zheng et al., 2021)	70.35	74.58	72.35	85.84	87.93	86.87
	HVPNeT(Chen et al., 2022)	73.87	76.82	75.32	85.84	87.93	86.87
	MAF(Xu et al., 2022)	71.86	75.10	73.42	86.13	86.38	86.25
	DebiasCL(Zhang et al., 2023a)	74.45	76.13	75.28	87.59	86.11	86.84
	TGF(Zhang et al., 2023b)	73.88	75.98	74.91	88.42	86.96	87.70
	BGA-MNER(Chen et al., 2023)	78.6	74.16	76.31	87.71	87.71	87.71
	HamLearning(Liu et al., 2024)	77.25	75.75	76.49	86.99	87.28	87.13
VCRMNER(Ours)	75.48	78.23	76.83	87.76	89.79	88.76	

Implementation details. Our experiments were conducted under one Nvidia Tesla T4, using the PyTorch 1.8.0 framework to build the model. We use roberta (Liu et al., 2019) base as the text encoder and clip base as the visual encoder and evaluator. The visual encoder was frozen. The learning rates for the Interaction Transformer and the text encoder were set at $4e-5$, while the learning rate for the CRF was established at $1e-4$. We employed a linear warm-up strategy, with the warm-up rate set at $1e-2$. Within the model, the heads of multi-head attention were set to 8. Interaction transformer blocks were configured 3. The maximum sequence length for the text was determined to be 70, ensuring coverage of all words within the sentences. The model was trained over 40 epochs with a batch size of 18.

4.2 Main Experimental Results and Analysis

To validate the effectiveness of the proposed method VCRMNER, we select a total of 14 baseline methods for comparison, including both pure text-based approaches and multimodal methods. In our experiments conducted on the Twitter15 and Twitter17 datasets, we employed precision (P), recall rate (R), and F1 score (F1) as evaluation metrics. We compared our method with several competitive MNER methods. The results in Table 1 demonstrate that our method has outperformed the current sota methods.

Firstly, under the single-text modality, the method of fine-tuning a pre-trained language model demonstrates significant advantages over the approach using a non-pretrained BiLSTM model. This indicates that the rich prior knowledge embedded in pre-trained models plays a crucial role in the task of NER, thereby enhancing the recognition performance.

Secondly, by comparing methods between multimodal and single-text modalities, we found that approaches utilizing either entire image or visual regions consistently outperformed those relying solely on the single-text modality. These results adequately demonstrate the importance of visual information in MNER tasks. Furthermore, methods that utilize visual regions, such as HVP (Chen et al., 2022), BGA-MNER (Chen et al., 2023), and TGF (Zhang et al., 2023b), have shown clear advantages over those using entire images, like UMT (Yu et al., 2020) and UMGF (Zhang et al., 2021a).

Lastly, methods such as MAF (Xu et al., 2022) and DebiasCL (Zhang et al., 2023a), which assess the significance of visual images, have proven to be crucial in enhancing the effectiveness of MNER tasks, as evidenced by experimental results. This underscores the indispensability of evaluating visual regions in MNER tasks. Compared to methods that assess relevance using sentences and entire image, such as MAF (Xu et al., 2022), DebiasCL

Table 3: Results of ablation study for the MNER task.

Method	twitter15			twitter17		
	P	R	F1	P	R	F1
w/o Inter-former	75.68	76.46	76.06	87.44	87.64	87.54
w/o ev	74.64	76.21	75.42	86.26	88.75	87.49
w/o prompt	74.17	77.96	76.02	87.2	89.27	88.22
VCRMNER(Ours)	75.48	78.23	76.83	87.76	89.79	88.76

Table 4: Results of cross-domain performance on different methods.

cross-domain	17→15			15→17		
	P	R	F1	P	R	F1
HamLearning	69.17	66.84	67.98	71.03	59.4	64.7
BGA-MNER	72.17	67.98	68.71	70.81	59.6	64.91
VCRMNER(Ours)	71.87	68.31	69.08	72.45	60.25	65.11

(Zhang et al., 2023a), and HamLearning (Liu et al., 2024), our approach has demonstrated significant advantages. This further confirms the critical importance of precise evaluation of visual regions in MNER tasks.

4.3 Further experiments and analysis

Ablation Study. To explore the effectiveness of each component of our proposed method, we conducted comprehensive ablation studies. The results of these studies are shown in Table 3, where "Inter-former" refers to the interaction transformer module, "ev" denotes our initial evaluation of visual regions, and "prompt" indicates our method of assessing visual regions using entity category words instead of prompts.

The experimental results demonstrate that every component of our proposed method is effective; removing any part leads to a decrease in model performance. The most significant decline in performance occurs when the evaluation module is ablated, highlighting the critical importance of visual cue assessment in MNER. Eliminating the method of using entity category words for assessment in favor of prompts significantly reduces performance, indicating that prompts align more closely with visual representations than do entity category words. By ablating the Interaction Transformer module and directly concatenating text and visual representations, the lack of effective inter-modal interaction leads to a substantial disparity between visual and textual modalities, resulting in decreased performance.

Cross-domain generalizability analysis. We swapped the test sets of Twitter15 and Twitter17.

For instance, we trained on Twitter17 and tested on Twitter15. From Table 4, we observe that our model maintains competitive performance across various test sets. Compared to previous state-of-the-art methods, our model demonstrates certain advantages, indicating the strong generalization capability of the proposed approach.

5 Conclusion

In this paper, we proposed a new framework to implement the MNER task in two stages. It guides Mask R-CNN to mine visual objects through prompts and obtains visual objects closely related to the entity category words. Through the interactive transformer, we refined the visual cues and narrowed the semantic gap between modalities. We have constructed a variety of experiments to prove that our method is effective and achieves SOTA effects.

6 Limitation

While we have placed the calculation of prompt centers outside the training process to avoid excessive increases in computational complexity during model training and inference, the necessity of evaluating visual cues requires us to employ the CLIP text encoder. Consequently, the final model includes the CLIP text encoder, inevitably leading to an increase in the overall number of model parameters.

References

Feng Chen, Jiajia Liu, Kaixiang Ji, Wang Ren, Jian Wang, and Jingdong Chen. 2023. Learning implicit

- entity-object relations by bidirectional generative alignment for multimodal ner. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 4555–4563.
- Xiang Chen, Ningyu Zhang, Lei Li, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Good visual guidance make a better extractor: Hierarchical visual prefix for multimodal entity and relation extraction. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1607–1618.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.
- Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.
- Yuting Hu, Liang Zheng, Yi Yang, and Yongfeng Huang. 2017. Twitter100k: A real-world dataset for weakly supervised cross-media retrieval. *IEEE Transactions on Multimedia*, 20(4):927–938.
- Yucheng Huang, Kai He, Yige Wang, Xianli Zhang, Tieliang Gong, Rui Mao, and Chen Li. 2022. Copner: Contrastive learning with prompt guiding for few-shot named entity recognition. In *Proceedings of the 29th International conference on computational linguistics*, pages 2515–2527.
- Meihuizi Jia, Xin Shen, Lei Shen, Jinhui Pang, Lejian Liao, Yang Song, Meng Chen, and Xiaodong He. 2022. Query prior matters: a mrc framework for multimodal named entity recognition. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 3549–3558.
- Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. 2020a. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11336–11344.
- Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020b. A survey on deep learning for named entity recognition. *IEEE transactions on knowledge and data engineering*, 34(1):50–70.
- Peipei Liu, Hong Li, Yimo Ren, Jie Liu, Shuaizong Si, Hongsong Zhu, and Limin Sun. 2024. Hierarchical aligned multimodal learning for ner on tweet posts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18680–18688.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Di Lu, Leonardo Neves, Vitor Carvalho, Ning Zhang, and Heng Ji. 2018. Visual attention model for name tagging in multimodal social media. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1990–1999.
- Seungwhan Moon, Leonardo Neves, and Vitor Carvalho. 2018. Multimodal named entity recognition for short social media posts. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 852–860.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.
- Lin Sun, Jiquan Wang, Kai Zhang, Yindu Su, and Fangsheng Weng. 2021. Rpbert: a text-image relation propagation-based bert model for multimodal ner. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 13860–13868.
- Hanna M Wallach et al. 2004. Conditional random fields: An introduction. *University of Pennsylvania CIS Technical Report MS-CIS-04-21*, 24:33–42.
- Xuwu Wang, Junfeng Tian, Min Gui, Zhixu Li, Jiabo Ye, Ming Yan, and Yanghua Xiao. 2022. Promptmner: prompt-based entity-related visual clue extraction and integration for multimodal named entity recognition. In *International Conference on Database Systems for Advanced Applications*, pages 297–305. Springer.
- Zhiwei Wu, Changmeng Zheng, Yi Cai, Junying Chen, Ho-fung Leung, and Qing Li. 2020. Multimodal representation with embedded visual guiding objects for named entity recognition in social media posts. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1038–1046.
- Bo Xu, Shizhou Huang, Chaofeng Sha, and Hongya Wang. 2022. Maf: a general matching and alignment framework for multimodal named entity recognition. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 1215–1223.
- Jianfei Yu, Jing Jiang, Li Yang, and Rui Xia. 2020. Improving multimodal named entity recognition via entity span detection with unified multimodal transformer. Association for Computational Linguistics.

Jianfei Yu, Ziyang Li, Jieming Wang, and Rui Xia. 2023. Grounded multimodal named entity recognition on social media. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9141–9154.

Dong Zhang, Suzhong Wei, Shoushan Li, Hanqian Wu, Qiaoming Zhu, and Guodong Zhou. 2021a. Multimodal graph fusion for named entity recognition with targeted visual guidance. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 14347–14355.

Dong Zhang, Suzhong Wei, Shoushan Li, Hanqian Wu, Qiaoming Zhu, and Guodong Zhou. 2021b. Multimodal graph fusion for named entity recognition with targeted visual guidance. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 14347–14355.

Qi Zhang, Jinlan Fu, Xiaoyu Liu, and Xuanjing Huang. 2018. Adaptive co-attention network for named entity recognition in tweets. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Xin Zhang, Jingling Yuan, Lin Li, and Jianquan Liu. 2023a. Reducing the bias of visual objects in multimodal named entity recognition. In *Proceedings of the Sixteenth ACM international conference on web search and data mining*, pages 958–966.

Zhengxuan Zhang, Weixing Mai, Haoliang Xiong, Chuhan Wu, and Yun Xue. 2023b. A token-wise graph-based framework for multimodal named entity recognition. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pages 2153–2158. IEEE.

Changmeng Zheng, Junhao Feng, Ze Fu, Yi Cai, Qing Li, and Tao Wang. 2021. Multimodal relation extraction with efficient graph alignment. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5298–5306.

Changmeng Zheng, Zhiwei Wu, Tao Wang, Yi Cai, and Qing Li. 2020. Object-aware multimodal named entity recognition in social media posts with adversarial learning. *IEEE Transactions on Multimedia*, 23:2520–2532.

A Appendix

Prompts Used in the Experiment:

The prompts employed in our experiment were categorized into four main groups: *Person (PER)*, *Location (LOC)*, *Organization (ORG)*, and *Miscellaneous*. Each category was designed to capture a specific aspect of visual content, facilitating a comprehensive analysis across diverse image types.

Person (PER)

In the *Person* category, we included a range of human subjects. This involved images such as a

photo of a person, an image of a woman, and a photo of a child. Additional variations included a picture capturing someone, an image of a person with glasses, a portrait of an elderly man, a snapshot of a teenager, a group photo of a family, a candid shot of a person laughing, and a studio portrait of a young adult. These selections aimed at representing different age groups, genders, and social contexts.

Location (LOC)

The *Location* category encompassed various geographical and architectural elements. It featured a photo of a famous landmark, a scenic view of a well-known city, the landscape of a famous natural wonder, a street view in a recognizable city, the architecture of a well-known building, a panoramic view of a historic site, a night shot of a city skyline, a sunrise behind famous city landmarks, a detailed architectural close-up of a historical building, and a picturesque view of a village. This variety ensured that both urban and natural settings were adequately represented.

Organization (ORG)

For the *Organization* category, we focused on institutional and corporate imagery. This included the exterior of a famous institution, a logo of a well-known company, the entrance of a renowned university, a branded product from a famous manufacturer, an official sign of a governmental organization, the front view of an international airport, the headquarters of a global tech company, a franchise store of a popular brand, the emblem of a prestigious college, and a product lineup of a leading electronics brand. These images were chosen to reflect the diversity of organizational structures and their public representations.

Miscellaneous

Lastly, the *Miscellaneous* category covered a wide array of objects and scenes not fitting into the previous categories. This included a close-up photo of a consumer electronic device, a portrait of an animal, an image depicting a traditional cultural festival, a detailed image of a plant, a macro shot of a unique flower, a still life photo of a classical instrument, an artistic depiction of a folk dance, a photo of intricate jewelry, a high definition image of an exotic bird, and a festive scene from a national holiday. The aim here was to introduce a broader spectrum of visual interests and cultural elements.

Neuro-Conceptual Artificial Intelligence: Integrating OPM with Deep Learning to Enhance Question Answering Quality

Xin Kang¹, Veronika Shteingardt², Yuhan Wang¹, and Dov Dori²

¹Tokushima University, Tokushima, Japan

²Technion – Israel Institute of Technology, Haifa, Israel

kang-xin@is.tokushima-u.ac.jp, veronika-s@campus.technion.ac.il,
c612494013@tokushima-u.ac.jp, dori@technion.ac.il

Abstract

Knowledge representation and reasoning are critical challenges in Artificial Intelligence (AI), particularly in integrating neural and symbolic approaches to achieve explainable and transparent AI systems. Traditional knowledge representation methods often fall short of capturing complex processes and state changes. We introduce **Neuro-Conceptual Artificial Intelligence (NCAI)**, a specialization of the neuro-symbolic AI approach that integrates conceptual modeling using Object-Process Methodology (OPM) ISO 19450:2024 with deep learning to enhance question-answering (QA) quality. By converting natural language text into OPM models using in-context learning, NCAI leverages the expressive power of OPM to represent complex OPM elements—processes, objects, and states—beyond what traditional triplet-based knowledge graphs can easily capture. This rich structured knowledge representation improves reasoning transparency and answer accuracy in an OPM-QA system. We further propose transparency evaluation metrics to quantitatively measure how faithfully the predicted reasoning aligns with OPM-based conceptual logic. Our experiments demonstrate that NCAI outperforms traditional methods, highlighting its potential for advancing neuro-symbolic AI by providing rich knowledge representations, measurable transparency, and improved reasoning.

1 Introduction

Integrating neural and symbolic approaches in AI seeks to combine the learning capabilities of neural networks with the interpretability of symbolic reasoning (Besold et al., 2017; Garcez and Lamb, 2023). However, traditional knowledge representations, such as triplet-based knowledge graphs, are limited in capturing complex processes, state changes, and hierarchical relationships inherent in dynamic systems (Wang et al., 2017; Heinzerling

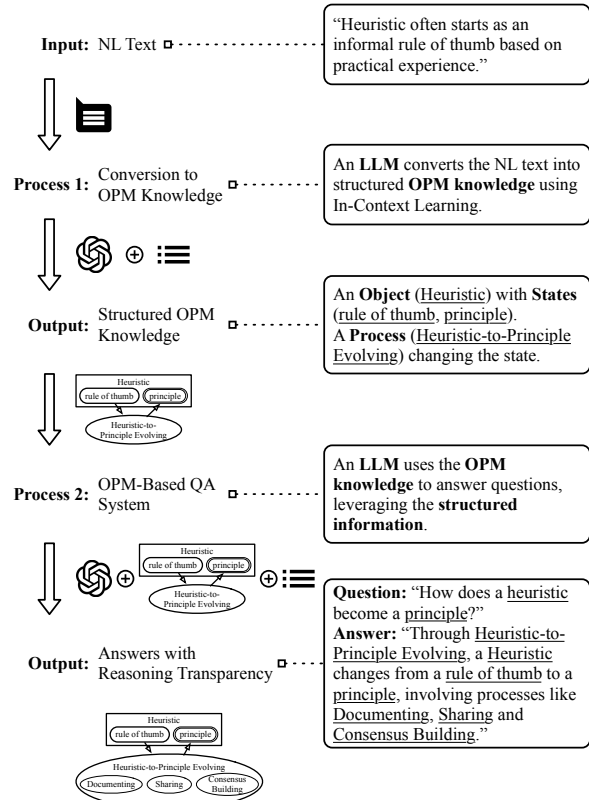


Figure 1: Overview of the NCAI framework, illustrating how the LLM converts natural language text into structured OPM knowledge and uses it in OPM-QA for transparent reasoning. Starting from the text “Heuristic often starts as an informal rule of thumb . . .”, the model generates an OPM model and answers questions by referencing processes like *Heuristic-to-Principle Evolving*.

and Inui, 2021; Shi et al., 2021). Additionally, neural networks are often viewed as a black box due to their opaque decision-making processes, which poses significant challenges in domains requiring transparent reasoning, such as healthcare and finance (Lipton, 2018; Rudin, 2019; Doshi-Velez and Kim, 2017; Tjoa and Guan, 2020).

Recent advancements have focused on enhancing AI reasoning capabilities by integrating language models with external knowledge sources.

For example, combining language models with knowledge graphs has been applied to improve question answering systems (Yasunaga et al., 2021; Oguz et al., 2022; Shi et al., 2021; Zhang et al., 2023). Despite these efforts, fully capturing dynamic behaviors and providing transparent reasoning paths remains a challenge.

To address these limitations, we introduce **Neuro-Conceptual Artificial Intelligence (NCAI)**, a specialization of the neuro-symbolic AI approach, in which the symbolic component is an Object-Process Methodology (OPM ISO 19450:2024) conceptual model. OPM is a conceptual modeling language and methodology that unifies the system’s structural and behavioral aspects within a single model (Dori, 2002; Dori et al., 2016). It represents objects (things that exist) and processes (things that transform objects) in both graphical and textual modalities. By combining OPM with the large language model (LLM), NCAI enhances reasoning transparency and answer accuracy in QA tasks.

An overview of the NCAI framework is illustrated in Figure 1. The framework begins by converting natural language text into structured OPM knowledge using in-context learning with an LLM. This structured knowledge is then used in an OPM-QA, which leverages the expressive power of OPM to represent complex processes and state changes that traditional triplet-based knowledge graphs cannot easily capture. By integrating conceptual modeling with deep learning, NCAI creates a pipeline that transforms unstructured text into a rich knowledge representation, enabling more effective AI reasoning and interpretability.

Our contributions in this work are threefold:

(1) We propose NCAI, which integrates OPM with deep learning to enhance reasoning transparency and answer accuracy.

(2) We develop OPM-QA that utilizes OPM knowledge to improve question-answering quality.

(3) We introduce transparency evaluation metrics to quantitatively assess how faithfully the predicted reasoning aligns with the conceptual logic defined by OPM, and we demonstrate the effectiveness of NCAI through experiments showing improved performance over traditional methods.

2 Related Work

Neuro-Symbolic AI Approaches Neuro-symbolic AI integrates neural networks with

symbolic reasoning to harness the strengths of both paradigms (Besold et al., 2017; Garcez and Lamb, 2023). Challenges in achieving reasoning transparency and interpretability persist, with approaches such as symbolic knowledge distillation (West et al., 2022) and factual knowledge editing (De Cao et al., 2021) addressing these issues. Frameworks like TransferNet (Shi et al., 2021) and interpretable reasoning models for dialogue generation (Yang et al., 2022) aim to provide clear reasoning paths. In sentiment analysis and mental health, neuro-symbolic frameworks like TAM-SenticNet (Dou and Kang, 2024) and causal inference models (Ding et al., 2024b,a) enhance explainability and logical inference. Specifically, in aspect-based sentiment analysis (ABSA), models such as the Multi-Agent Collaboration (MAC) (Kang et al., 2024) and approaches to improve AI transparency using generative agents (Kang, 2024) demonstrate the potential of neuro-symbolic AI in providing transparent and rational sentiment analysis.

Interpretability and Transparency in Language Models Ensuring transparency and interpretability in AI decision-making is critical, particularly in complex systems (Lipton, 2018; Rudin, 2019). Various methods have been developed to enhance the interpretability of language models, including representation dissimilarity measures (Brown et al., 2023), SHAP-based explanation techniques (Mosca et al., 2022), and prompt-based explainers like PromptExplainer (Feng et al., 2024). Evaluation benchmarks for interpretability (Wang et al., 2022) and approaches to improve faithfulness and robustness (El Zini and Awad, 2022; Horovicz and Goldshmidt, 2024; Zhao et al., 2024) further contribute to making language models more transparent. Despite these advancements, achieving full transparency remains challenging, especially in applications requiring a clear understanding of the reasoning process.

Language Models and Knowledge Graphs for Question Answering Integrating language models with knowledge graphs has been a significant focus to enhance QA capabilities. Approaches like QA-GNN (Yasunaga et al., 2021), DRLK (Zhang et al., 2022), and UniK-QA (Oguz et al., 2022) combine language models with graph neural networks and dynamic interactions to improve reasoning in QA tasks. Frameworks such as CIKQA (Zhang et al., 2023) and Triple-R (Kanaani et al., 2024)

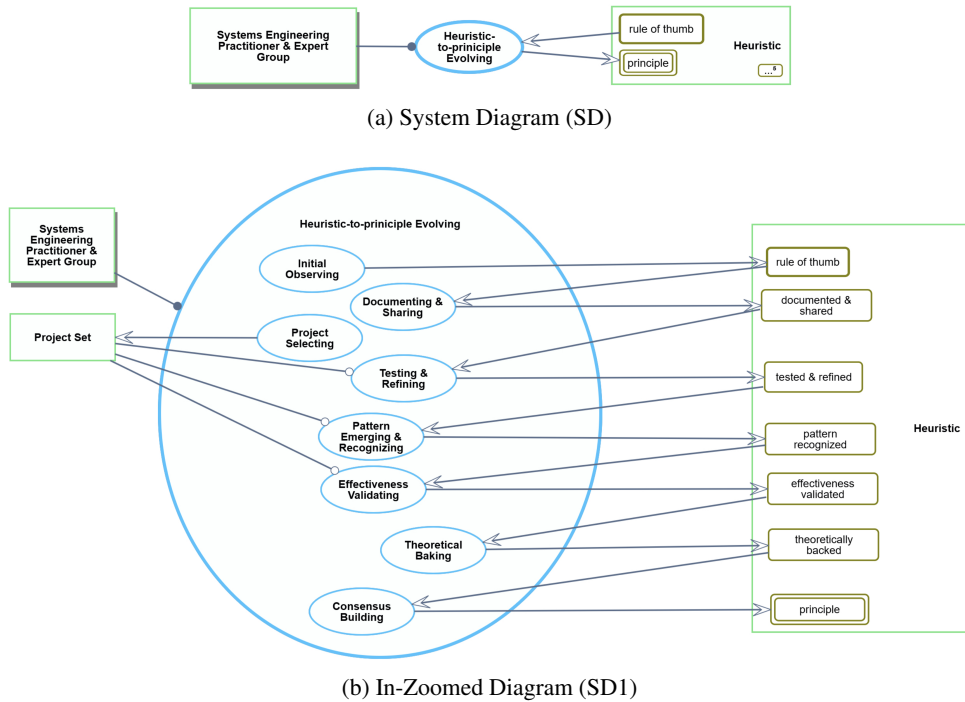


Figure 2: Constructed OPDs illustrating the transformation of a *Heuristic* from a *rule of thumb* to a *principle* through various OPM elements—processes, objects, and states—within the OPM framework.

emphasize the integration of external knowledge sources for more accurate and interpretable reasoning. Additionally, methods like TaPERA (Zhao et al., 2024) enhance faithfulness and interpretability in long-form table QA through content planning and execution-based reasoning. These integrations, while improving performance, often involve complex architectures and still face challenges in achieving complete reasoning transparency.

3 NCAI Framework

3.1 Object-Process Methodology for NCAI

OPM unifies objects and processes within a single model, representing structural and behavioral aspects in both graphical and textual forms (Dori, 2002; Dori et al., 2016). OPM’s bimodal property provides Object-Process Diagram (OPD) and Object-Process Language (OPL), enhancing understanding and reasoning transparency.

To illustrate OPM’s capabilities, we use a running example based on natural language text describing the evolution of a *Heuristic* from a *rule of thumb* to a *principle*. This text serves as the input to the NCAI framework, as shown in Figure 1, and is provided in Appendix A.

Using this text, we constructed OPDs representing the processes and state changes of a *Heuristic*. The diagrams can be created and visualized using

the OPCloud software (Dori et al., 2018; Kohen and Dori, 2021).

Figure 2 presents the constructed OPDs. The System Diagram (SD) in Figure 2a captures the overall transformation of object *Heuristic* from state *rule of thumb* to state *principle* through the process *Heuristic-to-Principle Evolving*. The In-Zoomed Diagram (SD1) in Figure 2b provides a detailed view of the subprocesses involved, such as *Documenting & Sharing*, *Testing & Refining*, *Pattern Emerging & Recognizing*, *Effectiveness Validating*, *Theoretical Backing*, and *Consensus Building*.

The corresponding OPL for the System Diagram (SD) and the In-Zoomed Diagram (SD1) are presented in Appendix B. These OPLs provide a textual representation that details the processes and state changes of the evolution of a heuristic from a rule of thumb to a principle.

OPM’s bimodal property, combining graphical OPD and textual OPL, facilitates a comprehensive representation of complex processes and state changes. The in-zooming mechanism allows for hierarchical decomposition, where processes can be detailed further in subsequent diagrams, enhancing understanding of intricate systems.

3.2 Converting Natural Language to OPM using In-Context Learning

We employ in-context learning to guide the LLM in converting natural language text into OPM models. The process involves providing the LLM with a carefully crafted prompt that includes OPM syntax, semantics, and examples. The prompt details can be found in (Dori and Shteingardt, 2025).

Let T_{NL} be the natural language text (Appendix A) and P_{OPM} the prompt containing OPM instructions and examples. The input to the LLM is:

$$I = P_{\text{OPM}} \circ T_{\text{NL}}, \quad (1)$$

where \circ denotes concatenation. The LLM generates the OPL representation:

$$T_{\text{OPL}} = \text{LLM}(I). \quad (2)$$

This process leverages the LLM’s ability to generate structured textual OPL representations from unstructured text, utilizing in-context learning to guide the model’s output toward the desired OPL format. The OPL generated by the LLM is presented in Appendix C.

While the preliminary results are encouraging, designing prompts that yield accurate and syntactically correct OPM models from free-form text introduces several challenges. These include pinpointing the primary process, focusing on essential conceptual elements, and clarifying ambiguous relationships in the natural language source. To address these issues, we iteratively refine prompts, adjust instructions, and incorporate carefully chosen examples. Through this iterative approach, the LLM learns to better navigate textual ambiguities and produce more coherent OPM models, thus reducing the need for extensive manual refinement and enabling more reliable neuro-symbolic reasoning pipelines.

3.3 OPM Knowledge-Based Question-Answering System

We developed OPM-QA, an OPM knowledge-based Question-Answering system, that integrates OPM knowledge with the LLM to enhance answer accuracy and reasoning transparency. This system is a core component of NCAI, leveraging the structured knowledge representation of OPM to improve the reasoning capabilities of the LLM.

OPM-QA employs in-context learning by providing the LLM with OPL as the OPM knowledge, a set of example question-answer pairs, and the test

questions as context. The knowledge K_{OPL} is derived from the constructed OPM (see Appendix B) and provides a structured and formalized representation. This structured knowledge allows the LLM to reason more effectively when generating answers.

For each test question q_i in the set of test questions Q_{test} , the input to the LLM is formulated as:

$$I_i = K_{\text{OPL}} \circ E_{\text{QA}} \circ q_i, \quad (3)$$

where E_{QA} is the set of example question-answer pairs, and \circ denotes concatenation. The LLM processes this input and generates an answer:

$$a_i = \text{LLM}(I_i). \quad (4)$$

To assess the impact of using structured OPM knowledge on the QA performance, we compare the OPM-QA with a baseline QA system using natural language knowledge (NL-QA). In NL-QA, we replace K_{OPL} with the natural language knowledge K_{NL} , which corresponds to the text provided in Appendix A. This allows us to compare the effectiveness of the structured OPM knowledge against unstructured natural language knowledge in the QA task.

4 Experiments

4.1 Experiment Setup

The purpose of our experiment is to evaluate the effectiveness of the NCAI framework in performing multi-hop reasoning tasks and enhancing reasoning transparency. We aim to compare the performance of OPM-QA with the baseline NL-QA.

Data: We manually developed a dataset of 50 multi-hop reasoning question-answer pairs, following the FanOutQA benchmark (Zhu et al., 2024). These questions are based on the knowledge of the process that transforms informal rules of thumb into well-established principles. The questions require the model to integrate information from multiple statements to arrive at an answer, testing both answer accuracy and reasoning transparency. Examples of the QA pairs are provided in Appendix E, Table 3.

Knowledge Sources: The OPL knowledge K_{OPL} is the OPL generated from the constructed OPM model in Appendix B. The natural language knowledge K_{NL} is the text provided in Appendix A. The QA systems use either K_{OPL} or K_{NL} , along with 5

example QA pairs E_{QA} as context, to answer the 50 test questions Q_{test} .

QA Systems: The QA systems employ in-context learning by providing the LLM with the respective knowledge source, a set of example QA pairs, and the test questions. The LLM used for both systems is GPT-4o (version o1-preview-2024-09-12), with parameters set to $temperature = 0$ and $top_p = 1$ to ensure deterministic output. By using the same LLM and parameter settings, we ensure a fair comparison between OPM-QA and NL-QA. The prompt used in these QA systems is shown in Appendix D. It has been carefully designed to be general enough for QA tasks across various domains and yet instructive enough to guide the model to answer with explicit reference to the OPM elements—processes, objects, and states, thereby increasing reasoning transparency and answer accuracy.

Evaluation Metrics: We evaluate system outputs using a combination of metrics that capture different aspects of answer quality and reasoning transparency. To assess how well the generated answers align with the ground truth in terms of content, we use Loose Accuracy and Strict Accuracy. Loose Accuracy measures the fraction of reference tokens that also appear in the predicted answer after lemmatization, removing stop words, and stripping punctuation, providing a relatively lenient measure of correctness. Strict Accuracy applies a non-linear weighting (with a parameter $k = 1.5$) to penalize partial matches more severely, thus enforcing a stricter standard of correctness.

While Loose Accuracy and Strict Accuracy focus on token-level overlap, ROUGE-1 (R-1), ROUGE-2 (R-2), and ROUGE-L (R-L) (Lin, 2004) quantify lexical overlap through n -gram and sequence-based comparisons, capturing syntactic similarity between the generated answer and the reference. BLEURT (BT) (Sellam et al., 2020) complements these metrics by providing a more semantic-oriented evaluation, as it uses a learned model to judge the meaning and quality of the generated text. The GPT Judge Score (GPT) (Zhu et al., 2024) further evaluates factual consistency and logical coherence, reflecting how well the answer maintains internal logical structure and correctness from a large language model’s perspective.

To address the need for a quantitative measure of reasoning transparency, we propose Transparency Precision (P_T), Transparency Recall (R_T), and

Transparency F1 ($F1_T$). Let \mathcal{E}_p be the set of OPM elements—processes, objects, and states—identified in the prediction, and \mathcal{E}_g the set of OPM elements in the ground truth. Let $\mathcal{E}_{p \cap g}$ be the intersection of these sets, representing correctly matched OPM elements. We define:

$$P_T = \frac{|\mathcal{E}_{p \cap g}|}{|\mathcal{E}_p|}, \quad (5)$$

$$R_T = \frac{|\mathcal{E}_{p \cap g}|}{|\mathcal{E}_g|}, \quad (6)$$

$$F1_T = \frac{2 \cdot P_T \cdot R_T}{P_T + R_T}. \quad (7)$$

Here, P_T measures how accurately the predicted reasoning structure identifies the correct OPM elements, R_T gauges how completely it recovers them, and $F1_T$ balances both. Together, these transparency metrics provide a statistical measure of how faithfully the system’s reasoning aligns with the conceptual logic defined by OPM, offering a principled, quantitative response to calls for more objective assessments of reasoning transparency.

4.2 Results

Table 1 presents the results of our evaluation. For Loose Accuracy, OPM-QA achieves 0.858 ± 0.162 , greatly exceeding NL-QA’s 0.638 ± 0.212 . This indicates that OPM-QA captures a significantly larger fraction of reference tokens under a lenient matching criterion. The difference is statistically significant ($P < 0.001$). Strict Accuracy, which imposes a harsher penalty on partial matches, shows OPM-QA at 0.806 ± 0.213 compared to NL-QA’s 0.530 ± 0.252 . This improvement is also statistically significant ($P < 0.001$), demonstrating that OPM-QA provides answers that are both more complete and more precisely aligned with the ground truth.

Regarding syntactic overlap measures, OPM-QA significantly outperforms NL-QA in all ROUGE metrics. The ROUGE-1 score for OPM-QA is 0.772 ± 0.159 versus NL-QA’s 0.558 ± 0.195 , ROUGE-2 is 0.607 ± 0.201 compared to 0.373 ± 0.198 , and ROUGE-L is 0.715 ± 0.155 compared to 0.504 ± 0.174 . All these differences are highly statistically significant ($P < 0.001$). These results confirm that OPM-QA’s generated answers exhibit considerably more lexical and subsequence-level similarity to the reference answers, adhering

Metric	OPM-QA	NL-QA	P-value
Loose Accuracy	0.858 ± 0.162	0.638 ± 0.212	< 0.001
Strict Accuracy	0.806 ± 0.213	0.530 ± 0.252	< 0.001
ROUGE-1	0.772 ± 0.159	0.558 ± 0.195	< 0.001
ROUGE-2	0.607 ± 0.201	0.373 ± 0.198	< 0.001
ROUGE-L	0.715 ± 0.155	0.504 ± 0.174	< 0.001
BLEURT	0.596 ± 0.165	0.474 ± 0.111	< 0.001
GPT Judge Score	0.920 ± 0.274	0.800 ± 0.404	0.086
Transparency Precision	0.917 ± 0.161	0.759 ± 0.417	0.015
Transparency Recall	0.953 ± 0.143	0.455 ± 0.329	< 0.001
Transparency F1	0.922 ± 0.136	0.546 ± 0.342	< 0.001

Table 1: Evaluation results comparing OPM-QA and NL-QA across correctness, lexical similarity, semantic quality, factual consistency, and transparency. P-values indicate that OPM-QA significantly outperforms NL-QA on all metrics with high statistical confidence, except for GPT Judge and Transparency Precision, where the differences are less significant.

better to the structural and phrasing patterns of the ground truth.

In terms of semantic quality, the BLEURT score for OPM-QA is 0.596 ± 0.165 , which surpasses NL-QA’s 0.474 ± 0.111 . This difference is statistically significant ($P < 0.001$). This suggests that OPM-QA not only matches lexically but also maintains closer semantic fidelity to the intended meanings of the ground truth answers.

Factual consistency and logical coherence are further evidenced by the GPT Judge Score of 0.920 ± 0.274 for OPM-QA compared to NL-QA’s 0.800 ± 0.404 . This difference is not statistically significant ($P = 0.086$), although it still indicates a notable improvement in maintaining factual and logical integrity within the answers.

Most notably, the transparency metrics reveal OPM-QA’s substantial advantage in conceptual alignment. OPM-QA achieves a Transparency Precision of 0.917 ± 0.161 and Transparency Recall of 0.953 ± 0.143 , whereas NL-QA scores 0.759 ± 0.417 and 0.455 ± 0.329 , respectively. The Precision difference is statistically significant ($P = 0.015$), while Recall remains highly significant ($P < 0.001$). Consequently, Transparency F1 for OPM-QA is 0.922 ± 0.136 compared to NL-QA’s 0.546 ± 0.342 , with a P-value of $P < 0.001$. This metric, which balances Transparency Precision and Transparency Recall, underscores the overall superior performance of OPM-QA in aligning with the ground truth both accurately and comprehensively.

Overall, the majority of these metrics demonstrate statistically significant improvements, affirming the superior performance of OPM-QA over

NL-QA. Additionally, the enhancements in Transparency Precision metrics, despite being less statistically significant, further highlight OPM-QA’s effectiveness in achieving greater factual consistency and precision in answers. Detailed evaluation results for 10 representative QA examples and additional evaluation tables are provided in Appendix E, including Tables 3, 4, and 5, which further confirm these findings.

4.3 Discussion

The experimental results confirm that grounding the reasoning process in a conceptual model leads to both improved accuracy and clearer interpretability. Compared to its counterpart, the OPM-QA system consistently aligns its reasoning with the well-defined ontology provided by the OPM model. While the NL-QA system may occasionally produce correct or partially correct answers, it often does so without revealing the underlying conceptual structure. In contrast, OPM-QA not only identifies the correct OPM elements—processes, objects, and states—required to transform the heuristic from one state to another but also presents a reasoning chain that is faithful to the conceptual logic defined by OPM.

Table 2 illustrates a representative case where the question focuses on the processes that guide the heuristic from a documented and shared state to a theoretically backed one. The ground truth answer specifies all of the required processes involved in this transformation. OPM-QA’s answer successfully enumerates each of these processes, maintaining exact alignment with the conceptual elements

Question	What processes change <i>Heuristic</i> from <i>documented & shared</i> to <i>theoretically backed</i> ?
Answer^{GT}	<i>Heuristic</i> changes from <i>documented & shared</i> to <i>theoretically backed</i> through <i>Testing & Refining</i> , <i>Pattern Emerging & Recognizing</i> , <i>Effectiveness Validating</i> , and <i>Theoretical Backing</i> .
Answer^{OPL}	The processes that change <i>Heuristic</i> from <i>documented & shared</i> to <i>theoretically backed</i> are <i>Testing & Refining</i> , <i>Pattern Emerging & Recognizing</i> , <i>Effectiveness Validating</i> , and <i>Theoretical Backing</i> .
Answer^{NL}	The processes that change <i>Heuristic</i> from <i>documented & shared</i> to <i>theoretically backed</i> are Testing & Refinement, Pattern Recognition, Formal Studies, and <i>Theoretical Backing</i> .

Table 2: Comparison of ground truth answer Answer^{GT} and answers from OPM-QA and NL-QA Answer^{OPL} and Answer^{NL} for a sample question, highlighting the matched processes in blue corresponding to the reasoning transparency. While the ground truth and OPM-QA specify all relevant processes, NL-QA mentions only one correct process (*Theoretical Backing*). OPM-QA demonstrates a complete and conceptually aligned reasoning structure, whereas NL-QA’s reasoning chain remains incomplete.

defined by the OPM model. In doing so, OPM-QA achieves high transparency metrics, as measured by the previously defined precision, recall, and F1 scores for transparency. Conversely, NL-QA identifies fewer correct conceptual elements, and in some cases introduces extraneous or irrelevant processes. This discrepancy highlights not merely a difference in correctness, but also a fundamental gap in the clarity and coherence of the reasoning steps offered by the two QA systems.

In addition to the tabular comparison, Figure 3 visually confirms that OPM-QA’s reasoning pathway closely follows the conceptual map provided by OPM. The figure displays an in-zoomed portion of the OPM model (SD1), where the processes critical to changing the heuristic’s state are clearly marked. Each process chosen by OPM-QA is found exactly where it should be according to the conceptual model. Observing these elements in the figure shows that OPM-QA’s improved transparency metrics correspond to verifiable reasoning sequences that can be directly traced in the conceptual diagram. This contrasts with NL-QA, whose reasoning cannot be similarly verified, leaving users and experts uncertain of how and why specific processes were mentioned or omitted.

Taken together, these findings demonstrate that integrating conceptual modeling into the QA framework moves beyond improving standard performance metrics. The introduction of quantitative transparency metrics, supported by direct comparisons in both textual and visual forms, underscores how OPM-QA’s answers are not just better in terms

of correctness, but also clearer, more verifiable, and more trustworthy. This alignment of reasoning with a conceptual backbone is particularly valuable in complex domains where understanding the logic behind an answer is as important as the answer itself. As a result, the synergy between neuro-symbolic reasoning and OPM-based conceptual structures offers a promising avenue toward AI systems that users and domain experts can scrutinize, trust, and ultimately shape with confidence.

5 Conclusion

We propose Neuro-Conceptual Artificial Intelligence (NCAI), a neuro-symbolic approach that integrates OPM conceptual modeling with deep learning to overcome limitations in traditional knowledge representation and reasoning. By embedding OPM-based conceptual logic into a QA system, NCAI captures complex processes and state changes that conventional triplet-based representations and black box neural models struggle to address. Through this structured, bimodal OPM representation, NCAI provides not only improved answer accuracy but also a demonstrably transparent and interpretable reasoning pathway. The introduction of transparency metrics (P_T , R_T , $F1_T$) offers quantitative support for the alignment with OPM-defined conceptual structures, moving beyond purely qualitative assessments of interpretability.

Our experimental results demonstrate that NCAI substantially outperforms traditional methods on both standard accuracy-based measures and

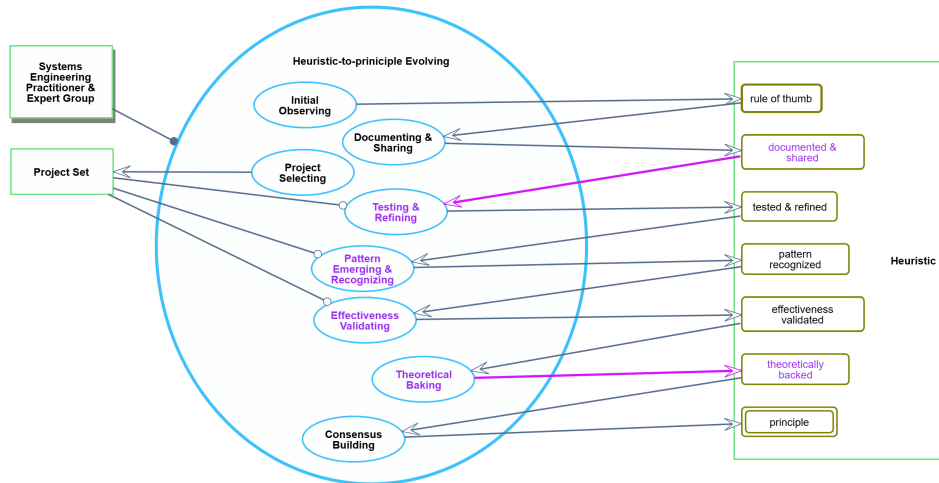


Figure 3: In-Zoomed Diagram (SD1) highlighting the specific processes in blue involved in transforming *Heuristic* from *documented & shared* to *theoretically backed*. These highlighted processes match exactly those identified by OPM-QA in Table 2, demonstrating a coherent and transparent reasoning path.

transparency-focused metrics. By leveraging OPM as a symbolic backbone and employing the LLM under structured guidance, NCAI brings neuro-symbolic AI closer to genuine explainability. Although this work focuses on QA, our conceptual modeling approach may generalize to other tasks requiring robust and interpretable reasoning. Future research will examine scalability to larger, more complex domains, refine prompt designs to handle richer conceptual structures, and integrate our approach with emerging prompting and agentic frameworks. The code and dataset are available on <https://github.com/kangxin/NCAI>.

Limitations

One limitation of our study is that it relies on a relatively small, self-constructed dataset of 50 question-answer pairs. While sufficient for an initial proof of concept, the generalizability and scalability of NCAI to larger and more complex real-world scenarios remain to be explored. In future work, we intend to evaluate NCAI on larger publicly available benchmarks and more intricate conceptual domains, potentially requiring more efficient prompt designs or incremental model updates to handle extensive OPM knowledge.

Additionally, although QA serves as a proof-of-concept task to demonstrate the feasibility of integrating OPM with LLM, applying this approach to other downstream tasks, such as predictive modeling and real-time decision-making in dynamic environments, would require additional domain-

specific adaptations and possibly integration with external data sources. While the OPM-based reasoning structure holds promise beyond QA, confirming its utility in these broader contexts remains an area for future investigation.

Moreover, while our method improves transparency through OPM-driven conceptual alignment, certain ambiguities in the source text can still challenge the strict adherence of the LLM to OPM syntax and conventions. The generated OPM representations might require subsequent refinement by human modelers or more specialized training to ensure full syntactic correctness. Developing standardized benchmarks and further metrics for reasoning transparency, as well as exploring more advanced prompting and agentic design patterns, can help refine the approach, but these steps also remain as future endeavors.

Acknowledgments

This research has been supported by the Project of Discretionary Budget of the Dean, Graduate School of Technology, Industrial and Social Sciences, Tokushima University.

References

- Thomas R Besold, Artur S d'Avila Garcez, et al. 2017. Neural-symbolic learning and reasoning: A survey and interpretation. *arXiv preprint arXiv:1711.03902*.
- Davis Brown, Charles Godfrey, Nicholas Konz, Jonathan Tu, and Henry Kvinge. 2023. [Understanding the inner-workings of language models through](#)

- representation dissimilarity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6543–6558, Singapore. Association for Computational Linguistics.
- Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. **Editing factual knowledge in language models**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Fei Ding, Xin Kang, and Fuji Ren. 2024a. **Neuro or symbolic? fine-tuned transformer with unsupervised lda topic clustering for text sentiment analysis**. *IEEE Transactions on Affective Computing*, 15(2):493–507.
- Fei Ding, Xin Kang, Linhuang Wang, Yunong Wu, Satoshi Nakagawa, and Fuji Ren. 2024b. **Causal inference and prefix prompt engineering based on text generation models for financial argument analysis**. *Electronics*, 13(9):1746.
- Dov Dori. 2002. *Object-Process Methodology: A Holistic Systems Paradigm; with CD-ROM*. Springer Science & Business Media.
- Dov Dori, Ahmad Jbara, Natali Levi, and Niva Wengrowicz. 2018. **Object-process methodology, opm iso 19450–opcloud and the evolution of opm modeling tools**. *Systems Engineering Letters, Project Performance International (PPI) SyEN*, 61:6–17.
- Dov Dori and Veronika Shteingardt. 2025. **Converting knowledge from text to opm models using generative ai prompts: The neuro-conceptual approach**. In *Israel Data Science and AI Initiative 4th Annual Conference*. IDSAI.
- Dov Dori et al. 2016. *Model-based systems engineering with OPM and SysML*, volume 15. Springer.
- Finale Doshi-Velez and Been Kim. 2017. **Towards a rigorous science of interpretable machine learning**. *arXiv preprint arXiv:1702.08608*.
- Rongyu Dou and Xin Kang. 2024. **TAM-SenticNet: A neuro-symbolic ai approach for early depression detection via social media analysis**. *Computers and Electrical Engineering*, 114:109071.
- Julia El Zini and Mariette Awad. 2022. **Beyond model interpretability: On the faithfulness and adversarial robustness of contrastive textual explanations**. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1391–1402, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Zijian Feng, Hanzhang Zhou, Zixiao Zhu, and Kezhi Mao. 2024. **PromptExplainer: Explaining language models through prompt-based learning**. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 882–895, St. Julian’s, Malta. Association for Computational Linguistics.
- Artur d’Avila Garcez and Luis C Lamb. 2023. **Neurosymbolic ai: The 3 rd wave**. *Artificial Intelligence Review*, 56(11):12387–12406.
- Benjamin Heinzerling and Kentaro Inui. 2021. **Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries**. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1772–1791, Online. Association for Computational Linguistics.
- Miriam Horovicz and Roni Goldshmidt. 2024. **TokenSHAP: Interpreting large language models with Monte Carlo shapley value estimation**. In *Proceedings of the 1st Workshop on NLP for Science (NLP4Science)*, pages 1–8, Miami, FL, USA. Association for Computational Linguistics.
- Mohammadamin Kanaani, Sajjad Dadkhan, and Ali A. Ghorbani. 2024. **Triple-R: Automatic reasoning for fact verification using language models**. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16831–16840, Torino, Italia. ELRA and ICCL.
- Xin Kang. 2024. **Transparency in the dimABSA Task with Neuro-Symbolic and Generative AI**. *IEICE Technical Report*, 124(90(CQ2024 16-41)):74–79.
- Xin Kang, Zhifei Zhang, Jiazheng Zhou, Yunong Wu, Xuefeng Shi, and Kazuyuki Matsumoto. 2024. **TMAK-plus at SIGHAN-2024 dimABSA task: Multi-agent collaboration for transparent and rational sentiment analysis**. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pages 88–95, Bangkok, Thailand. Association for Computational Linguistics.
- Hanan Kohen and Dov Dori. 2021. **Designing and developing opcloud, an opm-based collaborative software environment, in a mixed academic and industrial setting: An experience report**. *Academia Letters*.
- Chin-Yew Lin. 2004. **ROUGE: A package for automatic evaluation of summaries**. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Zachary C Lipton. 2018. **The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery**. *Queue*, 16(3):31–57.
- Edoardo Mosca, Ferenc Sziget, Stella Tragianni, Daniel Gallagher, and Georg Groh. 2022. **SHAP-based explanation methods: A review for NLP interpretability**. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4593–4603, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2022.

- UniK-QA: Unified representations of structured and unstructured knowledge for open-domain question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1535–1546, Seattle, United States. Association for Computational Linguistics.
- Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215.
- Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.
- Jiaxin Shi, Shulin Cao, Lei Hou, Juanzi Li, and Hanwang Zhang. 2021. TransferNet: An effective and transparent framework for multi-hop question answering over relation graph. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4149–4158, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Eliot Tjoa and Cuntai Guan. 2020. A survey on explainable artificial intelligence (xai): Towards medical xai. *arXiv preprint arXiv:1907.07374*.
- Lijie Wang, Yaozong Shen, Shuyuan Peng, Shuai Zhang, Xinyan Xiao, Hao Liu, Hongxuan Tang, Ying Chen, Hua Wu, and Haifeng Wang. 2022. A fine-grained interpretability evaluation benchmark for neural NLP. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 70–84, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.
- Peter West, Chandra Bhagavatula, Jack Hessel, Jena Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2022. Symbolic knowledge distillation: from general language models to commonsense models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4602–4625, Seattle, United States. Association for Computational Linguistics.
- Shiquan Yang, Rui Zhang, Sarah Erfani, and Jey Han Lau. 2022. An interpretable neuro-symbolic reasoning framework for task-oriented dialogue generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4918–4935, Dublin, Ireland. Association for Computational Linguistics.
- Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics.
- Hongming Zhang, Yintong Huo, Yanai Elazar, Yangqiu Song, Yoav Goldberg, and Dan Roth. 2023. CIKQA: Learning commonsense inference with a unified knowledge-in-the-loop QA paradigm. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 114–124, Dubrovnik, Croatia. Association for Computational Linguistics.
- Miao Zhang, Rufeng Dai, Ming Dong, and Tingting He. 2022. DRLK: Dynamic hierarchical reasoning with language model and knowledge graph for question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5123–5133, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Yilun Zhao, Lyuhao Chen, Arman Cohan, and Chen Zhao. 2024. TaPERA: Enhancing faithfulness and interpretability in long-form table QA by content planning and execution-based reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12824–12840, Bangkok, Thailand. Association for Computational Linguistics.
- Andrew Zhu, Alyssa Hwang, Liam Dugan, and Chris Callison-Burch. 2024. FanOutQA: A multi-hop, multi-document question answering benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 18–37, Bangkok, Thailand. Association for Computational Linguistics.

A Natural Language Text

Natural Language Text

A plausible set of concise steps of how the process that transforms informal rules of thumb into well-established principles that guide systems engineering practice follows.

1. Initial observation: Heuristics often start as informal rules of thumb based on practical experience.
2. Documentation and sharing: These observations get documented and shared among practitioners.
3. Testing and refinement: The heuristics are tested in various projects and refined based on outcomes.
4. Pattern recognition: As similar heuristics prove useful across multiple projects and domains, recognizable patterns emerge, enabling heuristic generalization.
5. Formal studies: Researchers conduct formal studies to validate the effectiveness of the heuristic.
6. Theoretical backing: The heuristics are connected to underlying theories in systems engineering and related fields.
7. Consensus building: As evidence accumulates, a consensus forms in the systems engineering community about the validity and importance of the heuristic.

B OPL of Constructed OPM Model

OPL for System Diagram (SD)

1. Heuristic can be principle, rule of thumb or at one of five other states. State rule of thumb is initial. State principle is final.
2. Heuristic-to-principle Evolving changes Heuristic from rule of thumb to principle.
3. Systems Engineering Practitioner & Expert Group handles Heuristic-to-principle Evolving.

OPL for In-Zoomed Diagram (SD1)

1. Heuristic-to-principle Evolving from SD zooms in SD1 into Initial Observing, Documenting & Sharing, Project Selecting, Testing & Refining, Pattern Emerging & Recognizing, Effectiveness Validating, Theoretical Baking, and Consensus Building, which occur in that time sequence.
2. Heuristic can be documented & shared, effectiveness validated, pattern recognized, principle, rule of thumb, tested & refined or theoretically backed. State rule of thumb is initial. State principle is final.
3. Systems Engineering Practitioner & Expert Group handles Heuristic-to-principle Evolving.
4. Documenting & Sharing changes Heuristic from rule of thumb to documented & shared.
5. Testing & Refining changes Heuristic from documented & shared to tested & refined.
6. Testing & Refining requires Project Set.
7. Pattern Emerging & Recognizing changes Heuristic from tested & refined to pattern recognized.
8. Pattern Emerging & Recognizing requires Project Set.
9. Effectiveness Validating changes Heuristic from pattern recognized to effectiveness validated.
10. Effectiveness Validating requires Project Set.
11. Theoretical Baking changes Heuristic from effectiveness validated to theoretically backed.
12. Consensus Building changes Heuristic from theoretically backed to principle.
13. Initial Observing changes Heuristic to state rule of thumb.
14. Project Selecting yields Project Set.

C OPL Generated by LLM

OPL Generated by LLM

1. Principle Establishing changes Heuristic from rule of thumb to principle.
2. Practitioner handles Documenting.
3. Practitioner handles Sharing.
4. Researcher handles Formal Studying.
5. Systems Engineering Community handles Consensus Building.
6. Principle Establishing zooms into Observing, Documenting, Sharing, Testing, Refining, Pattern Recognizing, Formal Studying, Connecting, and Consensus Building, which occur in that time sequence.
7. Heuristic can be rule of thumb, documented, shared, tested, refined, recognized pattern, validated, theorized, principle.
8. The state rule of thumb is initial. State principle is final.
9. Observing yields Heuristic.
10. Documenting changes Heuristic from rule of thumb to documented.
11. Sharing changes Heuristic from documented to shared.
12. Testing changes Heuristic from shared to tested.
13. Refining changes Heuristic from tested to refined.
14. Testing requires Project.
15. Refining consumes Outcome.
16. Pattern Recognizing changes Heuristic from refined to recognized pattern.
17. Formal Studying changes Heuristic from recognized pattern to validated.
18. Connecting changes Heuristic from validated to theorized.
19. Connecting consumes Theory.
20. Consensus Building changes Heuristic from theorized to principle.

D Prompt for the QA System

Prompt for OPM-QA and NL-QA Systems

Prompt:

You are a knowledgeable assistant. Your task is to answer questions based on the provided domain knowledge. Your answers should align closely with the domain knowledge, use precise terminology, and remain concise and accurate. Focus on identifying and describing key processes, objects, and states explicitly, and clarify their relationships where relevant.

Domain Knowledge:

[OPL Knowledge in Appendix B or NL Knowledge in Appendix A]

Examples of Question-Answer Pairs:

Q: [example question 1]

A: [example answer 1]

...

Q: [example question N]

A: [example answer N]

New Question:

Q: [question]

A (concise and precise):

E Examples of Questions, Answers, and Evaluation Results

Table 3: 10 example questions and ground truth answers from the QA dataset.

ID	Question	Ground Truth Answer
1	What is the relationship between Testing & Refining and Pattern Emerging & Recognizing in Heuristic evolution?	Testing & Refining changes Heuristic from documented & shared to pattern recognized, and Pattern Emerging & Recognizing then changes it from pattern recognized to effectiveness validated.
2	How does Heuristic achieve theoretical backing before becoming a principle?	Heuristic achieves theoretical backing by undergoing effectiveness validated, pattern recognized, and finally Consensus Building.
3	How does Heuristic change from effectiveness validated to principle?	Heuristic changes from effectiveness validated to principle through Consensus Building.
4	How does the Heuristic-to-principle Evolving process relate to the different states of Heuristic?	The Heuristic-to-principle Evolving process changes from documented & shared, tested & refined, pattern recognized, effectiveness validated, and finally to principle.
5	What processes change Heuristic from rule of thumb to pattern recognized?	Heuristic changes from rule of thumb to pattern recognized through Testing & Refining, and Pattern Emerging & Recognizing.
6	What processes change Heuristic from rule of thumb to tested & refined?	Heuristic changes from rule of thumb to tested & refined through Testing & Refining processes.
7	How does Heuristic transition from effectiveness validated to principle?	Heuristic transitions from effectiveness validated to principle through Consensus Building.
8	How does Heuristic transition from documented & shared to pattern recognized?	Heuristic transitions from documented & shared to pattern recognized through Testing & Refining and Pattern Emerging & Recognizing processes.
9	How does Heuristic evolve from documented & shared to effectiveness validated?	Heuristic evolves from documented & shared to effectiveness validated through Testing & Refining, Pattern Emerging & Recognizing, and Formulation.
10	What processes are involved in Heuristic evolving from rule of thumb to effectiveness validated?	Heuristic evolves from rule of thumb to effectiveness validated through Sharing, Testing & Refining, Pattern Emerging & Recognizing, and Formulation processes.

Table 4: Evaluation results of OPM-QA for 10 examples. Performance was measured using Loose Accuracy (LA), Strict Accuracy (SA), ROUGE-L (R-L), BLEURT (BT), the GPT Judge Score (GPT), Transparency Precision (P_T), Transparency Recall (R_T), and Transparency F1 (F_T).

ID	Answer	LA	SA	R-1	R-2
1	Testing & Refining precedes Pattern Emerging & Recognizing in Heuristic evolution. Testing & Refining changes Heuristic from documented & shared to tested & refined, while Pattern Emerging & Recognizing subsequently changes Heuristic from tested & refined to pattern recognized. Both processes require a Project Set.	1.000	1.000	0.696	0.59
2	Heuristic achieves theoretical backing through Theoretical Baking, which changes it from effectiveness validated to theoretically backed. Subsequently, it undergoes Consensus Building to transition from theoretically backed to principle.	0.619	0.487	0.714	0.40
3	Heuristic changes from effectiveness validated to principle through Theoretical Baking, which transitions it to theoretically backed, followed by Consensus Building, which transitions it to principle.	1.000	1.000	0.750	0.63
4	The Heuristic-to-principle Evolving process transitions Heuristic through the states of rule of thumb (initial), documented & shared, tested & refined, pattern recognized, effectiveness validated, theoretically backed, and finally to principle (final), following a defined sequence of processes.	0.895	0.846	0.756	0.60
5	The processes that change Heuristic from rule of thumb to pattern recognized are Documenting & Sharing, Testing & Refining, and Pattern Emerging & Recognizing.	1.000	1.000	1.000	0.76
6	The processes that change Heuristic from rule of thumb to tested & refined are Documenting & Sharing followed by Testing & Refining.	1.000	1.000	0.957	0.57
7	Heuristic transitions from effectiveness validated to principle through Theoretical Baking, which changes it to theoretically backed, followed by Consensus Building, which changes it to principle.	1.000	1.000	0.750	0.63
8	Heuristic transitions from documented & shared to pattern recognized through the processes of Testing & Refining, which changes it to tested & refined, followed by Pattern Emerging & Recognizing, which changes it to pattern recognized.	1.000	1.000	0.774	0.62
9	Heuristic evolves from documented & shared to effectiveness validated through the processes of Testing & Refining, Pattern Emerging & Recognizing, and Effectiveness Validating, in that sequence.	0.857	0.794	0.828	0.66
10	The processes involved in Heuristic evolving from rule of thumb to effectiveness validated are Initial Observing, Documenting & Sharing, Testing & Refining, Pattern Emerging & Recognizing, and Effectiveness Validating.	0.812	0.732	0.800	0.66

Table 5: Evaluation results of NL-QA for 10 examples. Performance was measured using Loose Accuracy (LA), Strict Accuracy (SA), ROUGE-L (R-L), BLEURT (BT), the GPT Judge Score (GPT), Transparency Precision (P_T), Transparency Recall (R_T), and Transparency F1 (F_T).

ID	Answer	LA	SA	R-1	R-2
1	Testing & Refining provides the practical outcomes and data that enable Pattern Emerging & Recognizing, where recurring success across projects reveals generalizable patterns.	0.417	0.269	0.364	0.19
2	Heuristic achieves theoretical backing through formal studies that validate its effectiveness and by connecting it to underlying theories in systems engineering and related fields.	0.286	0.153	0.421	0.22
3	Heuristic changes from effectiveness validated to principle through Consensus Building, where accumulated evidence and community agreement establish its validity and importance as a guiding principle.	0.778	0.686	0.560	0.43
4	The Heuristic-to-principle Evolving process transitions the heuristic through distinct states: starting as a rule of thumb (Initial Observation), becoming documented and shared (Documentation and Sharing), then tested and refined (Testing and Refinement). It progresses to a pattern recognized state (Pattern Recognition), followed by effectiveness validated (Formal Studies), and finally becomes theoretically backed (Theoretical Backing) and widely accepted through consensus (Consensus Building). Each process step transforms the heuristic into a more formalized and reliable principle.	0.895	0.846	0.493	0.23
5	The processes that change Heuristic from rule of thumb to pattern recognized are Initial Observation, Documentation and Sharing, Testing and Refinement, and Pattern Recognition.	0.692	0.576	0.828	0.51
6	The processes Initial Observation, Documentation and Sharing, and Testing and Refinement change Heuristic from rule of thumb to tested & refined.	0.818	0.740	0.917	0.54
7	Heuristic transitions from effectiveness validated to principle through Theoretical Backing and Consensus Building, where it is connected to underlying theories and gains community agreement on its validity and importance.	0.889	0.838	0.615	0.50
8	Heuristic transitions from documented & shared to pattern recognized through Testing & Refinement, where it is applied across projects, and Pattern Recognition, where recurring usefulness across domains leads to generalization.	0.636	0.508	0.581	0.48
9	Heuristic evolves from documented & shared to effectiveness validated through Testing & Refinement, followed by Pattern Recognition and Formal Studies.	0.643	0.515	0.815	0.64
10	The processes involved in Heuristic evolving from rule of thumb to effectiveness validated are Initial Observation, Documentation and Sharing, Testing and Refinement, Pattern Recognition, and Formal Studies.	0.625	0.494	0.824	0.62

Emergence of symbolic abstraction heads for in-context learning in large language models

Ali Al-Saeedi and Aki Härmä

Department of Advanced Computing Sciences (DACS), Faculty of Science and Engineering, Maastricht University, The Netherlands

Abstract

Large Language Models (LLMs) based on self-attention circuits are able to perform, at inference time, novel reasoning tasks, but the mechanisms inside the models are currently not fully understood. We assume that LLMs are able to generalize abstract patterns from the input and form an internal symbolic internal representation of the content. In this paper, we study this by analyzing the performance of small LLM models trained with sequences of instantiations of abstract sequential symbolic patterns or templates. It is shown that even a model with two layers is able to learn an abstract template and use it to generate correct output representing the pattern. This can be seen as a form of symbolic inference taking place inside the network. In this paper, we call the emergent mechanism abstraction head. Identifying mechanisms of symbolic reasoning in a neural network can help to find new ways to merge symbolic and neural processing.

1 Introduction

Recognizing abstract patterns is a fundamental ability that humans have, allowing them to generalize from a few instances and make inferences on unseen scenarios. LLMs seem to be able to perform similar reasoning tasks and even exceed human performance in some cases (Biever, 2023). Symbolic machine reasoning systems have a long history (Turing, 1950; Berkeley, 1959; Wiener, 1965) but the emergence of the capability in current machine learning systems is not fully understood. Large transformers (Vaswani et al., 2017) and state-space models (Gu and Dao, 2024) exhibit intriguing emergent properties. Extremely large models appear capable of executing tasks like in-context learning (Brown et al., 2020) and chain-of-thought reasoning, which are not directly derivable from their training data. The reasoning abilities of LLMs with tabular, non-text data have been illustrated recently,

as noted in (Jiang et al., 2024). Furthermore, the reasoning prowess of LLMs has been highlighted in robotic control (Zeng et al., 2023), autonomous vehicle navigation (Chen et al., 2023), and the processing of IoT sensor data (An et al., 2024).

The *mechanistic interpretability* of large language models (LLMs) remains a highly active field of research, with numerous recent theories about how specific behaviors manifest in such extensive models (Wei et al., 2022; Nichani et al., 2024; Allen-Zhu and Li, 2024; Huang et al., 2023). This line of research is driven by the common understanding that current LLMs are computationally extremely expensive and environmentally unsustainable, for most use cases, and still from their theoretical capacity (Härmä et al., 2024). The current methods for the minimization of the models, e.g., using distillation techniques, produce only relatively small gains (Xu et al., 2024). A better understanding of the mechanisms can help to improve the design of LLM architectures and training paradigms.

The *induction head* mechanism is considered a key factor behind in-context learning, enabling a language model to identify a recurring pattern from its input and either replicate it in the output or merge it with previously stored knowledge (Olsson et al., 2022). Other theories explaining the emergence of certain behaviors in large language models include the concept of task-vectors (Hendel et al., 2023; Akyürek et al., 2023) and Bayesian inference occurring during the model’s inference phase (Xie et al., 2022).

In this paper, we investigate the ability of small transformer models to recognize, learn, and generalize abstract sequential symbolic patterns, or templates. A template refers to an abstract sequential symbolic pattern that follows a defined structure but can be instantiated with different symbolic elements. For example, the template *ABCABCAB* represents a repeated sequence where *A*, *B*, and *C*

are symbolic placeholders that can take on specific values. Templates represent the underlying structure of patterns, allowing us to explore whether models can learn these abstract patterns and generalize to new instances that the model has never seen, such as *ABACABAC*, that follow similar structural rules. These templates can be instantiated into specific sequences, such as 12312312 or 45645645, by assigning values to the placeholders. Understanding whether models can generalize across unseen instantiations and solve such patterns dynamically during inference is in the focus of this work.

Furthermore, we aim to determine whether a model can recognize such instantiations in-context, that is, whether it can recognize the symbolic mappings during inference without retraining and by using this understanding to solve new instances of the same abstract template. This ability would indicate that the model is not only learning patterns from its training data but also reasoning dynamically based on the input it encounters during inference.

Our experiments demonstrate that small transformer models with two or three layers can successfully solve the task of abstract pattern matching, whereas single-layer models fail to solve the abstract task, which aligns with the theory of single-layer models inability to perform the induction head task (Sanford et al., 2024). Additionally, we observed the emergence of an *abstraction head*. We define an *abstraction head* as an attention mechanism in transformer models that attends to previous instantiations of a pattern in an abstract manner, identifying the structural relationships between symbolic placeholders, and using this information to perform pattern matching on unseen instances with a similar abstract structure.

The identification of the mechanisms of symbolic reasoning emerging in the training of neural networks can help to build new types of neuro-symbolic processing paradigms. Moreover, it may also help to train neural networks with an internal visible symbolic reasoning mechanism. The recent survey by Bhuyan et al. (2024) gives a taxonomy of different neurosymbolic systems. The explicit training of abstraction head configurations could be seen as novel way to implement Type 6 neuro-symbolic systems in their taxonomy.

2 Methodology

To achieve our primary objective of understanding how LLMs perform on abstract sequential symbolic patterns, we designed a controlled experimental setup involving synthetic datasets of templates and their instantiations. To generate the data, we define abstract patterns of length eight using three symbolic variables: *A*, *B*, and *C*. For example, consider the abstract pattern *ABCABCAB*. By assigning specific values to *A*, *B*, and *C*, we generate different instantiations of the same template such as 12312312, 45645645, 78978978, and 15915915.

We generated all possible permutations of length 8, resulting in 6561 patterns. During this process, we observed that some patterns did not include all three variables, such as the pattern *ACACACAC* which only contains variables *A* and *C*, leaving out *B*. To ensure uniformity, we excluded such patterns, requiring that all three variables are present. Additionally, we excluded patterns where the last token appeared only once at the end (e.g. *ABABABAC*), as the last digit in each pattern serves as an evaluation metric in our experimental setup. To avoid duplication, we also treated patterns that are equivalent after instantiation as duplicates. For example, *ABCABCAB* and *ACBACBAC* were considered the same pattern and only one was used. This resulted in 1,806 unique patterns, which we split into 80% for training and 20% for testing to evaluate whether the models can learn new abstract patterns they have never encountered before from context.

After generating the abstract patterns, the next step is to concretize them using instantiations. This is achieved by generating all possible unique combinations of variables *A*, *B*, and *C*, ensuring that each variable is different. Per pattern, we obtain 504 different instantiations.

To prepare the data for training and evaluation, we combined every four instantiation in each input sequence, where every instance should represent the same abstract pattern. This choice ensures that the model is exposed to different representations of the same abstract structure, providing sufficient context for generalization. Furthermore, it ensures that the model is exposed to at least one instantiation of a pattern early in the sequence, before generalizing on the rest of the input sequence, which is a crucial pattern in determining the model’s ability of learning during inference time. For instance, the

abstract pattern $ABCABCAB$ could produce the following input sequence:

[12312312|45645645|78978978|15915915]

To further test the model’s ability, we added the unique values of the variables of each instantiation at the end of the input sequence. Since the variables A , B and C are abstract, we added the variables in the order they appeared first, without arranging them according to the abstract pattern. For example, for the pattern $ACCABBAC$, we append ACB at the end, instead of ABC , which reflects what appears first in the instantiation rather than the abstract pattern itself. Finally, the previous input sequence example would be altered to look as follows: [12312312|45645645|78978978|15915915|123|456|789|159]

In a complementary experiment, we tested the model by placing the variables A , B , and C at the beginning of the input sequence. This experiment exposed the model to the variables first before requiring it to perform the pattern matching task. For instance, the input sequence mentioned earlier would be tweaked to be: [123|456|789|159|12312312|45645645|78978978|15915915]

3 Models

To evaluate the ability of small LLM models to generalize and learn abstract patterns, we experiment with several transformer (Vaswani et al., 2017) architectures, designed to test the models capabilities. The primary task for training these models is autoregressive sequence prediction (Brown et al., 2020). In this setup, the model predicts the n_{th} token based on the $n-1$ previous tokens, using them as context to predict the next output. As for the main experiment, we focus on experimenting with models with one, two, and three layers consisting of eight, four and two attention heads, respectively. Meanwhile, we focus on the 2-layer, 4-head architecture for the second experiment. The hidden size dimension is set to 128 across all architectures, and each model includes a single feed-forward layer. Absolute positional embeddings are used to encode positional information in the input sequences. In this paper, we used the python library X-transformer to build the transformers(Wang, 2024) . As for the training, the models were trained on a batch size of 64 using Adam’s optimizer (Kingma and Ba, 2017), publicly available in PyTorch with a learning rate of 1×10^{-3} for a total of 100,000 steps.

4 Evaluation

To assess the ability of the models to generalize and learn new patterns, we designed two evaluation tasks for the main experiment: the last token prediction and variable matching. Accuracy is the primary evaluation metric for both tasks. For the last token prediction task, accuracy measures the proportion of correct predictions for the last digit in the 4th instantiation. For example, given the input:

[12312312|45645645|78978978|15915915]

the model predicts the bolded final token (5). This task evaluates whether the model can correctly recognize the structure of abstract patterns and generalize to the final token over the unseen abstract patterns in the test set. The second task evaluates the model’s ability to match variable mappings in the sequence. Given the previous input sequence, the model should predict that the next set of tokens should be: [1, 2, 3|4, 5, 6|, 7, 8, 9|1, 5, 9], and the proportion of those variables predicted correctly over the test set, represents the second task of variable matching. We will also look briefly at the training loss to compare the overall performance of the models.

For the complementary experiment, we focus on measuring the accuracy of the second, third, and fourth instantiations, since the model would have seen both the variables and one instantiation, allowing it to generalize on the rest of the instantiations.

5 Results

In this section, we present the results of training the models on instantiations based on abstract patterns, and testing them on the instantiations of the abstract patterns in the test set, which the model has not been exposed to. We will include the training loss of the models, in addition to the last token prediction and variable matching metrics. In addition, we visualize specific attention heads based on their importance and contribution to solving the tasks.

5.1 3-Layers&2-Head

Figure 1 shows the training loss for different runs of the 3-Layer, 2-head model. Although all models converge to the same training loss(0.68), different runs exhibit different behaviors. For instance, we notice bumps emerging in the training loss. Specifically, the model represented in green experiences a bump that occurs after approximately 15,000 steps, and the model represented in pink, where a similar

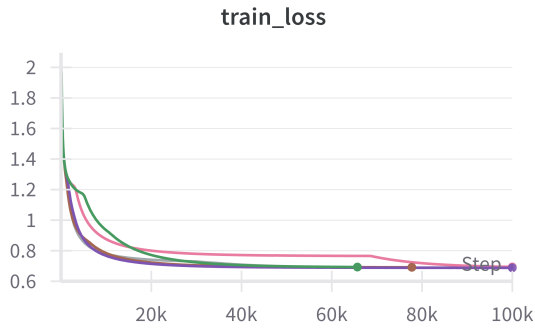


Figure 1: Training loss of five different runs of the 3-Layer, 2-Head model

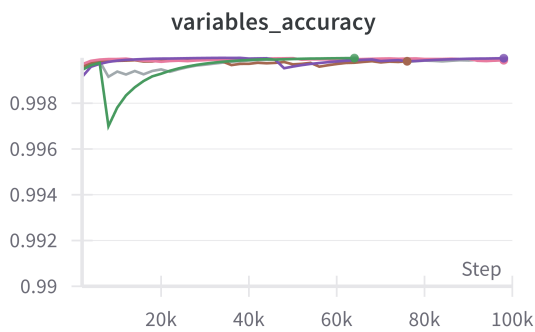


Figure 2: Accuracy of the variable matching task of five different runs of the 3-Layer, 2-Head model

bump occurs after approximately 70,000 steps. We can also see a correspondence between the number of steps where the bumps occur, and the number of steps before the sudden rise in the last digit accuracy shown in Figure 3 for both models represented in green and pink. Figure 3 also shows that not all models achieve perfect accuracy, with two models reaching an accuracy of approximately 0.98. Meanwhile, Figure 2 shows that all models successfully complete the variable matching task, achieving an accuracy of approximately 1.0 across all runs.

To further evaluate the performance of the 3-layer, 2-head model, we showcase an example from the test set, highlighting how one of the runs (represented in gray) solves the pattern.

Predicted pattern by the 3l2h Model:

[? 3 3 1 3 3 1 3 | 7 6 4 2 5 7 6 7 | 4 9 7 5 6 8 9 8 | 7
9 6 3 1 5 9 5 | 3 1 7 | 6 2 7 | 9 5 8 | 9 3 5]

Correct pattern:

[3 3 1 1 7 7 3 7 | 6 6 2 2 7 7 6 7 | 9 9 5 5 8 8 9 8 | 9
9 3 3 5 5 9 5 | 3 1 7 | 6 2 7 | 9 5 8 | 9 3 5]

To understand how the model solves the prob-

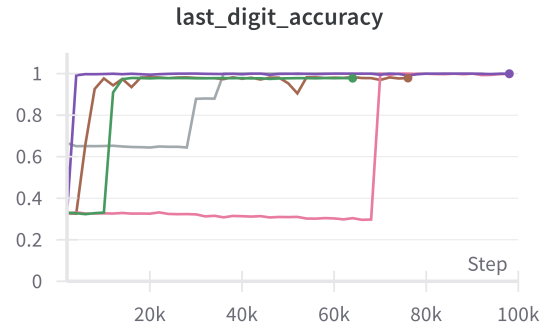


Figure 3: Accuracy of the last digit task of five different runs of the 3-Layer, 2-Head model

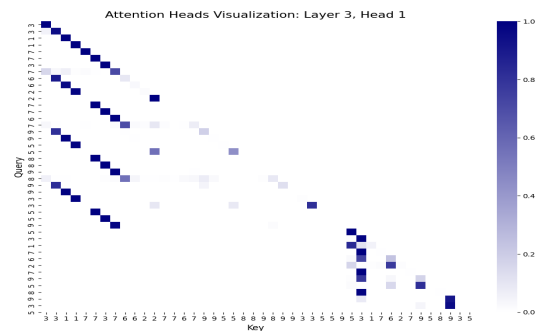


Figure 4: Attention head responsible for abstraction in 3-Layer, 2-Head model

lem, we visualized the attention patterns of solving a test instance during inference time. Among the six attention heads, two attention heads provided us with useful insight into how both tasks are solved (Figures 4 and 5). Figure 4 visualizes the abstraction of patterns, where the attention mechanism focuses on the relationship between the first instantiation and the last three instantiations. Specifically, the tokens in the last three instantiations, which are attending back to the first instantiation to predict the next token. For example, in the second instantiation the bolded 6 in **66227767** is attending to the bolded 3 in **33117737** in the first instantiation. This behavior is consistent across all three later instantiations (second, third and fourth). Moreover, this pattern of attention is not only specific to the first token, with nearly all tokens in the instantiations attending back to their corresponding "abstract" next token in the first instantiation, with the exception of the fifth position.

5.2 2-Layers&4-Head

Figure 6 shows the training loss across multiple runs of the 2-layer, 4-head model, where we observe that only one run fails to converge to the

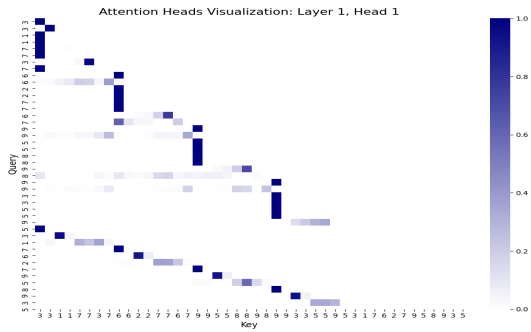


Figure 5: Attention head responsible for variable matching task in 3-Layer, 2-Head model

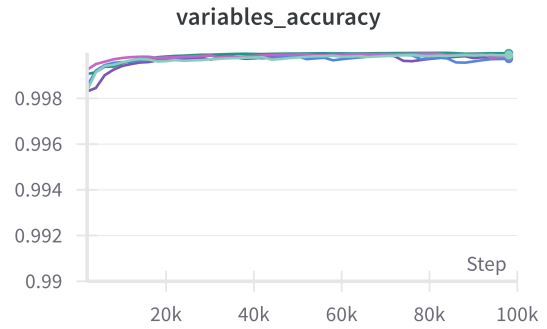


Figure 7: Accuracy of the variable matching task of five different runs of the 2-Layer, 4-Head model

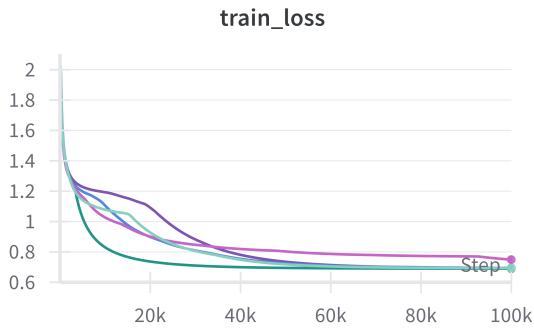


Figure 6: Training loss of five different runs of the 2-Layer, 4-Head model

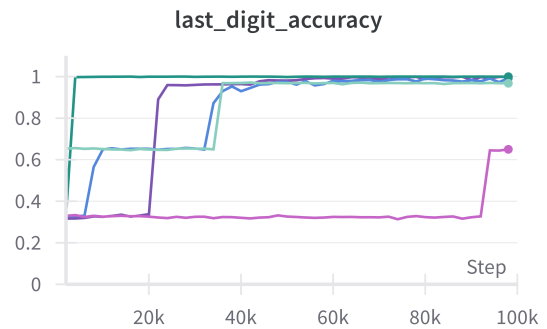


Figure 8: Accuracy of the last digit task of five different runs of the 2-Layer, 4-Head model

minimum loss (0.68). Figure 7 shows that the model consistently succeeds in the task of identifying unique variables. Meanwhile, this is not the case for the last-digit prediction task. Only two runs achieved perfect accuracy, two runs achieved near-perfect accuracy between 0.96 and 0.98, and one run reached a maximum accuracy of 0.6.

We now present an example showing how the model predicts an input sequence using one of the best-performing runs (represented in purple). **Predicted pattern by 214h model:**

```
[? 3 3 3 1 1 3 3 | 9 6 1 2 9 7 6 7 | 9 9 1 5 8 8 9 8 | 1
  9 3 3 8 5 9 5 | 3 1 7 | 6 2 7 | 9 5 8 | 9 3 5 ]
```

Correct pattern:

```
[3 3 1 1 7 7 3 7 | 6 6 2 2 7 7 6 7 | 9 9 5 5 8 8 9 8 | 9
  9 3 3 5 5 9 5 | 3 1 7 | 6 2 7 | 9 5 8 | 9 3 5 ]
```

To better understand how the model predicts patterns, we visualize the attention mechanisms of the two most significant heads out of the eight available attention heads in Figures 9 and 10. In Figure 9, most of the tokens in the second, third, and fourth instantiations are attending back to the first instantiation to predict the next token. For instance, we observe the attention of the highlighted

digits in the second instantiation, **66227767** , to the corresponding highlighted digits in the first instantiation(excluding black), **33117737**. We also observe that the attention is not always directed to the first pattern; at times, it shifts between other instances, such as the bolded 8 in the third instantiation: 9955**88**98 attending back to the bolded 7 in the second instantiation: 662277**67**, which is abstractly the next token. Figure 10 shows the attention mechanism used to solve the second task of matching the variables, where we observe that the attendance was on the correct next token, eleven out of twelve times.

5.3 1-Layer&8-Head

Figure 11 shows the training loss across four runs of the 1-layer, 8-head model, where all the runs fail to converge to the minimum training loss. This failure is reflected in Figure 13, which shows that the model is unable to solve the last digit prediction task. Thus, we do not present any examples of the model’s predictions. In contrast, Figure 12 shows that all runs successfully performed the variable

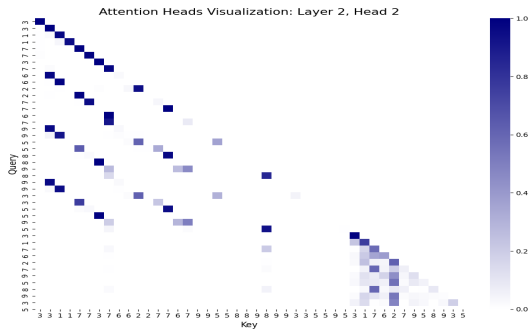


Figure 9: Attention head responsible for abstraction in 2-Layer,4-Head model

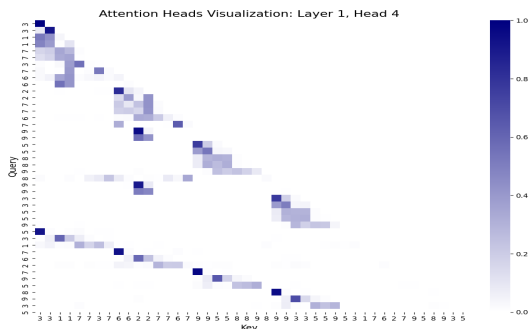


Figure 10: Attention head responsible for the variable matching task in 2-Layer, 4-Head model

matching task.

5.4 2-Layer&4-Head(Experiment 2)

Another experiment involved appending the variables at the beginning of the sequence, followed by the instantiations of the pattern. As shown in Figure 15, all models eventually converge to perfect accuracy. Figure 16 shows the attention head responsible for abstraction in this task, where the second, third, and fourth instantiations attend back to the first instantiation.

6 Discussion

This section analyzes the results presented in the previous section, focusing on the model’s performance on the training loss and the accuracy metrics we defined, attention mechanisms, and the models’ ability to generalize on abstract patterns during inference time.

6.1 3-Layer&2-Head

Referring back to Figure 1, we previously noted that all models converged to a training loss of approximately 0.68 which we assume to be the lowest achievable loss for this task. This limitation is caused by the model’s lack of knowledge about

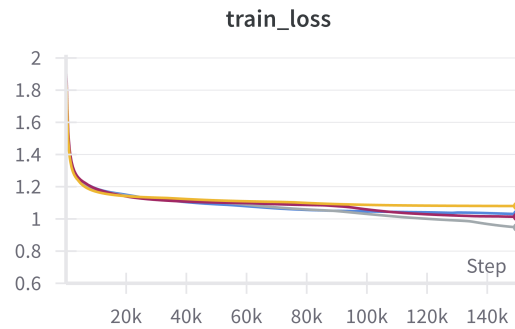


Figure 11: Training loss of four runs of the 1-Layer, 8-Heads model

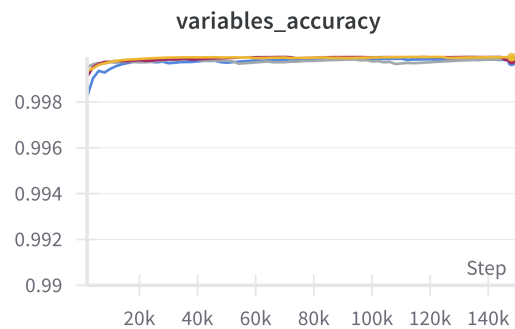


Figure 12: Accuracy of the variable matching task of five different runs of the 1-Layer, 8-Head model

the abstract pattern and the specific values of the variables A , B and C which it only learns during inference, relying on a trial-and-error process.

We also observed bumps in the training loss that correspond to the sudden rise in accuracy shown in Figure 3. We hypothesize that this reflects a change in the models’ behavior, where the models learns a critical strategy that allows it to generalize over the instances in the training set, thus improving its performance on the test set. In Figure 3, we observed that two out of five models did not achieve perfect accuracy but still managed to achieve a minimum accuracy of 0.98. Although we did not find a justification on why it is not acting like the other three models, we can still conclude that the 3-layer, 2-head model succeeds in this task, as all runs can generalize and achieve near-perfect accuracy.

6.2 The abstraction head

We now proceed to analyze the model’s predictions(see 5.1) across the four instantiations to better

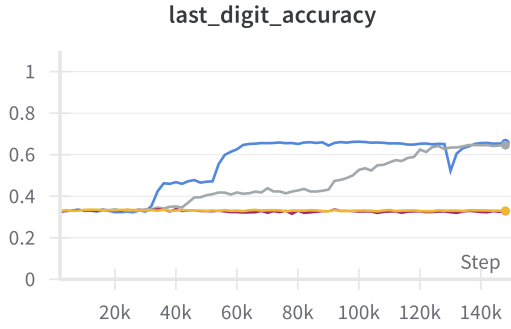


Figure 13: Accuracy of the last digit in the fourth instance of a pattern of four runs of the 1-Layers 8 heads model

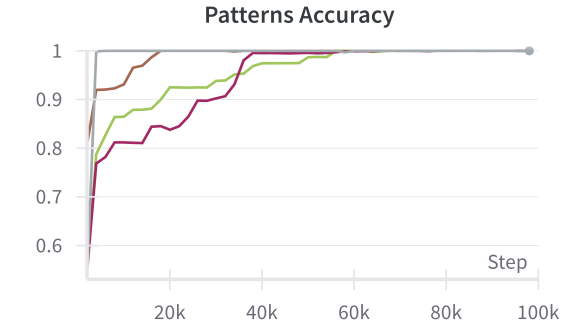


Figure 15: Accuracy of the last three instantiations over the test set

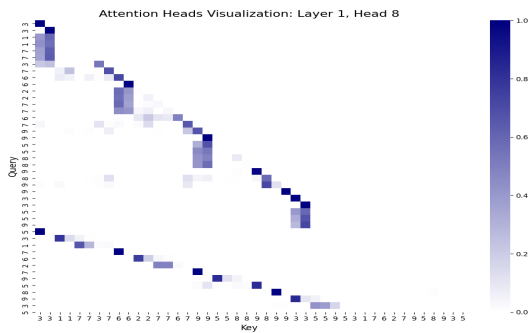


Figure 14: Attention head responsible for variable matching task in 1-Layer, 8-Head model

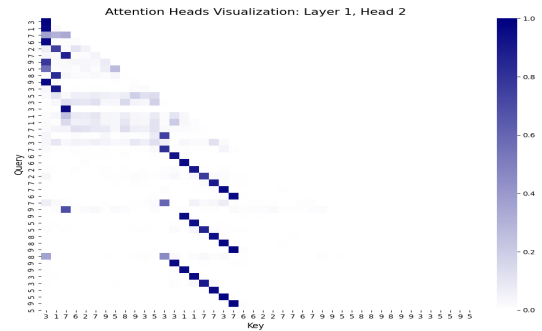


Figure 16: Attention head responsible for abstracting in 2-Layer, 4-Head model

understand its reasoning process and generalization capabilities. The model begins by predicting random tokens in the first eight positions due to the lack of prior information on the abstract pattern and the variables. After predicting the first eight tokens and receiving the prior context, the model gains two pieces of information: the abstract pattern is $AABBCCAC$ and the first three unique digits are $A = 3$, $B = 1$ and $C = 7$. Using this feedback, the model proceeds to predict the next instance. Initially, it tried to predict that the value of A is nine, but updates its understanding after receiving feedback on the correct value of A . The model then correctly predicts the value of A in the following position based on the feedback received, and predicts six. This process is repeated for the next four tokens, BBC , where the model similarly predicts the unseen values of B and C , refining its predictions based on the feedback received. For the last two positions: 66227767 , the model uses its prior knowledge that the abstract pattern is $AABBCCAC$ and that the values of the variables in the second instance are $A = 6$, $B = 2$, and $C = 7$ to predict the last two values correctly.

The same logic is applied to the third and fourth instantiation. Figure 4 shows the attention mechanism we assume is mainly responsible for solving the task of abstract pattern matching. In Figure 4, we observe that the current position in the second, third, and fourth instantiation consistently attends to the corresponding next position in the first instantiation. For example, we mentioned previously, the bolded 6 in **66227767** from the second instantiation attends to the bolded 3 in **33117737** from the first instantiation. This can be seen as the bolded A in **AABBCCAC** attending to the next position, represented as the bolded A in **AABBCCAC**. We identify this attention head as the one responsible for abstracting the patterns, and we refer to it as the *Abstraction Head*, AH. This head seems to have developed the ability to look back at first eight tokens, or the first instantiation and attend to its abstract form. This head aligns with the concept of induction heads, particularly in its ability to perform pattern completion, such as $[A^*][B^*] \dots [A] \rightarrow [B]$, where $A^* \approx A$ and $B^* \approx B$ (Olsson et al., 2022). However, the abstraction head we found operates at a more abstract level, focusing on pattern matching according to a template that has not been

specified to the model, and the model was able to figure out from only instantiations of the data about this abstract pattern. In Figure 2, we observe that the models succeed in the variable matching task within the first 2,000 steps. This task appears to require no abstraction, as the models develop a straightforward matching strategy, which is also shown in the attention pattern shown in Figure 5.

6.3 2-Layer&4-Head

Regarding the training loss of multiple runs of the 2-layer, 4-head architecture, as shown in Figure 6, we observe that four out of five runs successfully converge to the minimum loss. We hypothesize that the failed run is a result of undertraining and with enough training, all models are able to reach a near-perfect accuracy. We also observe a correlation between the last digit accuracy, as shown in Figure 8, and the training loss, specifically, the run represented in pink, which failed to converge to the minimum training loss, and was unable to achieve perfect accuracy in the last digit prediction task. Regarding the model’s prediction (see 5.2), we observe that it uses a similar approach to the 3-layer, 2-head model. Specifically, the model relies on a trial-and-error process when the values of the variables were unknown and started to guess random digits when it did not have any pieces of information about the abstract pattern. We also see that one of the attention mechanisms in the attention heads (Figure 9) is similar to the mechanism observed in Figure 4, specifically what we refer to as an *abstraction head*. However, the abstraction head in Figure 9 not only attends to the first instantiation but also sometimes focuses on the previous instantiations (second and third). While what causes this behavior remains unclear, the model still uses a valid approach to solve the pattern matching task. As shown in Figure 7, all runs successfully performed the variable matching task. Furthermore, Figure 10 reveals that the attention mechanism used for this task is very similar to the one observed in Figure 5, suggesting that the variable matching task is straightforward, where the model develops an attention head that performs a simple tracking back to the digits and predicts the correct variables.

6.4 1-Layer&8-Head

For this architecture, we observed that 1-layer might not be sufficient to solve the last digit prediction task, as shown in Figure 13. However, we found that a single layer can still be used to

tackle non-abstract tasks, such as variable matching, which does not require abstraction. This is demonstrated in Figure 12. The mechanism used for this task, shown in Figure 14, aligns with the behavior observed in other models.

6.5 2-Layer&4Heads(Experiment 2)

The complementary experiment confirmed that 2-layer models are capable of solving the task of abstract pattern matching, as demonstrated in Figure 15. Additionally, we observed the emergence of the *abstraction head*, illustrated in Figure 16. However, unlike the previous task, we found that some models were able to solve this task without developing similar abstraction heads. This suggests that these models may have discovered an alternative mechanism to solve the problem, which needs further investigation in future work.

7 Conclusions

In this study, we investigated the ability of small transformer models to recognize, learn, and generalize abstract sequential symbolic patterns through controlled experiments. We demonstrated that models with two or three layers successfully perform this task, unlike single-layer models. A key finding was the emergence of the *abstraction head*, an attention mechanism that directs its focus toward the first instantiation in an abstract manner. Our findings on the emergence of abstraction heads provide a foundation for advancing neuro-symbolic processing paradigms, potentially enabling the development of new Type 6 neuro-symbolic systems which contain symbolic processing inside a trained neural network.

8 Limitations of the work

The results of this paper are based on a very simplified experiments based on sequences of numbers. It is possible that the proposed abstraction mechanism is not a primary mechanism in reasoning tasks based on human language or in cases where the entities have more complex relations. The study was also limited to small transformer models with only 1-3 layers of self-attention models. It is possible that different abstraction mechanisms emerge in models of more layers, which may not be visible in the experiments reported in this paper.

References

- Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2023. [What learning algorithm is in-context learning? Investigations with linear models.](#) *arXiv preprint*. ArXiv:2211.15661 [cs].
- Zeyuan Allen-Zhu and Yuanzhi Li. 2024. [Physics of Language Models: Part 3.1, Knowledge Storage and Extraction.](#) *arXiv preprint*. ArXiv:2309.14316 [cs].
- Tuo An, Yunjiao Zhou, Han Zou, and Jianfei Yang. 2024. [IoT-LLM: Enhancing Real-World IoT Task Reasoning with Large Language Models.](#) *arXiv preprint*. Version Number: 2.
- Edmund Callis Berkeley. 1959. *Symbolic Logic and Intelligent Machines*. Literary Licensing, LLC.
- Bikram Pratim Bhuyan, Amar Ramdane-Cherif, Ravi Tomar, and T. P. Singh. 2024. [Neuro-symbolic artificial intelligence: a survey.](#) *Neural Computing and Applications*, 36(21):12809–12844.
- Celeste Biever. 2023. [ChatGPT broke the Turing test — the race is on for new ways to assess AI.](#) *Nature*, 619(7971):686–689. Bandiera_abtest: a Cg_type: News Feature Publisher: Nature Publishing Group Subject_term: Computer science, Mathematics and computing, Technology, Society.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language Models are Few-Shot Learners.](#) *arXiv:2005.14165 [cs]*. ArXiv: 2005.14165.
- Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. 2023. [Driving with LLMs: Fusing Object-Level Vector Modality for Explainable Autonomous Driving.](#) *arXiv preprint*. ArXiv:2310.01957 [cs].
- Albert Gu and Tri Dao. 2024. [Mamba: Linear-Time Sequence Modeling with Selective State Spaces.](#) *arXiv preprint*. ArXiv:2312.00752 [cs].
- Roe Hendel, Mor Geva, and Amir Globerson. 2023. [In-Context Learning Creates Task Vectors.](#) *arXiv preprint*. ArXiv:2310.15916 [cs].
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. [A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions.](#) *arXiv preprint*. ArXiv:2311.05232 [cs].
- Aki Härmä, Marcin Pietrasik, and Anna Wilbik. 2024. [Empirical Capacity Model for Self-Attention Neural Networks.](#) *arXiv preprint*. ArXiv:2407.15425 [cs, stat].
- Ruya Jiang, Chun Wang, and Weihong Deng. 2024. [Seek and Solve Reasoning for Table Question Answering.](#) *arXiv preprint*. ArXiv:2409.05286 [cs].
- Diederik P. Kingma and Jimmy Ba. 2017. [Adam: A method for stochastic optimization.](#) *Preprint*, arXiv:1412.6980.
- Eshaan Nichani, Alex Damian, and Jason D. Lee. 2024. [How Transformers Learn Causal Structure with Gradient Descent.](#) *arXiv preprint*. ArXiv:2402.14735 [cs, math, stat].
- Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. [In-context learning and induction heads.](#) *Preprint*, arXiv:2209.11895.
- Clayton Sanford, Daniel Hsu, and Matus Telgarsky. 2024. [One-layer transformers fail to solve the induction heads task.](#) *Preprint*, arXiv:2408.14332.
- A. M. Turing. 1950. [Computing machinery and intelligence.](#) *Mind*, LIX(236):433–460.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention Is All You Need.](#) *arXiv preprint*. ArXiv:1706.03762 [cs].
- Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. [Label words are anchors: An information flow perspective for understanding in-context learning.](#) *Preprint*, arXiv:2305.14160.
- Phil Wang. 2024. [lucidrains/x-transformers.](#) Original-date: 2020-10-24T22:13:25Z.
- Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. [Emergent Abilities of Large Language Models.](#) *arXiv preprint*. ArXiv:2206.07682 [cs].
- Norbert Wiener. 1965. *Cybernetics, Second Edition: or the Control and Communication in the Animal and the Machine*, 2nd edition edition. Mit Pr, Cambridge, MA, USA.
- Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. [An Explanation of In-context Learning as Implicit Bayesian Inference.](#) *arXiv preprint*. ArXiv:2111.02080 [cs].

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. [A Survey on Knowledge Distillation of Large Language Models](#). *arXiv preprint*. ArXiv:2402.13116.

Fanlong Zeng, Wensheng Gan, Yongheng Wang, Ning Liu, and Philip S. Yu. 2023. [Large Language Models for Robotics: A Survey](#). *arXiv preprint*. ArXiv:2311.07226 [cs].

A Appendix: Abstraction Heads

One aspect we explored was whether the abstraction head tends to emerge in a specific layer(e.g. the first or the last). Figure 17 shows the visualization of another successful run of the 3-Layers, 2-Head architecture. The abstraction head shown in Figure 4 emerges in the third layer, while the abstraction head in Figure 17, emerges in the second layer. However, even tho attention heads tend to appear in the last few layers, none of the experiments showed an abstraction head in the first layer. Research showed that attention heads that extract and makes use of critical information such as label appear in deep layers (Wang et al., 2023). This might suggest that abstraction heads appear in Deep Layers(i.e. layers closer to the output), and might show that attention heads that require abstraction(label, abstract pattern, etc.), are usually closer to the output, even in small models, which is a matter of further investigations.

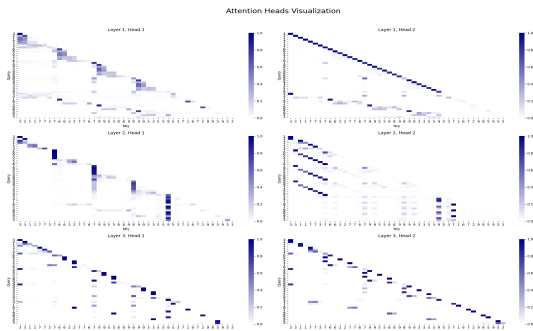


Figure 17: Visualization of all attention heads from another run

Linking language model predictions to human behaviour on scalar implicatures

Yulia Zinova and David Arps and Katharina Spalek and Jacopo Romoli

Heinrich Heine University Düsseldorf

{zinova, david.arps, katharina.spalek, jacopo.romoli}@hhu.de

Abstract

We explore the behaviour of language models on adjectival scales in connection with negation when prompted with material used in human experiments. We propose several metrics extracted from the language model predictions and analyze those metrics in relation to human data. We then use these metrics to propose new items to be tested in both human and model-based experiments.

1 Outline

In this paper, we describe various experiments that explore the relationship between scalar implicatures and language modeling. Scalar implicatures are inferences such as the one in (1).

- (1) a. The project is difficult.
b. \rightsquigarrow The project is not impossible.

Here, *difficult* and *impossible* form a scale. The inference is such that using the weaker item on that scale (1-a) leads to the negation of the stronger item (1-b).

In the first part of the paper, we aim to elicit an implicature (1-b) from the language model as the next token prediction. To do this, we prompt the model with the original sentence (containing a weak scalar item) followed by the repetition of the initial portion of the same sentence and a negation (2). We base this experiment on the material presented in Sun et al. 2024 and explore the output as well as the underlying processing of prompts that include negation.

- (2) The project is difficult. This means that the project is not (PROMPT)

Next, we introduce metrics that are based on the model behaviour. These metrics prioritize lexical items that are likely to co-occur in the top predictions of the language model. We use these metrics

to automatically extract new pairs of adjectives. From the obtained list of pairs we then select such pairs where the adjectives are on a scale and repeat the negation experiment using corpus data and both established and new adjective pairs. We analyze the model behaviour for both sets of pairs, focusing on the desirable *crossing* pattern of adjectival activation when the model encounters the negation.

In the last part of the paper, we use the proposed model-related metrics in connection with human experiments. On one hand, with the help of one of the metric variants we can explain some part of the variability on the human ratings for scalar implicatures and negative strengthening. On the other hand, we discover that almost all the scales used in human experiments receive low values according to our metrics. We then extract new adjective pairs from the model and propose a set of scalar pairs to use in future human experiments that would be more evenly distributed from the perspective of the language model.¹

2 Language models and negation

2.1 Introduction

Recent papers have demonstrated time and again that negation poses a challenge for language models that is not resolved by increasing the model and the dataset size (Kassner and Schütze, 2020; Lipkin et al., 2023; Zhang et al., 2023; Sullivan, 2024). This becomes especially relevant in connection with the natural language inference task: the performance on datasets that focus on negation even after fine-tuning is significantly lower than on general datasets where negation does not play a special role (Hossain et al., 2020; Truong et al., 2023).

Another challenge for language models is related to pragmatic inferences that are not tradition-

¹Our implementation is made available at <https://github.com/davidarps/lm-scales>

ally included in the NLI datasets but are relevant for human daily conversations (Hu et al., 2023). These include presuppositions, scalar implicatures and other related inference types, such as negative strengthening, extensively studied in the theoretical literature (Horn 1984; Hirschberg 1985; Degen 2015; Gotzner and Romoli 2022, among others) but severely underrepresented in NLI datasets (Jeretic et al., 2020). Negative strengthening refers to a type of implicature whereby the meaning of a scalar expression containing a negation (3-a) is enriched using its non-negated antonym (example (3-b), (27) in Gotzner and Romoli 2022).

- (3) a. The room is not large.
 b. \rightsquigarrow The room is (rather) small.

A recent dataset that aims to address the problem of underrepresented inference types provides premise-hypothesis pairs that include scalar items, such as *some/not all* and *warm/hot* (SIGA, Nizamani et al. 2024). It contains premise-hypothesis pairs preceded by a context (4-a) and labeled as *contradiction*, *entailment* or *neutral*. In case of example (4), the label for the pair (4-b)-(4-c) is *contradiction*.

- (4) a. Five weeks later, I had my first batch of polished stones in nearly 40 years. I was also disappointed.
 b. The polished stone looked good
 c. The polished stone looked great

The challenge in creating such datasets, apart from extracting or generating the data, is data annotation, especially given the fact that the rate with which humans predict scalar implicatures in experimental studies varies significantly between items (Van Tiel et al., 2016; Sun et al., 2018; Gotzner et al., 2018b; Ronai and Xiang, 2022). Multiple experimental studies aimed to explain this variation with the help of various linguistic properties as well as the relation to priming (Ronai and Xiang, 2023; Lacina and Gotzner, 2024) but achieved only partial success: no combination of the proposed factors could explain the full range of human rating variation.

Since the only available naturalistic dataset for scalar inferences (SIGA, Nizamani et al. 2024) focuses on the implicatures or their absence in a positive context, it does not allow to evaluate the behaviour of the language models with respect to the scalar terms in the context of negation. Such an evaluation is an important missing step, since the

underlying process of implicature computation involves reasoning about the alternatives and their negated variant (Van Tiel et al., 2016; Gotzner et al., 2018a). For this reason, in the first experiment we test the behaviour of the (smallest) OPT language model for next word prediction, trying to elicit a completion following a prompt that includes a negation similar to the experimental setup of Van Tiel et al. 2016.

We show that the language model exhibits a significant amount of copying in such a scenario, what on the surface level looks like ignoring the negation (and leads to a contradicting sentence completion). We examine the underlying representations and find evidence for the desired trends in processing the negation that often do not reach the level to become visible in the output.

2.2 Experiment 1

In the first experiment we evaluate negation processing by a language model using both scales and contexts from Sun et al. (2024). To approach this task, we test whether a language model is likely to predict an adjective compatible with a scalar implicature as the next word. We use a setting that is compatible with computing a scalar implicature based on the gradable adjectives.

Model Previous work has shown that models of different sizes show similar performance on token-level predictions related to scalar implicatures (Arps and Zinova, 2024). Therefore, all experiments are conducted with only one model, namely OPT-125m (Zhang et al., 2022). OPT-125m is a decoder-only (causal) language model with twelve layers and an embedding size of 768. It has been trained on next-token prediction on 180B tokens of predominantly English books and web-crawled data from different domains.

Data In this experiment we use the scales and the sentences from Sun et al. (2024). The prompts for the experiment were constructed following the scheme in (5): the first sentence contains a weak adjective (5-a) and is taken from the material of Sun et al. (2024). In our prompt, this sentence is followed by a second sentence that starts with a connector (5-b) and continues with the same prompt as in (5-a) repeated up to the adjective position and followed by a negation (5-c). We then obtained the model predictions over all the vocabulary for the next word following the complete prompt (5) (including negation). The expected item according

to the implicature pattern would be (the negation of) *brilliant*, the stronger alternative of *intelligent*.

- (5) a. This student is intelligent.
 b. Put differently,²
 c. this student is not ...

The results of this experiment are in line with the previous predictions concerning language models and negation: in most cases the model predicted the same weak adjective it observed in the first part of the prompt as one of the top predictions. The same weak adjective has rank 0 after negation in 383 out of 1276 cases (30%), rank 1 in 102 cases (8%) and rank between 2 and 4 in another 145 cases (11%). This means that in 30% of all the cases the resulting sentence (in this case "*Put differently, this student is not intelligent*") contradicts the preceding part of the prompt (5-a). This is not surprising given the difficulty of the task, previous findings and the absence of fine-tuning.

In order to check whether the model ignores the negation, as suggested by Kassner and Schütze (2020) and by the surface evaluation above, we have traced the activation of the weak and the strong gradable adjectives at the position before the weak gradable adjective is introduced in (5-a), at the same point (before the negation) in the last part of the prompt (5-c) and after the negation (end of the prompt (5-c)).

Despite the very high surface copying rate, we can observe that the model does not ignore the negation, which is visible on the cumulative representations of scalar adjective activation for a specific scale. To obtain such a representation, we have collected the logit activation at the following points, accumulating them over various prompts: [0] at the beginning of the first sentence, [1] before the scalar adjective in the first sentence, [2] after the adjective in the first sentence, [3] at the beginning of the second sentence, [4] before the negation in the second sentence, and [5] after the negation in the second sentence. Example (6) shows these points in the exemplar prompt provided above in (5).

- (6) [0] This student is [1] intelligent [2]. Put differently, [3] this student is [4] not [5] ...

The case illustrated in Fig. 1 demonstrates the desired behaviour of a language model in the context

²We have tested various connectors such as *It means that* and *In other words* as well as an empty connector, but did not observe any significant variation in the model behaviour.

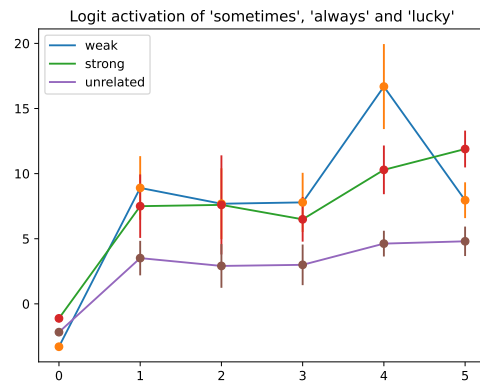


Figure 1: Activation of *sometimes*, *always* and *lucky* (an unrelated adjective) across various prompts at the different points in the prompt.

of implicature computation given a negation: although the activation of the weak item (the one present in the first sentence) is higher at point [4] (before the negation), the insertion of a negation leads to a drop of the activation of the weak adjective and a rise of the activation of the strong adjective at point [5]. The magnitude of these effects is such that the activation of the strong adjective after negation is higher than that of a weak adjective and the model does not copy the adjective that occurred in the prompt. The standard deviation bars on the plot show that in this case the effect can be reliably observed over individual prompts. We will call this behaviour of the model *crossing*. Note that *crossing* does not guarantee that the strong adjective will appear as the most likely token after negation, it only guarantees the non-copying behaviour of the model.

In the other case, illustrated in Fig. 2, both the effect of decreasing the activation of the weak scalar item and the effect of increasing the activation of the respective strong scalar item is observed, so the approaching trend of the two activations does not reach the level at which we could observe a reflection of this trend in the next word prediction behaviour: the weak item remains the most likely continuation and the model exhibits the copying behaviour. We will call this scenario *approaching*.

The last scenario illustrated in Fig. 3 includes the already observed effect of decreasing the activation of the weak scalar item but the activation of the strong scalar item also drops slightly. As a result, the difference in the activations decreases but similar to the *approaching* case there is no surface evidence of this trend. We will call this scenario

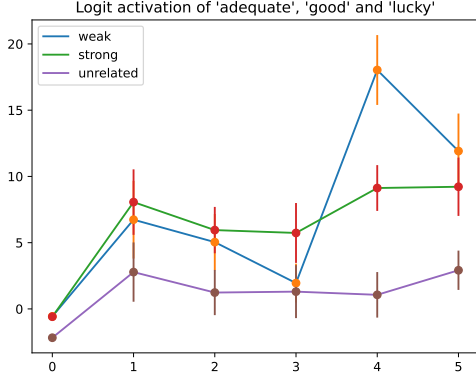


Figure 2: Activation of *adequate*, *good* and *lucky* (an unrelated adjective) across various prompts at the different points in the prompt.

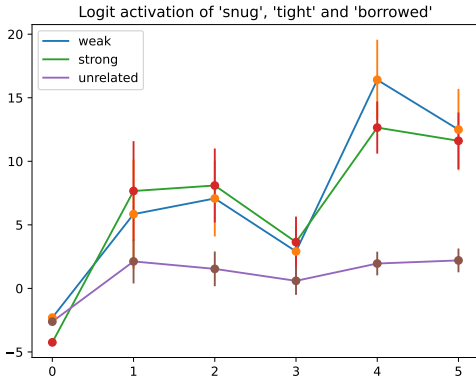


Figure 3: Activation of *snug*, *tight* and *borrowed* (an unrelated adjective) across various prompts at the different points in the prompt.

difference lowering.

These results show that introducing a negation in the second sentence does influence underlying activation of the language model but in most cases does not lead to a visible change of the output. In the next section, we introduce a negation-independent metric for adjective pairs. We hypothesize that the probability of the two adjectives to be simultaneously encountered among the top candidates for the next token in a positive scenario (corresponding to close and high activations of both adjectives at point [1]) correlates with the probability of *crossing* behaviour of the model if it is prompted following the schema (5).

2.3 Extracting new adjective pairs from the language model

We propose a method to evaluate the quality of scales from the literature, using language model

behavior.

Corpus Our corpus-based experiments are performed on the training data of the BabyLM 2023 challenge (Warstadt et al., 2023). This data consists of mostly transcribed and child-directed speech from different sources. We preprocess the data using the Boot-BERT pipeline (Samuel, 2023).

Identifying Matches The method starts with an unlabeled tokenized text corpus \mathcal{C} , a next word prediction language model \mathcal{M} and a collection $\mathcal{A} = \{a_1 \dots a_m\}$ of m (adjective) terms. \mathcal{M} provides, for each token $s_{i,j}$ in each sentence $s_i \in \mathcal{C}$, a probability distribution $p_{\mathcal{M}}(s_{i,j+1}|s_{i,:j})$ over all possible next tokens $s_{i,j+1}$ given a prefix $s_{i,:j}$. Specifically, we collect all $k = 10$ most likely continuations for every prefix in the corpus, and filter the corpus for prefixes $s_{i,j}$ where a scalar term is among the k most likely next tokens. We call these situations *matches*. Matches are independent from whether that scalar term is actually present in \mathcal{C} as a continuation.

Co-occurrence counts from matches Assume that $\text{count}(a)$ is the number of times that the adjective a is matched across the corpus. Further assume that $\text{cc}(a_r, a_s)$ is the number of times that two adjectives a_r and a_s are matched at the same prefix $s_{i,j}$ in the corpus. To account for the fact that the adjectives occur with different frequencies, we compute the following scores:

The scaled cooccurrence score, conditioned on one of the terms:

$$\text{cc}_{\log}(a_r, a_s) = \frac{\log \text{cc}(a_r, a_s)}{\log \text{count}(a_r)}$$

By this we obtain two cooccurrence scores, conditioned either on the weak or on the strong scalar item. We call the following score *scale by strong*:

$$\text{cc}_{\log}(a_{\text{weak}}, a_{\text{strong}}) = \frac{\log \text{cc}(a_{\text{weak}}, a_{\text{strong}})}{\log \text{count}(a_{\text{weak}})}$$

And the following score *scale by weak*:

$$\text{cc}_{\log}(a_{\text{strong}}, a_{\text{weak}}) = \frac{\log \text{cc}(a_{\text{strong}}, a_{\text{weak}})}{\log \text{count}(a_{\text{strong}})}$$

Thus we collect two scores that can be used either separately or combined. One way to combine them and make the resulting metric symmetric is calculating the harmonic mean of these scaled cooccurrence scores:

$$\text{cc-hm}(a_r, a_s) = 2 \frac{\text{cc}_{\log}(a_r, a_s) * \text{cc}_{\log}(a_s, a_r)}{\text{cc}_{\log}(a_r, a_s) + \text{cc}_{\log}(a_s, a_r)}$$

The harmonic mean prioritizes pairs of adjectives such that each of them is likely to be found in the top predictions of the model when the other one is in the top predictions. We can now sort all cooccurring adjective pairs a_r, a_s by their $\text{cc-hm}(a_r, a_s)$, and put special focus on the pairs with very high cooccurrence scores.

Finding new scalar adjective pairs From the pairs obtained on the previous step we have manually selected those that are scalar alternatives and identified the weaker and the stronger scale mates between them (see Table 5 in the Appendix). We have also attempted an automatic filtering of non-scalar pairs and automatic strength evaluation following the proposal by [de Melo and Bansal \(2013\)](#).

Among the versions we attempted were (1) rank extraction from the language model for the patterns suggested by [de Melo and Bansal \(2013\)](#) when the first adjective of the pattern is in the ground truth as well as (2) corpus search in the corpus used for the other experiments as well as (3) Google n-gram inquiry.

Neither method brought results reliable enough to justify automatic scale and strength extraction from the proposed list. Since for human experiments items have to be often evaluated according to additional criteria, our proposal at the moment is to supply a list of pairs with their model scores and leave it to the linguists to select the suitable pairs.

2.4 Experiment 2

In this experiment we have tested the behaviour of the language model on the adjective pairs from [Sun et al. 2024](#) together with some extra scales from [Lacina and Gotzner 2024](#) and on the scales extracted in the previous section using the harmonic mean score (all the scales are provided in the Appendix). For a fair evaluation, we have used the same corpus (BabyLM challenge, [Warstadt et al. 2023](#)) for all the scales. We have identified all the sentences where the weak adjective from either list occurs in the corpus and applied the pattern shown in (5) to all such sentences.

For each example we have computed the logit of the weak and the strong adjectives before and after negation (analogous to the points [4] and [5] in the first experiment). We have then computed

the proportion of cases when the activation of the weak item decreases after the negation is introduced (matched term lowering), the proportion of cases where the activation of the strong item after negation exceeds the activation of the weak item (crossing) and the proportion of cases where the two activations approach each other but the activation of the weak item remains higher than that of the strong item (approaching). The results are presented in Table 1.

	Sun et al	New pairs
Matched term lowering	1.00	0.99
Crossing	0.31	0.33
Difference lowering	0.99	0.99
Approaching	0.22	0.15

Table 1: Comparison of model behaviour for scales from human experiments and scales extracted on the basis of the language model data.

As can be seen in Table 1, exchanging the scalar pairs in the experiment led to an increase of the crossing instances, but this increase remained small. Another interesting observation is related to the approaching scenario: The number of approaching instances reduces when the scale selection is performed according to the model cooccurrence scores. This can be interpreted as that, loosely said, stronger related items tend to increase and decrease their activation together, which is not a desirable trend in the current setup. At the same time we observe an approaching behaviour in almost all (99%) of the cases with either selection of the items.

This means that if the trend behind the negation processing could be magnified, in principle it would be possible to achieve a desired (non-copying) behaviour under negation in almost all the cases. It is left for future research to explore such possibilities.

3 Model metrics and human behaviour

In order to evaluate the obtained metrics on human data, we extract all the values as described above for the scalar adjectival pairs that were used in human experiments ([Gotzner et al., 2018a,b](#); [Lacina and Gotzner, 2024](#)). We then compute the correlations between our metrics and human data. This reveals that the most helpful metric is *scale by strong*: that of the high ranking of the weak item in those sentences where the strong item is ranked

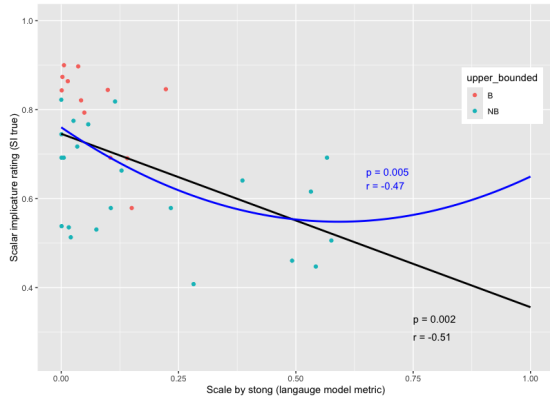


Figure 4: Correlation of *scale by strong* metric from the OPT language model and human scalar implicature ratings.

high (e.g. how often is *attractive* present in the top ten prediction of those sentences that have *stunning* in top ten predictions). This metric strongly negatively correlates with human scalar implicature rating ($-0, 51$) and positively correlates with human rating for negative strengthening ($0, 34$). It must be noted, however, that our results are limited by the range of scales for which human data is available: as described above, for the metrics we have calculated how often the two adjectives of the pair co-occur within the top ten predictions of the language model. We have discovered that within this metric, although values between zero and one are possible (see Appendix), the actual values for the original experimental items (see Figure 4) are very low (less than 0.6).

Figure 4 illustrates that both linear and quadratic correlations are plausible ($r = -0.51$ for linear and $r = -0.47$ for quadratic correlation) given the data from the previous experiments due to the limited range of values of the *scale by strong* metric. The extension of the two correlation curves demonstrates that obtaining the experimental results for items that lie on the right spectrum of the *scale by strong* metric is essential for making a decision about the validity of either correlation.

One parameter to take into account is the upper boundedness of the scale. It has been shown to be the main predictor for the human scalar implicature and negative strengthening ratings from a collection of linguistic features (Van Tiel et al. 2016 as well as Sun et al. 2018; another relevant but less strong predictor is semantic similarity). Although statistical evaluation of non-upper-bounded scales is not possible due to the insufficient amount of data, the two categories are shown

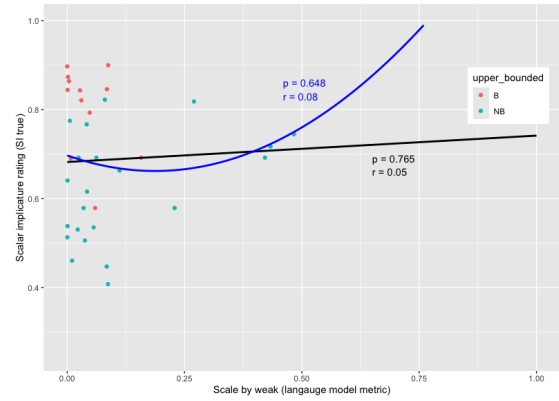


Figure 5: Correlation of *scale by weak* metric from the OPT language model and human scalar implicature ratings.

on Figure 5. It can be observed that all the human ratings of bounded scales are higher than these of non-bounded scales and at the same time almost all the language model scores for those scales are very low. The question whether this accidental or systematic can be studied by exploring the linguistic properties of the scales that receive high scores by the language model metrics.

In relation to this it is also worth exploring Fig. 5 that depicts the possible correlations of human scalar ratings with the model scores *scale by weak*. Although the statistical analysis produces very low correlation values ($r = 0.05$ for linear and $r = 0.08$ for quadratic correlation), the visual inspection reveals that most of the values for the model scores are below 0.12, so the correlation analysis can not be reliably performed on the basis of this data.

Since *scale by weak* score is very low for most of the scales, the harmonic mean score that takes into account both *scale by strong* and *scale by weak* also does not provide a significant correlation for the available set of data. Similarly to the case depicted on Fig. 5 the set of data does not exclude the possibility of discovering such a correlation given a different set of data.

Since the value of the proposed metrics are very low for most of the items found in the experimental literature, we suggest that more experiments should be performed with different scales that are better distributed according to those metrics. As described above, we propose a list of pairs that satisfies these criteria from the model perspective and leave it to the linguists to pick the best experimental items from it. This list is provided in the Appendix and the suggestions are marked in bold.

4 Discussion

In this paper we described several experiments related to scalar adjectives. First of all, we could establish that despite the overt copying behaviour of the analyzed language model the underlying activation exhibits desirable trends. Second, we have proposed model-based metrics to evaluate the scales and experimented with scales that receive high ratings according to these metrics. We could achieve a slight increase in desirable behaviour (crossing pattern between the activations of the weak and the strong items), although these results provided a lesser increase than we had expected.

We believe that the observed underlying behaviour of the language model while processing the negation opens new perspectives in adjusting the model predictions by magnifying the desired trends.

Finally, we have explored the connection between the human experimental results and proposed metrics. We could observe that one of the metrics (*scale by strong*) can be used to partially explain the variability in human ratings of the scalar implicatures associated with different scales. At the same time we could see that the scales used in human experiments have very low scores on all the proposed language model metrics. In this light we suggest new material for further experiments.

This type of work opens a field of automatic generation of possible experimental material as well as running the experiments using the language model before transferring them to the lab. This can lead to a significant decrease in time needed for experimental design as well as to lowering the cost of running various versions of the same experiment since some relevant design problems can be already observed and corrected on the level of the language model experiments.

References

- David Arps and Yulia Zinova. 2024. [It is difficult, but not impossible: Measuring scalar activation in language models](#). In *Proceedings of the 28th Workshop on the Semantics and Pragmatics of Dialogue - Poster Abstracts*, Trento, Italy. SEMDIAL.
- Gerard de Melo and Mohit Bansal. 2013. [Good, great, excellent: Global inference of semantic intensities](#). *Transactions of the Association for Computational Linguistics*, 1:279–290.
- Judith Degen. 2015. Investigating the distribution of some (but not all) implicatures using corpora and web-based methods. *Semantics and Pragmatics*, 8:11–1.
- Nicole Gotzner and Jacopo Romoli. 2022. [Meaning and alternatives](#). *Annual Review of Linguistics*, 8(Volume 8, 2022):213–234.
- Nicole Gotzner, Stephanie Solt, and Anton Benz. 2018a. Adjectival scales and three types of implicature. In *Semantics and Linguistic Theory*, pages 409–432.
- Nicole Gotzner, Stephanie Solt, and Anton Benz. 2018b. Scalar diversity, negative strengthening, and adjectival semantics. *Frontiers in psychology*, 9:1659.
- Julia Bell Hirschberg. 1985. *A theory of scalar implicature (natural languages, pragmatics, inference)*. Ph.D. thesis, University of Pennsylvania.
- Laurence Horn. 1984. Towards a new taxonomy for pragmatic inference: Q-and r-based implicature. *Meaning, form and use in context*.
- Md Mosharaf Hossain, Venelin Kovatchev, Pranoy Dutta, Tiffany Kao, Elizabeth Wei, and Eduardo Blanco. 2020. [An analysis of natural language inference benchmarks through the lens of negation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9106–9118, Online. Association for Computational Linguistics.
- Jennifer Hu, Sammy Floyd, Olessia Jouravlev, Evelina Fedorenko, and Edward Gibson. 2023. [A fine-grained comparison of pragmatic language understanding in humans and language models](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4194–4213, Toronto, Canada. Association for Computational Linguistics.
- Paloma Jeretic, Alex Warstadt, Suvrat Bhooshan, and Adina Williams. 2020. [Are natural language inference models IMPPRESsive? Learning IMPLICature and PRESupposition](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8690–8705, Online. Association for Computational Linguistics.
- Nora Kassner and Hinrich Schütze. 2020. [Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.
- Radim Lacina and Nicole Gotzner. 2024. Exploring scalar diversity through priming: A lexical decision study with adjectives. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 46.
- Benjamin Lipkin, Lionel Wong, Gabriel Grand, and Joshua B Tenenbaum. 2023. [Evaluating statistical language models as pragmatic reasoners](#). *Preprint*, arXiv:2305.01020.

- Rashid Nizamani, Sebastian Schuster, and Vera Demberg. 2024. [SIGA: A naturalistic NLI dataset of English scalar implicatures with gradable adjectives](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14784–14795, Torino, Italia. ELRA and ICCL.
- Eszter Ronai and Ming Xiang. 2022. Three factors in explaining scalar diversity. In *Proceedings of Sinn und Bedeutung*, volume 26, pages 716–733.
- Eszter Ronai and Ming Xiang. 2023. Tracking the activation of scalar alternatives with semantic priming. *Experiments in Linguistic Meaning*, 2:229–240.
- David Samuel. 2023. [Mean BERTs make erratic language teachers: the effectiveness of latent bootstrapping in low-resource settings](#). In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 221–237, Singapore. Association for Computational Linguistics.
- Michael Sullivan. 2024. [It is not true that transformers are inductive learners: Probing NLI models with external negation](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1924–1945, St. Julian’s, Malta. Association for Computational Linguistics.
- Chao Sun, Ye Tian, and Richard Breheny. 2018. A link between local enrichment and scalar diversity. *Frontiers in Psychology*, 9:2092.
- Chao Sun, Ye Tian, and Richard Breheny. 2024. A corpus-based examination of scalar diversity. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 50(5):808.
- Thinh Hung Truong, Timothy Baldwin, Karin Verspoor, and Trevor Cohn. 2023. [Language models are not naysayers: An analysis of language models on negation benchmarks](#). *Preprint*, arXiv:2306.08189.
- Bob Van Tiel, Emiel Van Miltenburg, Natalia Zevakhina, and Bart Geurts. 2016. Scalar diversity. *Journal of semantics*, 33(1):137–175.
- Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. [Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora](#). In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.
- Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu
- Wang, and Luke Zettlemoyer. 2022. [Opt: Open pre-trained transformer language models](#). *Preprint*, arXiv:2205.01068.
- Yuhui Zhang, Michihiro Yasunaga, Zhengping Zhou, Jeff Z. HaoChen, James Zou, Percy Liang, and Serena Yeung. 2023. [Beyond positive scaling: How negation impacts scaling trends of language models](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7479–7498, Toronto, Canada. Association for Computational Linguistics.

Appendix: established scale used for the experiments and proposed scales

	weak	strong
0	adequate	good
1	allowed	obligatory
2	attractive	stunning
3	big	enormous
4	cheap	free
5	dark	black
6	difficult	impossible
7	few	none
8	funny	hilarious
9	hard	unsolvable
10	hungry	starving
11	intelligent	brilliant
12	low	depleted
13	memorable	unforgettable
14	old	ancient
15	possible	certain
16	rare	extinct
17	scarce	unavailable
18	silly	ridiculous
19	small	tiny
20	snug	tight
21	some	all
22	special	unique
23	tired	exhausted
24	ugly	hideous
25	warm	hot
26	wary	scared

Table 2: Scales from (Sun et al., 2024) that we used in our experiments

	weak	strong
0	angry	annoyed
1	bad	mediocre
2	good	excellent
3	overweight	obese
4	pretty	beautiful
5	warm	hot

Table 3: Additional scales from (Lacina and Gotzner, 2024)

	weak	strong
0	afraid	scared
1	amazing	incredible
2	angry	mad
3	bad	terrible
4	big	huge
5	calm	quiet
6	clear	obvious
7	courageous	fearless
8	damaged	destroyed
9	difficult	hard
10	frightening	terrifying
11	good	great
12	great	awesome
13	great	good
14	honest	frank
15	odd	strange
16	overweight	obese
17	pleased	proud
18	popular	famous
19	pretty	beautiful
20	silly	stupid
21	small	tiny
22	smart	intelligent
23	some	all
24	surprised	shocked
25	tasty	delicious
26	useless	worthless
27	warm	hot
28	wealthy	rich

Table 4: Scales annotated by the authors which have a high cc-hm score.

	weak	strong	<i>scale by weak</i>	<i>scale by strong</i>		weak	strong	<i>scale by weak</i>	<i>scale by strong</i>		weak	strong	<i>scale by weak</i>	<i>scale by strong</i>
0	red	purple	0.78	0.98	50	new	recent	0.68	0.85	100	interesting	remarkable	0.53	0.77
1	second	third	0.90	0.98	51	unconscious	dead	0.85	0.57	101	full	crowded	0.49	0.77
2	better	worse	0.82	0.97	52	pale	white	0.85	0.63	102	lost	all	0.76	0.59
3	yellow	white	0.97	0.81	53	difficult	dangerous	0.81	0.85	103	necessary	vital	0.57	0.76
4	pink	red	0.96	0.79	54	full	empty	0.72	0.85	104	best	better	0.70	0.76
5	purple	black	0.96	0.73	55	difficult	impossible	0.77	0.85	105	long	difficult	0.64	0.76
6	good	great	0.92	0.96	56	several	all	0.85	0.67	106	understandable	acceptable	0.75	0.54
7	pink	white	0.95	0.75	57	important	critical	0.65	0.85	107	violent	cruel	0.65	0.75
8	orange	red	0.95	0.77	58	good	complete	0.61	0.85	108	full	all	0.75	0.60
9	interesting	amusing	0.61	0.95	59	damaged	broken	0.84	0.73	109	good	superior	0.46	0.75
10	important	essential	0.70	0.95	60	similar	identical	0.67	0.84	110	authorized	required	0.75	0.50
11	yellow	black	0.94	0.79	61	good	high	0.74	0.84	111	nervous	afraid	0.75	0.66
12	bulky	heavy	0.94	0.49	62	dark	black	0.84	0.81	112	damaged	lost	0.75	0.59
13	different	separate	0.74	0.94	63	oval	round	0.84	0.52	113	light	white	0.74	0.75
14	interesting	exciting	0.78	0.94	64	good	large	0.71	0.84	114	comfortable	luxurious	0.44	0.75
15	gray	white	0.94	0.64	65	polite	friendly	0.83	0.71	115	deep	loud	0.69	0.74
16	brown	black	0.94	0.78	66	sad	angry	0.82	0.83	116	emotional	moral	0.71	0.74
17	brown	white	0.94	0.78	67	concerned	alarmed	0.54	0.83	117	unusual	unique	0.74	0.62
18	gray	black	0.94	0.63	68	possible	probable	0.47	0.83	118	serious	dangerous	0.71	0.74
19	happy	proud	0.87	0.93	69	interesting	unusual	0.62	0.83	119	sick	dead	0.74	0.61
20	brown	blue	0.93	0.83	70	accurate	true	0.83	0.67	120	cool	cold	0.72	0.74
21	red	black	0.93	0.89	71	useful	valuable	0.79	0.83	121	certain	all	0.74	0.56
22	blue	black	0.92	0.86	72	steep	high	0.83	0.55	122	tender	soft	0.74	0.68
23	concerned	worried	0.90	0.92	73	good	perfect	0.65	0.83	123	acceptable	necessary	0.73	0.60
24	green	white	0.92	0.82	74	violent	dangerous	0.82	0.73	124	cheap	free	0.73	0.57
25	tired	exhausted	0.69	0.92	75	new	better	0.74	0.82	125	personal	private	0.72	0.73
26	kind	generous	0.60	0.92	76	mediocre	poor	0.82	0.43	126	smooth	shiny	0.64	0.73
27	overweight	obese	0.85	0.92	77	allowed	required	0.75	0.82	127	worried	frightened	0.55	0.73
28	tired	sleepy	0.57	0.91	78	serious	fatal	0.60	0.82	128	possible	likely	0.69	0.73
29	bruised	broken	0.91	0.59	79	interesting	attractive	0.69	0.82	129	knowing	caring	0.61	0.73
30	interesting	important	0.91	0.85	80	amused	pleased	0.81	0.64	130	dangerous	fatal	0.57	0.73
31	tedious	difficult	0.91	0.51	81	special	unique	0.74	0.81	131	good	excellent	0.53	0.73
32	damaged	destroyed	0.90	0.79	82	useless	impossible	0.81	0.69	132	third	half	0.73	0.72
33	good	better	0.83	0.90	83	interesting	beautiful	0.81	0.75	133	bent	broken	0.72	0.50
34	green	black	0.90	0.80	84	uncommon	rare	0.80	0.58	134	reasonable	high	0.72	0.52
35	attractive	beautiful	0.89	0.70	85	great	perfect	0.66	0.80	135	different	inferior	0.42	0.72
36	moist	wet	0.89	0.69	86	pleased	surprised	0.80	0.76	136	black	all	0.72	0.57
37	important	vital	0.58	0.89	87	severe	fatal	0.67	0.80	137	interesting	true	0.71	0.69
38	some	all	0.89	0.86	88	equal	identical	0.71	0.80	138	unpleasant	dangerous	0.71	0.47
39	relieved	happy	0.89	0.54	89	bad	evil	0.67	0.80	139	active	aggressive	0.62	0.71
40	harmful	dangerous	0.89	0.69	90	simple	obvious	0.76	0.80	140	surprised	frightened	0.52	0.71
41	good	cheap	0.58	0.88	91	near	close	0.76	0.80	141	general	any	0.71	0.57
42	hot	boiling	0.63	0.88	92	important	necessary	0.68	0.79	142	white	all	0.71	0.56
43	good	new	0.87	0.86	93	free	all	0.79	0.62	143	bright	warm	0.70	0.71
44	warm	hot	0.87	0.80	94	ready	willing	0.70	0.79	144	artistic	scientific	0.70	0.59
45	important	obvious	0.73	0.87	95	interesting	valuable	0.64	0.78	145	smooth	glossy	0.38	0.70
46	old	ancient	0.73	0.86	96	great	certain	0.67	0.78	146	some	black	0.58	0.70
47	sunny	warm	0.86	0.69	97	willing	eager	0.56	0.78	147	possible	any	0.70	0.61
48	distinct	separate	0.86	0.77	98	aggressive	violent	0.77	0.73	148	easy	pleasant	0.51	0.70
49	impractical	impossible	0.86	0.45	99	neglected	abandoned	0.77	0.60					

Table 5: All adjective pairs obtained from the ngram-based filtering in Sec. 2.3. Candidates for scale are marked in boldface.

Generative FrameNet: Scalable and Adaptive Frames for Interpretable Knowledge Storage and Retrieval for LLMs Powered by LLMs

Harish Tayyar Madabushi¹, Taylor Pellegrin², and Claire Bonial³

¹ University of Bath, UK

² Oak Ridge Associated Universities, USA

³ DEVCOM US Army Research Laboratory, USA

htm43@bath.ac.uk, taylor.a.pellegrin.ctr@army.mil, claire.n.bonial.civ@army.mil

Abstract

Frame semantics provides an explanation for how we make use of conceptual *frames*, which encapsulate background knowledge and associations, to more completely understand the meanings of words within a context. Unfortunately, FrameNet, the only widely available implementation of frame semantics, is limited in both scale and coverage. Therefore, we introduce a novel mechanism for generating task-specific frames using large language models (LLMs), which we call *Generative FrameNet*. We demonstrate its effectiveness on a task that is highly relevant in the current landscape of LLMs: the interpretable storage and retrieval of factual information. Specifically, Generative Frames enable the extension of Retrieval-Augmented Generation (RAG), providing an interpretable framework for reducing inaccuracies in LLMs. We conduct experiments to demonstrate the effectiveness of this method both in terms of retrieval effectiveness as well as the relevance of the automatically generated frames and frame relations. Expert analysis shows that Generative Frames capture a more suitable level of semantic specificity than the frames from FrameNet. Thus, Generative Frames capture a notion of frame semantics that is closer to Fillmore’s originally intended definition, and offer potential for providing data-driven insights into Frame Semantics theory. Our results also show that this novel mechanism of Frame Semantic-based interpretable retrieval improves RAG for question answering with LLMs—outperforming a GPT-4 based baseline by up to 8 points. We provide open access to our data, including prompts and Generative FrameNet.¹

1 Introduction, Motivation and Context

Frame semantics (Fillmore et al., 2006) is a linguistic theory that emphasizes understanding word

meanings through the semantic and conceptual “frames” or “schemas” within which they operate. This theory is exemplified by FrameNet, a manually curated dataset of frames designed to represent commonly occurring concepts (Baker et al., 1998; Ruppenhofer, 2006).² Although FrameNet has been touted for its utility in improving tasks such as textual entailment, it has also been criticized for its limited coverage and for being too abstract to effectively support many downstream applications (e.g., Burchardt et al. (2009)). In this work, we propose a novel mechanism for generating domain-specific frames at the appropriate level of abstraction for a given downstream task. We refer to this approach and the resultant frames as *Generative FrameNet*.

We focus on the downstream task of retrieving relevant facts to answer specific questions. We demonstrate a method for generating more contextually relevant frames that retain their utility in the evolving landscape of LLMs, which have the inherent tendency to generate plausible sounding, yet inaccurate output. This phenomenon, referred to as “hallucinations,” has been a significant stumbling block in broad deployment of LLMs in applications requiring accuracy (Ji et al., 2023). Hallucinations themselves are not limited to factual inaccuracies, and include other modes of failure.

The capabilities of LLMs typically improve with an increase in their “size,” which is a combination of a model’s parameters and the size of the pre-training corpus. Until recently, this was seen by some as being evidence that further scaling would eventually address the shortcomings of LLMs, including hallucinations. For example, LLMs were claimed to develop “emergent abilities”: specifically, it was believed that LLMs, when scaled to several billion parameters developed capabilities including those required to solve tasks involv-

¹<https://github.com/H-TayyarMadabushi/Generative-FrameNet>

²<https://framenet.icsi.berkeley.edu/>

ing reasoning in humans, thus indicative that the LLMs were developing reasoning skills (Wei et al., 2022b). More recent work, however, has shown that this is not the case and that LLMs instead develop a single capability, which they leverage to solve tasks (Lu et al., 2023). This capability, called “in-context learning,” is, roughly put, the ability of models to solve a particular task based on a few examples provided in the prompt (Brown et al., 2020; Chowdhery et al., 2023). Lu et al. (2023) further suggest that the process of instructional fine-tuning LLMs to understand instructions (Wei et al., 2022a), enables models to leverage the same “in-context” abilities even in the absence of examples. This finding indicates that further scaling, while providing improved instruction following abilities, will not grant models the broader capacity for general reasoning.

The fact that LLMs are not likely to develop the ability to reason has profound implications to work on improving them, including to mitigating hallucinations. It implies that we must explore alternative approaches. This is especially the case when it comes to factual hallucinations as the ‘parametric memory’ in LLMs is orders of magnitude smaller than the pre-training data (Ji et al., 2023). As such, they must necessarily use some method of compressing their pre-training data. Without the ability to distinguish between the information that is relevant and what is not relevant in their pre-training data, their method of compression defaults to be the memorisation of frequent information. Less frequently occurring facts are not explicitly stored and instead the model has access to only statistical approximations. Given that the exact information stored is not explicit and also different for models of different scale and training regimes, the only way to get around hallucination is to explicitly provide LLMs with all but the most common information.

The most effective method of providing such information, and therefore mitigating factual inaccuracies to date has been Retrieval Augmented Generation (RAG), which involves the inclusion of relevant information to the prompt (Lewis et al., 2020). However, RAG comes with its own shortcomings. The retrieval of information relevant to answering a query is not straightforward (Gao et al., 2024). While LLMs can handle some noise in the retrieved context provided, a dramatic increase in noise unsurprisingly leads to deteriorating performance of models. This problem becomes even more important when the query requires reasoning

over multiple facts, each of which are progressively semantically further from the query. Overall, because logically connected information is not always semantically similar, existing keyword and distributional similarity based search and information retrieval (IR) systems are poorly suited for the specific IR requirements of LLMs (Fleischer et al., 2024). Existing methods of dealing with this problem in IR are not interpretable, and the deep neural methods relying on embeddings introduce another opaque mechanism, making failures difficult to diagnose and fix. Given this context, this work makes the following contributions:

1. We propose a novel mechanism of generating relevant frames at the level of abstraction required for specific problems using LLMs that we call Generative FrameNet.
2. We show the effectiveness of these frames on the task of retrieving relevant information for answering questions that remains extremely relevant even in the context of LLMs.
3. We additionally demonstrate, through a manual expert evaluation, the quality and relevance of these frames, showing that our method has the potential to provide data-driven resources and insights for the theory of Frame Semantics.

The rest of this paper is organised as follows: §2 provides an overview of Frame Semantics, and §3 provides an overview of our use of Frame Semantics for retrieval. We then demonstrate the shortcomings of an existing Frame Semantic resource, FrameNet (§4), before detailing our methods of generating and using custom frames for indexing facts in §5. §6 present our results including the effectiveness of our methods in addition to a manual analysis of the frame resource we create, before concluding in §7.

2 Frame Semantics

Frame semantics (Fillmore et al., 2006) is a theory of linguistics that emphasises that the meanings of words are best understood by the semantic and conceptual “frames” or “schemas” within which they function. As Fillmore puts it, “words represent categorisations of experience, and each of these categories is underlain by a motivating situation occurring against a background of knowledge and experience” (Fillmore et al., 2006, 373-374). A frame is the cognitive structure or background

against which the meaning of a word is defined and understood. Frames organise knowledge based on typical situations, actions, or common experiences.

A frame influences how the meanings of words are interpreted in different contexts. This facilitates basic word sense distinctions such as river “bank” and financial “bank”, but also nuanced interpretations of words such as “guilty” in everyday or religious contexts as opposed to legal contexts. Additionally, when a word invokes a frame, it also invokes related concepts within that frame. For example, the word “sell” invokes a commercial transaction frame involving a seller, a buyer, an item being sold, and a price. Thus, the frame helps to predict and explain the use of other related words and the roles they play within the same context.

3 Frame Semantic Retrieval

This section provides an overview of Frame Semantic Retrieval, our proposed mechanism of storing and indexing factual information to aid effective retrieval.

3.1 Development & Evaluation Data

In evaluating our mechanism of retrieval, we make use of Entailment Bank (Dalvi et al., 2021), which comprises science questions from school years 4 to 6, along with relevant facts and “entailment trees”.

Question	How might eruptions affect plants?
Associated “Factoids”	F1: eruptions emit lava; F2: eruptions produce ash clouds; F3: plants have green leaves; F4: plant producers die without sunlight; F5: ash clouds block sunlight.
Inference Steps	F2 + F5 implies I1: eruptions block sunlight; F4 + I1 implies I2: eruptions can cause plants to die.
Answer	eruptions can cause plants to die.

Table 1: Example question from Entailment Bank and associated factoids. LLMs find it significantly easier to generate the required entailment trees when presented with all relevant facts, demonstrating the continued relevance of effective and interpretable IR.

Consider Table 1, which presents an example from the Entailment Bank dataset. The original task involves building an entailment tree—a tree consisting of inference steps—and consists of three sub-tasks at different levels of difficulty:

Task 1 presents the model with all relevant facts

and requires the construction of the entailment tree; **Task 2** requires the model to perform the same task, but with 15 to 25 distractor facts included;

Task 3 involves first extracting the relevant facts before constructing the entailment tree.

The authors find that even a relatively small model, T5-11B (Raffel et al., 2020), can perform relatively well on Tasks 1 and 2, when fine-tuned. Task 3, they find, is much harder, highlighting the importance of efficient retrieval (see Dalvi et al. (2021) for details).

3.2 Frame Semantics for Information Indexing and Retrieval

Overall, these results reinforce our earlier points: retrieval is non-trivial and improving retrieval has the potential to significantly boost model performance. In the example presented above, using search terms derived just from the question (e.g., “eruption”) including more complex combinations (e.g., “eruption and plants”) may not effectively retrieve relevant information. Additionally, if the search terms are too broad, it can cause the retrieval of a significant number of irrelevant facts. Both the lack of relevant facts and a large number of unrelated facts can hinder the model’s performance.

Fillmore’s Frame Semantics theory posits that when the question of Table 1 is presented to an English speaker, the question would evoke a volcanic eruption frame and a plant life frame, including the frame elements of those frames. We contend that frame structures facilitate capturing the level of specificity found in the associated factoids, i.e. frame elements such as “lava, ash, plants, sunlight,” and the level of specificity needed to reason about such questions. Triggering such frames activates the elements, priming speakers to reason about the question using the relevant concepts (e.g., Bodner and Masson (2003)). *Thus, this work is motivated by the hypothesis that we can significantly narrow the search space if we index facts—stored as plain text—according to the frames they invoke and use the frames associated with the question along with the relations between frames to retrieve relevant facts.* To test our hypothesis, we focus our experiments on the retrieval of relevant facts.

Importantly, using frames associated with questions and relevant factoids, along with frame relations, offers an inherently interpretable method of indexing and retrieval. This approach also has the added benefit of enabling easy updates to fast-changing information.

3.3 Task: Relevant Fact Retrieval

Our choice of the specific task is motivated by our earlier observation that LLMs can perform reasonably well at answering complex questions when provided with relevant facts alongside some distractors. However, as described in the previous section, the retrieval of these relevant facts poses a significant challenge. Therefore, we focus on the task of retrieving relevant factoids for answering questions in Entailment Bank. Specifically, we focus on the information extraction subtask required in Task 3 described in §3.1. Notice that the effective retrieval of facts would simplify Task 3 to Task 2, the task of building entailment trees given the relevant facts and some distractors. Given how effective T5-11B (which by current standards consists of relatively few parameters) is on Task 2, simplifying Task 3 to Task 2 provides a template for solving tasks based exclusively on retrieved facts, which would in turn help with the mitigation of factual hallucinations in LLMs. We slightly modify Task 3 by constructing the corpus of facts that we extract from using all the facts required by any question across the relevant data split, instead of the complete text book corpus which is harder to process. This limitation is not a significant drawback, as we can always add more facts if needed. We have chosen not to do so currently due to cost constraints, but this could be addressed in the future by leveraging open LLMs. Regardless, we evaluate Frame Semantic retrieval and the baselines on exactly the same set of questions and facts to ensure a fair comparison to past work (Dalvi et al., 2021). All experiments are run on the complete Entailment Bank test set consisting of 340 questions and 1,109 corresponding factoids.

3.4 Empirical Evaluation Metrics

Given the nature of our task, we select Recall@k as our evaluation metric. The average length of entailment trees in the Entailment Bank dataset is 7.6 with very few having more than 10. Given that Task 2 (described previously in §3.1) includes between 15 and 25 distractors, we test our methods using Recall@k for $k \in \{35, 40, 45\}$. Success in this setting will demonstrate that our retrieval mechanism can effectively simplify Task 3, which requires retrieval from the entire corpus, into the simpler Task 2, which involves building entailment trees based on relevant facts and a few distractors.

3.5 Baselines

We use two different baselines, against which we compare the effectiveness of Frame Semantic indexing and retrieval. We briefly test a third baseline using frames from FrameNet, but find it to be particularly ill-suited for this task (for a manual comparison of Generative Frames and FrameNet frames in this context, see §4.1). Consequently, we discontinue further exploration. The first baseline is a simple keyword match baseline and is chosen due to our emphasis on interpretability and ease of correction. Since Frame Semantic retrieval implicitly provides interpretability, we choose a baseline that is similarly transparent. We first generate search terms by feeding the relevant question to RAKE (Rose et al., 2010), a tool for effectively extracting search terms. We then perform a simple string match to extract all factoids that contain the keywords. The second baseline we use is not directly comparable as it is not interpretable. This consists of using an LLM to generate relevant search terms. Both baselines can be boosted using several techniques. However, we choose not to test these methods, as the purpose of this study is not to create a mechanism that outperforms existing methods, but to establish the feasibility of the Frame Semantic indexing and retrieval process which has the advantages of being interpretable and based on cognitive linguistic theory.

4 FrameNet

Prior to the introduction of Generative Frames, created using LLMs (§5), we explore the effectiveness of FrameNet, an existing online database based on Frame Semantics, for the task at hand. The goal of FrameNet is to catalogue English words and their associated semantic frames, defining the various roles and relations in a frame and illustrating these with example sentences. Each “frame” in FrameNet captures a specific type of event, relation, or entity and the roles associated with it.

FrameNet is the product of years of manual effort. Unfortunately, the 1200 frames of FrameNet remain limited to the domains of annotated data and do not have broad coverage of all the frames that a single speaker would build up over a lifetime of experience. Indeed, such a coverage goal is ludicrous given the time and expense of manually constructing FrameNet. This challenge motivated our data-driven generation of semantic frames, which we will describe in §5.1. Nonetheless, to clearly justify

our choice to leverage Generative Frames in lieu of the existing FrameNet, we evaluate both FrameNet and our generated frames for suitability as an external knowledge base in our RAG approach.

4.1 Manual Frame Evaluation

We randomly sampled a set of questions from Entailment Bank such that we had five non-overlapping samples totaling 29 lines (questions and factoids) each. These 29 lines included 5 questions and the related supporting factoids for that question; questions were included for coherency but only factoids were annotated, as having the relevant frames for each factoid should provide the relevant factoids for the question, as described in §3. Each sample was used for a manual annotation and evaluation task designed to examine the coverage of FrameNet as well as the semantic granularity of any relevant frames. We report full annotation procedures and details in Appendix A; here we briefly summarize the tasks and FrameNet results. These tasks were repeated for the same samples with our Generative Frames as well; the results for that evaluation are reported in Section 6.1.

The first task, presented to two annotators, evaluates and provides judgments on the semantic granularity of the frames assigned by an automatic FrameNet tagger (Chanin, 2023) to one Entailment Bank sample. The frames are assigned in order of the detection of triggers for that frame in that sentence. For example:

Entailment Bank Factoid: gases released during the use of fossil fuels causes global warming.

Tagged Frames: USING, CAUSATION

For each frame assigned, the annotators assign a value from 1-3, where 1 indicates that the frame is too general to be useful in capturing the most salient concepts of the instance, 2 is a useful level of specificity, and 3 is too specific to capture the salient concepts invoked by the instance.

The second task, presented to the same two annotators, asks each annotator to assign up to two FrameNet frames to another Entailment Bank sample. In addition, for each instance, the annotator responds to a question as to whether the potentially applicable frames are too similar, and therefore can't be distinguished as to which is a better fit, and a separate question as to whether the resource lacks adequate coverage for capturing the semantics of the instance.

4.2 FrameNet: Manual Evaluation Results

The first task exploring the granularity of the frames demonstrated that the vast majority of frames tagged were too general to be useful (17 out of 25 annotation instances had modal values of “1: too general” across the frames assigned to that instance). Consider the example above: The USING frame was found to be too general by both annotators while the CAUSATION frame was found to be too general by one but of a useful granularity by the other annotator. These frames are triggered by the lexical items “use” and “causes” respectively. As the matrix verb, “cause” is certainly more central to an understanding of the factoid, but both are general concepts that can be applied to a range of sentences from many different conceptual domains.

The second task exploring the coverage of FrameNet demonstrated that it lacks coverage for the semantic domains of the Entailment Bank data; i.e. the natural world. Both annotators found that FrameNet lacked sufficient frame coverage for about 80% of the 24 factoid instances in the sample. Our Inter-Annotator Agreement (IAA) calculations for both tasks are presented in Appendix A.3.

Overall, our manual evaluation of FrameNet shows that, despite the immense value in the carefully curated resource, there are still broad swaths of domains such as the natural world that lack suitable coverage in FrameNet. Although frames can be triggered by and assigned to our data, these frames are too general to effectively capture the semantics of the domain in order to support reasoning and answering questions about it. This motivates the data-driven, semi-automatic development of a novel Frame Semantic resource, described next.

5 Frame Semantic Generation: Methods and Qualitative Analysis

In this section, we detail the methods used for frame generation, Frame Semantic indexing, and retrieval. Table 2 exemplifies all stages of the methodology. Given that one objective is to maintain interpretability and to potentially provide data-driven insights to the theory of Frame Semantics, we perform a qualitative analysis of the outputs of each of the stages. An empirical evaluation of the effectiveness of these methods is presented in §6.2.

The mechanism of retrieving information based on Frame Semantics consists of three distinct tasks: frame identification, duplicate testing, and frame relation identification. The first step of the frame

Task	Prompt	Output Example
<p>Frame Identification</p> <p>During pre-processing, facts are indexed by the the frames they invoke</p> <p>During inference, relevant facts are extracted based on frames invoked by the question and additional frames that are related</p>	<p>What is the single/two most important frame, based on the theory of Frame Semantics, relevant for answering the question/fact below. Do not include frames about answering questions or reasoning, that is implied. Do not include frames which are metaphorical. Ensure the the name of the frame is as descriptive as possible. Output a single frame and join words in the frame by underscores. Output nothing but the name of the frame.</p> <p>Question 1: How does the appearance of a constellation change during the night? Answer 1: celestial_motion ... Problem: Question Problem: <QUESTION> Answer Problem:</p>	<p>Input Question: Tides, such as those along the coast of Massachusetts, are caused by gravitational attractions acting on Earth. Why is the gravitational attraction of the Moon a greater factor in determining tides than the gravitational attraction of the much larger Sun? Output Frame: GRAVITATIONAL_INFLUENCE</p>
<p>Check if the new frame must be added to the frame set</p> <p>Used during inference</p>	<p>The following question has been tagged with the single frame listed. Is this frame significantly different from existing frames listed and should it be added as a new frame? Respond with True if it is significantly different otherwise False. Respond with True and False only.</p> <p>Example Question: From Earth, the Sun appears brighter than any other star because the Sun is the Example Tagged Frame:proximity Example Existing Frames 2: CELESTIAL_MOTION Example Answer: True ... Question Problem: <INPUT QUESTION> Tagged Frame Problem: <INPUT NEW FRAME> Existing Frames Problem: <INPUT EXISTING FRAME> Answer Problem:</p>	<p>Input Question: Melinda learned that days in some seasons have more daylight hours than in other seasons. Which season receives the most hours of sunlight in the Northern Hemisphere? Input Frame Assigned: SEASONAL_VARIATION_IN_DAYLIGHT Input List of Existing Frames: DAYLIGHT_VARIATION, SEASONAL_ADAPTATION, SEASONAL_BEHAVIOR, SEASONAL_CHANGE, SEASONAL_VARIATION Output (Add SEASONAL_VARIATION_IN_DAYLIGHT to Frame Set?): False Action Taken: Question tagged with DAYLIGHT_VARIATION</p>
<p>Identifying Frame Relations</p> <p>Used during inference</p>	<p>Listed below is a single frame relevant to a question. List those frames which are most likely to be associated with the facts required to answer this question. These frames are based on the theory of Frame Semantics. Do not include frames about answering questions or reasoning, that is implied. Do not include frames which are metaphorical. [...]</p> <p>Example Question 1: Stars are organized into patterns called constellations. One constellation is named Leo. Which statement best explains why Leo appears in different areas of the sky throughout the year? Example Question Frame: CELESTIAL_MOTION Example Output Frames: CONSTELLATION_CLASSIFICATION, STAR_CLASSIFICATION, CELESTIAL_MOTION Problem Question : <QUESTION> Problem Question Frame : <FRAME> Problem Output Frames:</p>	<p>Input Question: Which measurement is best expressed in light-years? Input Question Frame: DISTANCE_IN_ASTRONOMY Output set of Frames Related to Question Frame: CELESTIAL_DISTANCE, ASTRONOMICAL_UNIT, SPATIAL_MEASUREMENT</p>

Table 2: Prompts and associated outputs for each step in frame based indexing and retrieval. Terms enclosed in <brackets> represent placeholders and ... represent up to 5 similar in-context examples that are substituted with the actual examples, question or frame during inference. See text (Section 5) for detailed description of each of the steps.

identification task is a pre-processing step, which involves creating relevant frames where required, and indexing all relevant factoids based on between two and four of the most prominent frames that they invoke (See Row 1 of Table 2). After pre-processing, at inference time, the single most important frame associated with the question (the question frame, also depicted in Row 1 of Table 2) is identified.

The second task is to check for duplicate frames: in order to ensure that newly generated frames are not too similar to existing frames, we perform a duplication test, also using GPT-4, depicted in Row 2 of Table 2. This duplication test involves retrieving the five most semantically similar frames (using SentenceBERT based vector similarity) from the previously generated set and prompting GPT-4 to either (a) determine if the new frame should be added or (b) decide if one of the existing frames is sufficient, selecting the most appropriate one.

The third task is to identify frame relations. We identify frames associated with the question frame, which are likely to be associated with factoids relevant to answering the original question, but separated by one or more logical steps (frame relations, depicted in Row 3 of Table 2). We conduct the duplication test for the related frames as well before introducing them to our Generative Frames.

The complete prompts are made openly available on our project site. In all cases, we prompt GPT-4 (OpenAI et al., 2024) using a temperature of 0 to ensure reproducible results. Overall, this method allows us to use an LLM to generate candidate frames and to match these with previously generated frames, thus allowing us to build a frame based index of factoids that we use for retrieval through similarly generated frames associated with questions during inference.

This method supports two key functions: (a) generating frames associated with a given text, and (b) identifying frame relations at a single level of separation. While further traversal through additional levels of frame relations is technically possible, we opted against this due to the potential for noise. Future work will focus on developing a more structured and hierarchical frame architecture, which could allow traversal beyond a single step while maintaining precision. In the next two sections, we provide greater details on the frame and frame relation identification steps.

5.1 Frame Identification

There are two difficulties in identifying the frames associated with facts or questions. The first is the necessity to define a complete set of frames, and the second is the linking of these frames to the relevant fact or question. In addition to our manual evaluation of FrameNet (§4.1), we conducted exploratory experiments using FrameNet as a definitive source of all frames, which we used to compare against facts and questions from Entailment Bank; we showed that FrameNet is inadequate for our research purpose for two reasons. First, FrameNet’s focus on ‘trigger’ words to identify frames is problematic. This emphasis on individual trigger words, likely influenced by the tools available at the time of FrameNet’s inception, overlooks the fact that a sentence, as a whole, might invoke a frame that is difficult to identify through trigger words alone, which themselves can be challenging to extract within sentences. Second, as mentioned in our manual evaluation findings, the frames available within FrameNet cover a limited set of domains, which overlap minimally with the frames that are appropriate for the Entailment Bank dataset.

To address these issues, we bootstrap the creation of frames using an LLM, specifically GPT-4. We prompt GPT-4 to generate frames relevant to the input fact or question, allowing us to organically expand our set of frames. We use in-context examples, selected from the training set, to enable the model to better output relevant frames. This process involves initially prompting the model to generate frames without in-context examples for facts and questions in the training set one at a time. From these outputs, we identify outputs deemed relevant and of sufficient quality and use them as in-context examples to refine the model’s performance. These in-context examples are made available alongside the data released with this work.

We start with an empty ‘frame set’ and iteratively generate frames associated with facts and questions. For each fact or question, the frames output by GPT-4 are compared with the existing frames previously generated (or none in the initial instances). This duplication test is also done with the help of GPT-4. We first extract 5 frames, whose frame names are most semantically similar to that of the newly generated frame. This is done using Sentence BERT (Reimers and Gurevych, 2019), an effective semantic similarity metric that originally relied on BERT (Devlin et al., 2019), but

now makes use of custom contextual embeddings. We then prompt GPT-4 to determine if the newly generated frame must be added to the frame set.

As an example, GPT-4, when prompted to generate frames related to the Entailment Bank factoid “the gravitational pull of the sun on earth’s oceans causes the tides,” might generate GRAVITATIONAL INFLUENCE and TIDAL MOVEMENT. These frames are compared against the existing frames and the frame GRAVITATIONAL INFLUENCE might be replaced by the similar frame GRAVITATIONAL ATTRACTION already in our frame set. If a similar frame is not found, the original frame is added to the frame set. This same process is then used to generate frames associated with questions. We find that GPT-4 is a poor judge of identifying frames which are truly different from those already in the frame set. Thus, we always augment the original set of frames with five existing frames whose names are most semantically similar to the original. See also Table 2 for more examples.

5.2 Frame Relations

We call the overlap between the frames invoked by a question and those invoked by the facts necessary for answering that question a first-order overlap. This first-order overlap isn’t sufficient for extracting all facts relevant to answering a question. As such, we require a means of identifying relations between frames, so we can expand the set of relevant frames, as a proxy for the reasoning process.

Instead of importing definitions of frame relations, for example from FrameNet, we generate these relations using a data-driven approach. Specifically, we extract questions and associated facts from the training data. We then assign frames to both the questions and the facts using the methods described previously. The frames associated with the questions and the corresponding facts are assumed to hold a latent relation, which we use to generate similar frame relations at the time of answering questions. This is done by prompting GPT-4 with the relevant question and the frame associated with the frame and requiring GPT-4 to generate frames relevant to answering the question. While these relations are currently simplistic, we believe that iteratively refining them with input from linguists can make them more nuanced. Row three of Table 2 presents the prompt and an example output of this step.

6 Results

6.1 Qualitative Analysis & Evaluation

A qualitative analysis of resultant frames and frame relations demonstrates the effectiveness of this method. Table 2 presents some of the frames and frame relations automatically generated using the methods described above. The results are far from perfect, but are interesting from the perspectives of the diversity and adaptability they present. We note that these results are achieved through prompting alone. Given that LLMs, such as GPT-4, are unlikely to be designed to solve tasks such as this, it is not surprising that there is much room for improvement, although the results demonstrate the feasibility of this method. To robustly evaluate the quality of the frames and compare to FrameNet, we conduct the same two manual evaluation tasks described in §4.1, except this time we use our set of 941 Generative Frames resulting from the data-driven process described in §5.

The first evaluation task examines the semantic granularity of the frame assigned to a factoid in the same Entailment Bank sample evaluated for FrameNet (see §4.1 and Appendix A.1). Annotators supply a 1-3 value judgment on each Generative Frame automatically assigned in the process of our pipeline, where 1 indicates that a frame is too general, 2 indicates that a frame is of a useful granularity for reasoning about the question, and 3 is too specific. When using our Generative Frames, the majority were found to be of a useful granularity for capturing the semantics of the factoid (15 of 24 annotation instances, 63%, had modal values of “2: useful” across the frames assigned to that instance). In comparison to FrameNet, for which 0 frames were thought to be too specific, 2 of the instances received modal values of “3: too specific”. Only one instance had a modal value of “1: too general”, although 5 instances were tied for modal values of 1 or 2.

The second task evaluates the coverage of the frame resource (Appendix A.2). The same two annotators were tasked with assigning up to two Generative Frames to the same sample previously evaluated for FrameNet (§4.1). Additionally, the annotators responded to one question as to whether the potentially applicable frames are too similar, and one question as to whether the resource lacks adequate coverage. Given that our frames were generated to capture the Entailment Bank data, it is unsurprising that the two annotators agreed that

the resource had adequate coverage for 100% of the instances in the sample. Although our Generative Frames lack a search or annotation interface parallel to what was used during FrameNet annotation (instead annotators were simply presented with a long text list of all Generative Frames along with definitions and frame elements), the annotators agreed upon at least one of the assigned frames in 83% of 24 instances. This agreement is much higher than for FrameNet, which was 63%. This demonstrates that while it can be difficult to agree upon a triggered frame when the frames are very general (as in FrameNet), annotators tend to agree upon the triggered frame when it is of a more precise granularity in capturing the semantics of the factoid.

Overall, our evaluation shows that the Generative Frames have high coverage of our domain, and that coverage involves frames that are of a useful granularity for capturing the salient semantics of factoids, facilitating reasoning about the questions to which those factoids relate.

6.2 Empirical Evaluation

Recall@	RAKE Search (Baseline 1)	GPT-4 Search (Baseline 2)	Frame Semantic Retrieval (our method)
@35	0.330	0.385	0.439
@40	0.333	0.390	0.464
@45	0.338	0.396	0.473

Table 3: Recall@k between 35 and 45 comparing Frame Semantic retrieval to search based retrieval where the search terms are generated using a traditional keyword based method (RAKE) and using GPT-4. It is notable that Frame Semantic retrieval performs significantly better than both baselines across all selected values of k .

We present an empirical evaluation of the Frame Semantic retrieval methods described above. We compare the performance of Frame Semantic retrieval to the two search-based baselines described in Section 3.5. We present the results in Table 3. Overall, we find that Frame Semantic retrieval outperforms both the simple search-based baseline, as well as the baseline where search terms are generated using GPT-4, by a significant margin. Recall that we test our methods using Recall@k for $k \in 35, 40, 45$ to take into account the fact that this allows us to demonstrate that our retrieval mechanism can effectively simplify Task 3, which requires retrieval from the entire corpus, into Task 2, which involves building entailment trees based

on relevant facts and a few distractors. Our results show that we do effectively narrow down the search space and demonstrates the feasibility of frame-based indexing.

Frame semantic indexing and retrieval has significant advantages—each stage can be improved by fine-tuning LLMs for the specific purpose. Most importantly, the transparent nature of this process, which outputs frames at each stage, allows for the analysis and ‘debugging’ of each stage.

7 Conclusions and Future Work

This work presents a novel mechanism of generating relevant frames of the appropriate level of abstraction for any domain. We demonstrate the use of these frames in the challenging task of interpretable IR. Our qualitative manual evaluation and empirical evaluation demonstrate that our hypothesis, that we can effectively narrow the search space by indexing facts according to the frames they invoke along with related frames via frame relations, is supported. Thus, this work demonstrates the feasibility and effectiveness of this method in both retrieval and the automatic generation of frames which, when scaled to multiple tasks, also has the potential to provide data-driven insights to the theory of Frame Semantics.

In future work, we will create models that are fine-tuned for each of the tasks within this approach: frame generation, identification and frame relation identification. This approach is feasible, as the necessary training data can be bootstrapped using in-context examples and manual quality checks. We will also extend this work to multiple tasks. We emphasise that this work also provides a template for effectively integrating cognitive linguistics and LLM research, benefiting both fields.

Limitations

Our experiments are based on a single task in a specific domain. As a proof of concept of a novel method that is based on cognitive linguistic theory, these experiments are effective in showcasing the feasibility of this method. However, demonstrating the effectiveness of this method on multiple tasks is required for a more rigorous test, which we leave to future work. Additionally, our experiments, however, do not extend to testing LLMs for reduced hallucinations; prior work implies that improved retrieval will indeed lead to reduced hallucinations, but it is left to future work to rigorously test this.

References

- Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. [The Berkeley FrameNet project](#). In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada. Association for Computational Linguistics.
- Glen E Bodner and Michael EJ Masson. 2003. Beyond spreading activation: An influence of relatedness proportion on masked semantic priming. *Psychonomic Bulletin & Review*, 10(3):645–652.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Aljoscha Burchardt and Marco Pennacchiotti. 2008. Fate: a framenet-annotated corpus for textual entailment. In *LREC*.
- Aljoscha Burchardt, Marco Pennacchiotti, Stefan Thater, and Manfred Pinkal. 2009. Assessing the impact of frame semantics on textual entailment. *Natural Language Engineering*, 15(4):527–550.
- David Chanin. 2023. [Open-source frame semantic parsing](#). *Preprint*, arXiv:2303.12788.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayanan Pilla, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. [Palm: Scaling language modeling with pathways](#). *Journal of Machine Learning Research*, 24(240):1–113.
- Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021. [Explaining answers with entailment trees](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7358–7370, Online and Punta Cana, Dominican Republic. 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.
- Charles J Fillmore et al. 2006. Frame semantics. *Cognitive linguistics: Basic readings*, 34:373–400.
- Daniel Fleischer, Moshe Berchansky, Moshe Wasserblat, and Peter Izsak. 2024. [Rag foundry: A framework for enhancing llms for retrieval augmented generation](#). *Preprint*, arXiv:2408.02545.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. [Retrieval-augmented generation for large language models: A survey](#). *Preprint*, arXiv:2312.10997.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. [Survey of hallucination in natural language generation](#). *ACM Comput. Surv.*, 55(12).
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.
- Sheng Lu, Irina Bigoulaeva, Rachneet Sachdeva, Harish Tayyar Madabushi, and Iryna Gurevych. 2023. [Are emergent abilities in large language models just in-context learning?](#) *Preprint*, arXiv:2309.01809.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke

Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rameev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya,

Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. [Gpt-4 technical report](#). *Preprint*, arXiv:2303.08774.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, pages 1–20.

Josef Ruppenhofer. 2006. Framenet ii: Extended theory and practice. <http://framenet.icsi.berkeley.edu/>.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022a. [Finetuned language models are zero-shot learners](#). In *International Conference on Learning Representations*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022b. [Emergent abilities of large language models](#). *Transactions on Machine Learning Research*. Survey Certification.

A Manual Annotation & Evaluation Details

A.1 Evaluation Task 1: Granularity of the Frame Resource

Our first task explores the semantic granularity of the frames in FrameNet with respect to the Entailment Bank subject matter, which largely relates to phenomena of the natural world. We leverage the automatic FrameNet tagger of [Chanin \(2023\)](#) to assign FrameNet frames to each sentence of our sample; resulting in 1-4 frames assigned to

each sentence. The frames are assigned in order of the detection of triggers for that frame in that sentence.

Example 1:

Entailment Bank Factoid: gases released during the use of fossil fuels causes global warming
Assigned Frames: USING, CAUSATION

The sentences and annotations were presented to two linguist annotators who are native English speakers trained in linguistics and semantic role annotation schemas. In a spreadsheet, each annotator provided a judgement on each frame assigned. The judgement options were numerical values 1-3 corresponding to:

1=Frames are about a high-level concept and not helpful in summarising the question or factoid beyond what kind of factoid/question it is

2=Frames are about the topic and helpful in summarising the question/factoid

3=Frames are too specific to the topic at hand and provide very little in way of generalisation

NA=The frame assigned is not applicable at all; i.e. a tagger error

For the first task, we report the modal judgment value (e.g., 1, 2, or 3) across all frames tagged for that question/factoid. This gives us a broad sense of the granularity of the frames despite the fact that different instances have different numbers of frames tagged. We also measure Inter-Annotator Agreement (IAA) by computing simple agreement in the form of the percentage of frame judgments agreed upon across the two annotators.

A.2 Evaluation Task 2: Coverage of the Frame Resource

The second task explores the coverage of the Entailment Bank domain and the ability of the two annotators to assign appropriate frames to the second sample of questions and factoids. Each annotator is presented with each line of the Entailment Bank sample in a spreadsheet, and asked to leverage the online FrameNet search to find and assign up to 2 relevant frames. For each annotation instance, the annotator is asked to respond “yes,” or “no”:

Q1: The 2 frames assigned are too similar; I cannot tell which is more appropriate

Q2: The resource lacks coverage for capturing this question/factoid

For the first question, annotators could also respond “NA” if only one frame was determined to be applicable. For the second question, even if one frame was determined to be triggered by the question or factoid, the annotator could respond “yes - the resource lacks coverage...” if the frame was applicable but so general that it was not useful in capturing the semantic domain invoked by the instance.

For the second task, we report the percentage of “yes” and “no” answers to each question across all annotation instances. We report this for each annotator with the expectation that the percentages should be similar. We also report IAA of the frame assignment task in the form of the percentage of agreed upon assigned frames out of the total number of instances. Each annotation instance can be counted as a single match if either of the up to two assigned frames matched.

A.3 IAA Results

FrameNet, Annotation Task 1 Our Inter-Annotator Agreement (IAA) analysis found that the annotators agreed upon the same value for an assigned frame in 39 out of 48 frames (again, 1-4 frames can be assigned per instance), for an agreement percentage of about 81%. Thus, although the task is subjective, annotators tend to agree on the values assigned.

FrameNet, Annotation Task 2 The IAA analysis of the second task finds that annotators agreed upon the frame assigned (or that no applicable frame existed) in 63% of the 24 instances. Since annotators responded that 80% of the instances lacked frame coverage that succinctly captured the factoid, it is reasonable that IAA would be somewhat low for this task.³ The disagreements involved related frames; for example:

Example 2

Entailment Bank Factoid: Color is a kind of property

Annotator 1 Frame: COLOR

³Our IAA is lower than the relatively high frame agreement reported in Burchardt and Pennacchiotti (2008) of 88%, where FrameNet frames were assigned to text instances in support of a textual entailment task. Their frame assignment was limited to frames evoked by certain lexical triggers assigned in a previous step, so it is a simpler task with much more limited choices.

Generative Frames, Annotation Task 1 In our IAA analysis of the first task, we find that annotators agreed on the value judgement of the automatically assigned Generative Frame in 65% of the 71 total frames assigned (instances could be assigned up to 4 frames). This IAA is slightly lower than that of the FrameNet evaluation, likely because all of the FrameNet frames were very general, whereas the Generative Frames have a greater range from too general to too specific.

Generative Frames, Annotation Task 2 Given that our frames were generated to capture the Entailment Bank data, it is unsurprising that the two annotators agreed that the resource had adequate coverage for 100% of the instances in the sample.

However, the annotators did not agree upon the extent to which the applicable frames were too similar. One annotator only found 2 applicable frames for 3 of the 24 instances (for all others only one frame was assigned), and answered that the 2 frames were sufficiently distinguishable in all 3 of those cases. The other annotator found 2 applicable frames for 13 of the 24 instances, and answered that in all 13 cases, the 2 frames were too similar to distinguish. This reinforces the notion that the frames are of a finer semantic granularity in comparison to FrameNet, but also demonstrates that the annotators may have approached this task differently. While FrameNet has a nice search interface for its frames, we currently have no such tool for the Generative frames. Thus, one annotator may have taken an approach of searching through our spreadsheet listing Generative Frames until a well-fitting frame was found and then stopping, while the other may have searched more broadly to find multiple frames.

The annotators agreed upon at least one of the assigned frames in 83% of the 24 instances. This agreement is much higher than the equivalent for FrameNet, which was at 63%.

Author Index

- Al-Saeedi, Ali, 86
Arps, David, 97
- Bai, Yu, 18, 61
Bonial, Claire, 107
- Cai, Fang, 18
Chi, Haifeng, 61
- Das, Writwick, 51
Devereux, Barry, 40
Dong, Tiansi, 51
Dori, Dov, 71
- Farnan, Greg, 40
- Guo, Rui, 40
- Harma, Aki, 86
He, Lirong, 6
He, Shizhu, 31
Hudson, Taylor, 107
- Jang, Jinwoo, 1
Jia, Tao, 6
- Kang, Xin, 71
- Lee, Kangil, 1
Lim, Youngjin, 1
liu, baoqiang, 18
Liu, Kang, 31
Liu, Ming, 6
Liu, Xiang, 61
- McLaughlin, Niall, 40
- Qin, Yongbin, 6
- Romoli, Jacopo, 97
- Sheng, Yongpan, 6
Shin, Minsu, 1
Shteyngardt, Veronika, 71
Sifa, Rafet, 51
Spalek, Katharina, 97
Sun, Jinze, 6
- Sun, Wangtao, 31
- Tayyar Madabushi, Harish, 107
- Wang, Lianji, 61
Wang, Yuhan, 71
- Xue, Shuang, 18
- Ye, Na, 18
- Zhang, Guiping, 18, 61
Zhao, Jun, 31
Zinova, Yulia, 97