A Closer Look at In-Context Learning for Temporal Knowledge Graph Forecasting

Anonymous ACL submission

Abstract

While temporal knowledge graph forecasting (TKGF) approaches have traditionally relied heavily on complex graph neural network architectures, recent advances in large language models, specifically in-context learning (ICL), have presented promising out-of-the-box alternatives. While previous works have shown the potential of using ICL, its limitations and generalization capabilities for TKGF are underexplored. In this study, we conduct a comparative analysis of complexity (*e.g.*, number of hops) and sparsity (*e.g.*, relation frequency) confounders between ICL and supervised models using two annotated TKGF benchmarks. Our experimental results showcase that while ICL performs on par or outperforms supervised models in lower complexity scenarios, its effectiveness diminishes in more complex settings (*e.g.*, multi-step, more number of hops, etc.), where supervised models are superior.

1 Introduction

Knowledge graphs (KGs) are commonly used structures that store relational information as a graph (Bollacker et al., 2008; Vrandečić and Krötzsch, 2014). While using KGs for keeping static facts is common, they are unsuitable for holding complex dynamic (*i.e.,* temporal) information. Temporal knowledge graphs (TKGs) are extensions of KGs that enable the storage of such information (Leetaru and Schrodt, 2013; García-Durán et al., 2018). Consequently, TKGs allow practitioners to do various predictive tasks on complex temporal data. One critical task that has been empowered by TKGs is temporal knowledge graph forecasting (TKGF) (Gastinger et al., 2023), where the objective is to predict future facts from a set of prior facts before a specific time in a TKG. A hypothetical real-world example of TKGF is to answer the question, *"Who is USA going to Meet in June 2025?"* based on previous political events. This scenario can be represented by the query quadruple $q = (USA, Meets, ?, June\ 2025)$ and the time-constrained TKG $\mathcal{G}_t = \{(USA, Meets, UK, June\ 2024), (USA, Attends, G7, June\ 2025), (USA, Meets, Germany, June\ 2025), \dots\}$.

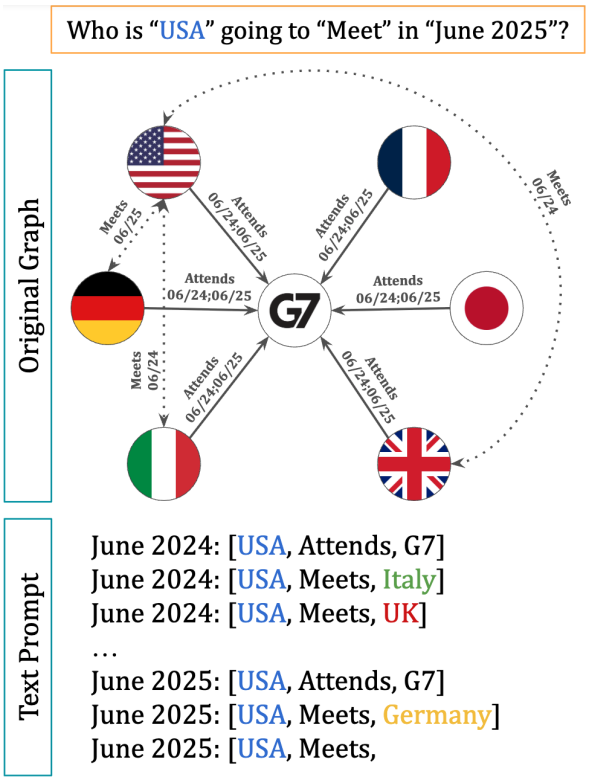


Figure 1: **Example of Graph to Text Prompt Conversion for ICL.** The given task is to predict which country is gonna meet the USA during the G7 summit, based on the previous interactions between the countries.

Recent studies have demonstrated large language models' (LLMs) effectiveness as general estimators across various function classes (Garg et al., 2022; Mirchandani et al., 2023). Consequently, these advancements have sparked interest in employing LLMs for temporal knowledge graph forecasting (TKGF). Specifically, LLMs have shown remarkable potential for TKGF, surpassing state-of-the-art supervised models in some scenarios using

Temporal Rule	Complexity			Sparsity	
	# Unique Entities	# Unique Relations	# Hops	Relation Frequency	Time Interval
$(E_1, \textit{express intent to meet}^{-1}, E_2, T_1) \Rightarrow (E_1, \underbrace{\textit{share information}}_{R}, E_2, T_2)$	2	2	1	f_R	$T_2 - T_1$
$(E_1, \textit{provide military aid}, E_2, T_1) \wedge (E_2, \textit{intend to protect}^{-1}, E_3, T_2)$ $\Rightarrow (E_1, \underbrace{\textit{provide military aid}}_{R}, E_3, T_3)$	3	2	2	f_R	$T_3 - T_1$
$(E_1, \textit{riot}, E_2, T_1) \wedge (E_2, \textit{make statement}, E_1, T_2) \wedge (E_1, \textit{riot}, E_2, T_3)$ $\Rightarrow (E_1, \underbrace{\textit{demonstrate or rally}}_{R}, E_2, T_4)$	2	3	3	f_R	$T_4 - T_1$

Table 1: **Confounder values examples.** The samples are taken from Liu et al. (2022) with some small modifications. Note that f_R refers to the frequency of relation R among all quadruples in the dataset.

in-context-learning (ICL) (Lee et al., 2023) (see Figure 1 for an example). Methods such as ICL present a cheap, fast, and ready-to-use alternative to traditional methods, many of which use computationally heavy graph neural network (GNN) architectures. However, despite all their benefits, the broad applications of such solutions for forecasting problems and LLMs' "grey box" nature (*e.g.*, opaque reasoning process, unpredictability across different temporal patterns) raise concerns regarding their limitations and generalizability.

In this study, we provide insights into the effect of various confounders – arising from relational and temporal patterns – on the effectiveness of ICL for TKGF. To this end, first, we utilize a state-of-the-art rule-based model to generate reasoning rules from well-known TKG benchmarks, ICEWS14 and ICEWS18 (García-Durán et al., 2018). Then, based on the generated rules, we create two labeled datasets containing confounder annotations for the test sets. Finally, we use these datasets to compare ICL-based models to state-of-the-art supervised models in single-step and multi-step settings (Gastinger et al., 2023) across complexity (*e.g.,* number of unique entities), and sparsity (*e.g.,* relation frequency) confounders (see Table 1 for more thorough examples). Our experimental results on the annotated datasets show that (1) ICL-based models outperform supervised models in scenarios with lower complexity, such as annotated samples with 1-hop patterns in single-step settings or samples involving only one unique relation, and (2) increasing the complexity of the patterns results in ICL-based models to underperform massively compared to the supervised models. This phenomenon is particularly evident in multi-step settings, where ICL-based models lag behind supervised models in all scenarios.

2 Background and Related Work

Formal Definition of TKGF. Formally, a TKG $\mathcal{G} = (\mathcal{Q}, \mathcal{E}, \mathcal{R}, \mathcal{T})$ comprises a set of quadruples $\mathcal{Q}$ in the form (s, r, o, t), where s and o are entities within $\mathcal{E}$, r is a relation within $\mathcal{R}$, and t is a timestamp from $\mathcal{T}$. The TKG forecasting task aims to predict a missing entity in future quadruples, either as $(s, r, ?, t)$ for tail prediction or $(?, r, o, t)$ for head prediction, using historical data from the graph. This process involves scoring all entities so that the true entity receives the highest ranking.

Supervised Models. Recent supervised models primarily utilize embedding-based GNNs to enhance their structural and sequential learning capabilities. Specifically, they have used autoregressive architectures to aggregate information both globally and locally in RE-Net (Jin et al., 2020), combined convolutional and recurrent architectures for modeling temporal sequences in RE-GCN (Li et al., 2021), introduced neural ordinary differential equations to model temporal sequences in TANGO (Han et al., 2021), and extended convolutional architectures to learn evolutionary patterns in CEN (Li et al., 2022). In parallel to these models, other approaches have been introduced in prior works, such as using a copy-mechanism in CyGNet (Zhu et al., 2021), leveraging reinforcement learning on temporal paths in TiTer (Sun et al., 2021), and learning temporal logic rules via temporal random walks in TLogic (Liu et al., 2022).

ICL-based Models. Recent advances in LLMs have drastically improved their capabilities, leading to emergent behaviors such as ICL. ICL allows LLMs to perform tasks conditioned solely on

Dataset	$\|\mathcal{E}\|$	$\|\mathcal{R}\|$	# of Facts		Time Granularity
			Train/Valid/Test	Annotated	
ICEWS14	6,869	230	75k/8.5k/7.3k	11,625	1 day
ICEWS18	23,033	256	373k/46k/50k	65,003	1 day

Table 2: **Dataset statistics.** Each dataset consists of historical facts divided into three subsets based on time.

the provided context without parameter optimization. Utilizing ICL, Lee et al. (2023) introduced the first LLM-based TKGF model, which showed performance on par with state-of-the-art supervised models without any training. Moreover, Xia et al. (2024) introduced an improved historical fact retriever and an alignment training procedure, posting better performances than the state-of-the-art supervised models[1]. While these ICL-based models have shown interesting achievements toward the TKGF task, we still lack a proper understanding of their limitations, a gap we aim to bridge.

3 Experimental Setup

3.1 Datasets

Our experiments focused on two prominent TKGF datasets: ICEWS14 (García-Durán et al., 2018) and ICEWS18 (Jin et al., 2020) (see Table 2). We specifically chose these datasets because 1) they are commonly used by almost all the prior works in the literature and 2) they pose a much more significant challenge to the forecasting models compared to other existing datasets such as WIKI (Leblay and Chekol, 2018) and YAGO (Rebele et al., 2016). Moreover, to keep our results consistent and comparable to previous works, we use the same splits as Gastinger et al. (2023).

3.2 Weak Supervision

One of the challenges we faced in our experiments was the absence of annotations for different confounders in the existing datasets. To overcome this issue, we employ weak supervision (Voskarides et al., 2018; Zhang et al., 2024; Tong et al., 2024) using TLogic (Liu et al., 2022), a state-of-the-art rule-learning-based TKG model, to annotate test samples with temporal multi-hop patterns. To this end, first, we ran the rule-learning part of TLogic on the combination of all quadruples from the train, valid, and test sets with the number of hops $\in \{1, 2, 3\}$. Then, we annotated each test sample using the matching pattern with the highest score[2], if such a rule existed. Finally, for the annotated test quadruples, we extract various confounders from their associated patterns, including the number of unique entities and relations, the pattern's length denoted as "hop", the relation frequency of the test query, and the time interval (see Table 2 for annotation statistics).

3.3 Models

For our ICL-based baseline, we utilize the model as described by Lee et al. (2023), which employs `gpt-neox-20b` (Black et al., 2022). This method is an inference-time approach that demonstrates performance comparable to supervised models. Moreover, for the TKG baselines we used state-of-the-art models with the hyperparameters and implementation as provided by (Gastinger et al., 2023): RE-Net (Jin et al., 2020), RE-GCN (Li et al., 2021), TANGO (Han et al., 2021), CyGNet (Zhu et al., 2021), and CEN (Li et al., 2022).

3.4 Implementation Details

We retain the top 100 entities with the highest scores (or the highest log probability) to evaluate each prediction. This protocol is done due to a limitation of the ICL-based model preventing it from predicting entities that do not appear in its context, which at most contains 100 historical facts, bounded by the context length of the underlying model (*i.e.,* `gpt-neox-20b`). Moreover, this protocol allows us to evaluate and fairly compare the ICL-based and supervised models across our experiments. As for our metrics, we report the Hits@{1,3} based on the list of retained entities for each prediction. All baseline models report both head and tail prediction performance by generating a head query $(?, r, o, t)$ and a tail query $(s, r, ?, t)$ for each test quadruple (s, r, o, t), following standard practices in the literature. We report the average head and tail prediction performances. The codebase uses PyTorch (Paszke et al., 2019) and Huggingface (Wolf et al., 2020) libraries.

4 Experiments

Table 3 presents our experimental results on both single-step (top) and multi-step (bottom) queries, grouped by the *number of hops* as the confounder. We can observe that the ICL-based models only

[1]The implementation has not been made public.

[2]For each matched reasoning path, TLogic combines rule confidence and temporal recency scores into one score.

Single-step	Train	ICEWS14						ICEWS18					
		H@1			H@3			H@1			H@3		
		1-hop	2-hop	3-hop	1-hop	2-hop	3-hop	1-hop	2-hop	3-hop	1-hop	2-hop	3-hop
RE-GCN	✓	42.6	15.2	38.7	63.6	32.2	56.1	34.5	19.5	31.9	54.7	35.5	50.2
TANGO	✓	36.4	12.0	36.2	54.5	24.8	50.2	29.7	16.3	28.3	48.8	31.0	45.5
CEN	✓	43.3	15.2	39.0	63.2	30.0	56.2	33.9	18.9	31.1	54.0	34.3	49.1
Average		40.8	14.1	38.0	60.4	29.0	54.2	32.7	18.3	30.4	52.5	33.6	48.3
Median		42.6	15.2	38.7	63.2	30.0	56.1	33.9	18.9	31.1	54.0	34.3	49.1
gpt-neox-20b-entity	✗	46.4	10.6	37.7	65.8	17.8	54.0	29.8	11.2	27.9	48.1	22.4	43.6
Δ Average		5.6	-3.5	-0.2	5.4	-11.2	-0.2	-2.9	-7.1	-2.5	-4.4	-11.2	-4.6
Δ Median		3.7	-4.6	-0.1	2.7	-12.2	-2.1	-4.1	-7.8	-3.2	-5.9	-11.9	-5.5
gpt-neox-20b-pair	✗	43.6	6.8	37.9	58.3	9.8	51.4	31.0	10.9	30.1	48.7	17.9	47.1
Δ Average		2.9	-7.4	-0.1	-2.2	-19.2	-2.7	-1.7	-7.3	-0.3	-3.8	-15.7	-1.2
Δ Median		1.0	-8.5	-0.8	-4.9	-20.2	-4.6	-2.9	-8.0	-0.4	-5.3	-16.4	-2.0

Multi-step	Train	ICEWS14						ICEWS18					
		H@1			H@3			H@1			H@3		
		1-hop	2-hop	3-hop	1-hop	2-hop	3-hop	1-hop	2-hop	3-hop	1-hop	2-hop	3-hop
RE-NET	✓	37.3	13.3	36.0	54.1	25.9	51.3	28.8	16.0	27.8	48.0	31.4	45.0
RE-GCN	✓	36.6	15.7	34.9	55.4	30.0	49.0	29.5	18.2	28.9	48.3	33.0	45.8
CyGNet	✓	35.5	11.9	34.5	53.6	26.0	49.9	25.5	13.4	26.1	44.9	28.3	44.1
Average		36.4	13.6	35.1	54.3	27.3	50.1	27.9	15.9	27.6	47.1	30.9	45.0
Median		36.6	13.3	34.9	54.1	26.0	49.9	28.8	16.0	27.8	48.0	31.4	45.0
gpt-neox-20b-entity	✗	34.3	8.7	32.1	49.6	16.9	44.6	19.7	8.9	19.7	31.3	17.8	30.7
Δ Average		-2.1	-4.9	-3.0	-4.8	-10.4	-5.4	-8.2	-7.0	-7.9	-15.8	-13.1	-14.2
Δ Median		-2.3	-4.6	-2.8	-4.5	-9.1	-5.3	-9.0	-7.2	-8.1	-16.7	-13.6	-14.3
gpt-neox-20b-pair	✗	30.9	6.5	32.6	43.7	8.9	43.4	23.7	8.7	25.6	37.9	14.4	38.5
Δ Average		-5.5	-7.1	-2.5	-10.6	-18.3	-6.7	-4.2	-7.2	-2.0	-9.2	-16.4	-6.5
Δ Median		-5.6	-6.8	-2.3	-10.4	-17.0	-6.5	-5.0	-7.3	-2.2	-10.2	-17.0	-6.6

Table 3: **Performance (Hits@K)** comparison between supervised models and ICL for single-step (top) and multi-step (bottom) prediction, grouped by the **number of hops** as the confounder. The first group consists of supervised models, whereas the second group consists of ICL models, *i.e.,* GPT-NeoX. The green and red colors represent where ICL is outperforming and underperforming the average performance of the supervised models.

perform better with 1-hop queries in the ICEWS14 dataset. Moreover, as the number of hops, an indicator of the pattern complexity, increases, supervised models outperform ICL-based models. Interestingly, this decline in performance is not monotonic in terms of complexity, making it even more challenging to predict the potential pitfalls. For example, LLMs' worst performance in ICEWS14 occurs in 2-hop queries, while the performance on 3-hop queries stays competitive. Moreover, we observe the same trend when analyzing other confounders related to pattern complexity. For example, ICL-based models outperform the supervised models in patterns involving two unique entities on ICEWS14. However, as the *number of unique entities* increases, the performance of ICL-based models declines (see Table 4 in Appendix A). Similarly, this trend is evident when the samples are grouped by *number of unique relations* (see Table 5 in Appendix A). When the samples are grouped by *relation frequency*, the ICL-based models perform on par or moderately outperform the supervised models only in the ICEWS14 single-step setting. In all other cases, the supervised models outperform the ICL-based models. However, the upward trend in Figure 2 (Appendix A) indicates that as relation frequency increases, the performance gap between the ICL-based and supervised models decreases. Moreover, when the samples are grouped by *time interval* (see Figure 3 in Appendix A), the supervised models consistently outperform the ICL-based models. We observe that ICL-based models perform worse in the multi-step setup across all confounders than their counterpart average supervised models. Finally, the performance gap is wider on the ICEWS18 (compared to ICEWS14), which could be attributed to it being more challenging.

5 Conclusion

In this paper, we presented an in-depth analysis of the effect of various confounders on the predictive power of ICL-based and supervised models for TKGF. Specifically, we created two annotated benchmarks for testing models across varied complexities and sparsity levels. Our experimental results indicate that while ICL is effective in low-complexity scenarios, its performance rapidly deteriorates as the complexity of the patterns increases. These findings highlight the need for more granular evaluation and testing of LLMs for TKGF.

Limitations

Our work is the first step toward a more granular evaluation of TKGF. As such, expanding the presented findings with more annotated datasets, identifying additional confounders, and evaluating a broader range of supervised and LLM-based models should be explored in future works. With the growing utilization of LLMs, comprehensive benchmarks allow us to make more grounded comparisons across models rather than being misled by potential spurious biases. Moreover, we observed fluctuations in performance gaps with increased complexities in different confounders. This phenomenon makes the performance comparison more uncertain, which should be further investigated in future works.

References

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, Usvsn Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 95–136, virtual+Dublin. Association for Computational Linguistics.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Alberto García-Durán, 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, Brussels, Belgium. Association for Computational Linguistics.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. 2022. What can transformers learn in-context? a case study of simple function classes. In *Advances in Neural Information Processing Systems*, volume 35, pages 30583–30598. Curran Associates, Inc.

Julia Gastinger, Timo Sztyler, Lokesh Sharma, Anett Schuelke, and Heiner Stuckenschmidt. 2023. Comparing apples and oranges? on the evaluation of methods for temporal knowledge graph forecasting. In *Machine Learning and Knowledge Discovery in Databases: Research Track*, pages 533–549, Cham. Springer Nature Switzerland.

Zhen Han, Zifeng Ding, Yunpu Ma, Yujia Gu, and Volker Tresp. 2021. Learning neural ordinary equations for forecasting future links on temporal knowledge graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8352–8364, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020. Recurrent event network: Autoregressive structure inferenceover temporal knowledge graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6669–6683, Online. Association for Computational Linguistics.

Julien Leblay and Melisachew Wudage Chekol. 2018. Deriving validity time in knowledge graph. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, page 1771–1776, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Dong-Ho Lee, Kian Ahrabian, Woojeong Jin, Fred Morstatter, and Jay Pujara. 2023. Temporal knowledge graph forecasting without knowledge using in-context learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 544–557, Singapore. Association for Computational Linguistics.

Kalev Leetaru and Philip A Schrodt. 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. 2022. Complex evolutional 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, Dublin, Ireland. Association for Computational Linguistics.

Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021. Temporal knowledge graph reasoning based on evolutional representation learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 408–417, New York, NY, USA. Association for Computing Machinery.

Yushan Liu, Yunpu Ma, Marcel Hildebrandt, Mitchell Joblin, and Volker Tresp. 2022. Tlogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(4):4120–4127.

Suvir Mirchandani, Fei Xia, Pete Florence, brian ichter, Danny Driess, Montserrat Gonzalez Arenas, Kanishka Rao, Dorsa Sadigh, and Andy Zeng. 2023. Large language models as general pattern machines. In *7th Annual Conference on Robot Learning*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Thomas Rebele, Fabian Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. 2016. Yago: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *The Semantic Web – ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part II*, page 177–185, Berlin, Heidelberg. Springer-Verlag.

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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yongqi Tong, Sizhe Wang, Dawei Li, Yifan Wang, Simeng Han, Zi Lin, Chengsong Huang, Jiaxin Huang, and Jingbo Shang. 2024. Optimizing language model's reasoning abilities with weak supervision. *arXiv preprint arXiv:2405.04086*.

Nikos Voskarides, Edgar Meij, Ridho Reinanda, Abhinav Khaitan, Miles Osborne, Giorgio Stefanoni, Prabhanjan Kambadur, and Maarten de Rijke. 2018. Weakly-supervised contextualization of knowledge graph facts. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 765–774, New York, NY, USA. Association for Computing Machinery.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yuwei Xia, Ding Wang, Qiang Liu, Liang Wang, Shu Wu, and Xiaoyu Zhang. 2024. Enhancing temporal knowledge graph forecasting with large language models via chain-of-history reasoning. *arXiv preprint arXiv:2402.14382*.

Tianyi Zhang, Linrong Cai, Jeffrey Li, Nicholas Roberts, Neel Guha, and Frederic Sala. 2024. Stronger than you think: Benchmarking weak supervision on realistic tasks. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

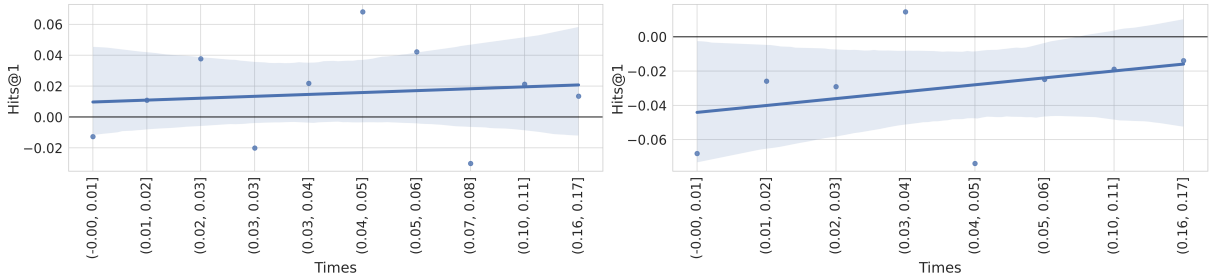
Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhang. 2021. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):4732–4740.

A Full Experimental Results

Single-step	Train	ICEWS14						ICEWS18					
		H@1			H@3			H@1			H@3		
		2	3	4	2	3	4	2	3	4	2	3	4
RE-GCN	✓	45.2	36.8	10.6	66.0	55.2	23.2	35.0	32.9	16.0	55.2	51.7	30.2
TANGO	✓	39.4	34.2	8.3	57.2	49.0	18.0	30.2	29.3	12.8	49.5	47.0	25.5
CEN	✓	46.1	36.9	9.8	65.8	54.6	23.0	34.4	32.3	14.7	54.6	50.9	28.3
Average		43.6	36.0	9.6	63.0	52.9	21.4	33.2	31.5	14.5	53.1	49.9	28.0
gpt-neox-20b-entity	✗	49.2	35.3	7.8	68.1	50.8	16.8	30.3	27.3	12.9	48.6	43.7	22.2
Δ Average		5.6	-0.7	-1.7	5.1	-2.1	-4.6	-2.9	-4.2	-1.6	-4.5	-6.1	-5.7
Δ Median		3.9	-1.5	-2.0	2.3	-3.8	-6.3	-4.1	-5.0	-1.8	-6.0	-7.1	-6.0
gpt-neox-20b-pair	✗	41.6	34.5	6.8	61.3	48.3	9.9	31.6	30.4	12.0	49.4	46.9	19.3
Δ Average		2.1	-1.4	-2.8	-1.6	-4.7	-11.5	-1.7	-1.1	-2.5	-3.8	-3.0	-8.7
Δ Median		0.4	-2.2	-3.1	-4.4	-6.3	-13.1	-2.9	-1.9	-2.7	-5.3	-4.0	-9.0

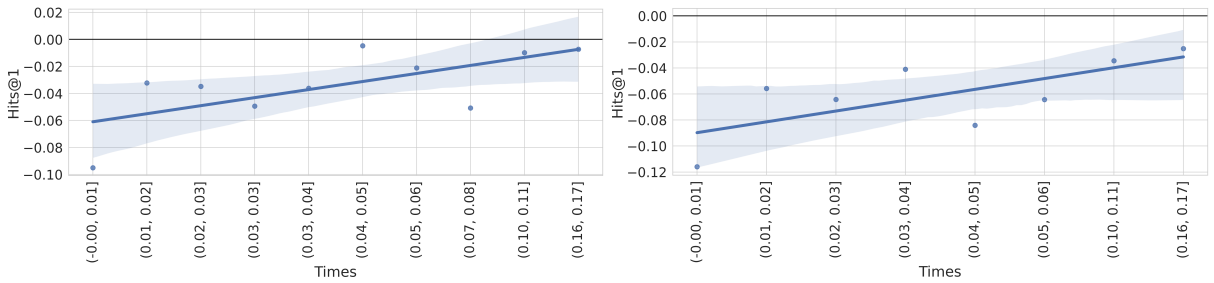
Multi-step	Train	ICEWS14						ICEWS18					
		H@1			H@3			H@1			H@3		
		2	3	4	2	3	4	2	3	4	2	3	4
RE-NET	✓	40.0	34.1	9.9	57.3	49.2	20.2	29.3	28.7	13.0	48.7	46.6	25.6
RE-GCN	✓	38.2	35.0	10.6	56.4	50.5	20.8	30.1	30.1	13.7	49.0	47.5	26.7
CyGNet	✓	37.9	33.3	8.3	56.2	49.6	17.0	26.0	26.7	11.4	45.4	45.6	23.7
Average		38.7	34.1	9.6	56.6	49.8	19.3	28.5	28.5	12.7	47.7	46.6	25.3
gpt-neox-20b-entity	✗	37.1	29.4	6.4	52.5	41.9	13.6	20.2	19.7	8.2	31.9	31.2	15.6
Δ Average		-1.5	-4.7	-3.2	-4.1	-7.8	-5.7	-8.2	-8.7	-4.5	-15.8	-15.4	-9.7
Δ Median		-1.1	-4.7	-3.5	-3.9	-7.7	-6.6	-9.1	-8.9	-4.8	-16.8	-15.4	-10.0
gpt-neox-20b-pair	✗	34.1	30.2	5.8	46.7	41.4	8.4	24.4	25.4	8.8	38.7	38.4	14.4
Δ Average		-4.6	-4.0	-3.8	-9.9	-8.4	-10.9	-4.0	-3.1	-3.9	-9.1	-8.2	-10.9
Δ Median		-4.2	-3.9	-4.1	-9.6	-8.2	-11.8	-4.8	-3.3	-4.2	-10.0	-8.2	-11.2

Table 4: **Performance (Hits@K)** comparison between supervised models and ICL for single-step (top) and multi-step (bottom) prediction, grouped by the number of **number of unique entities** as confounder. The first group consists of supervised models, whereas the second group consists of ICL models, *i.e.,* GPT-NeoX with a history length of 100. The green and red colors represent where LLM is outperforming and underperforming the average performance of the supervised models.



(a) ICEWS14 Single-step

(b) ICEWS18 Single-step



(a) ICEWS14 Multi-step

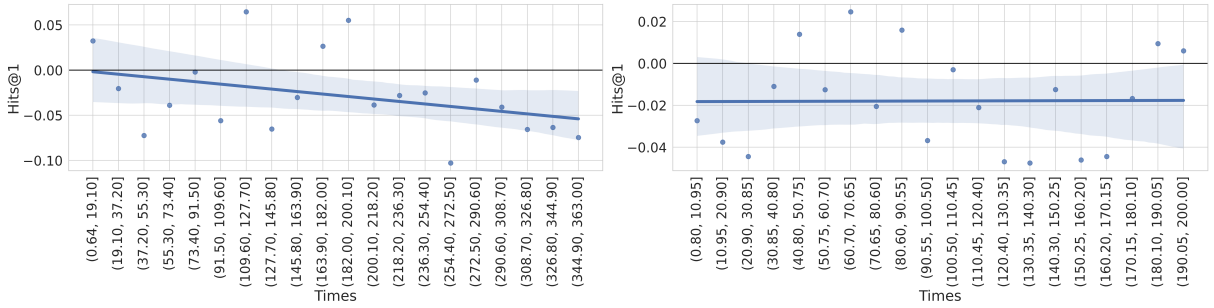
(b) ICEWS18 Multi-step

Figure 2: **Hits@1 difference** between the average performance of ICL and the average performance of supervised models, grouped by the **relation frequency** confounder, for single-step (top) and multi-step (bottom) prediction.

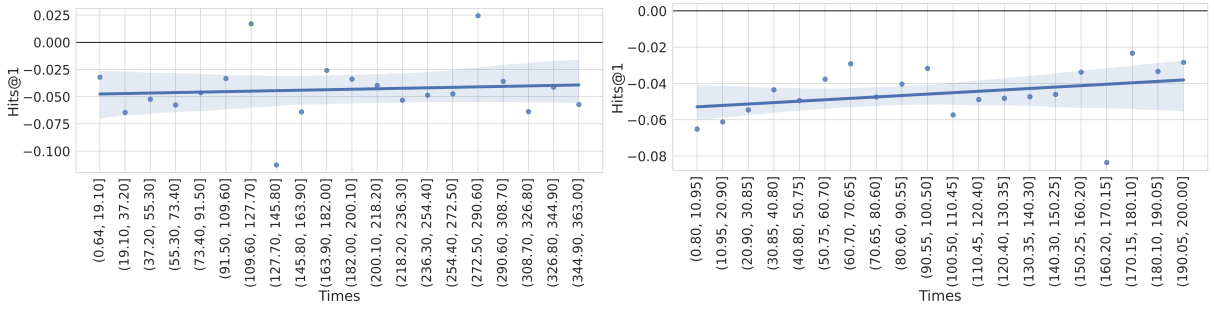
Single-step	Train	ICEWS14								ICEWS18							
		H@1				H@3				H@1				H@3			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
RE-GCN	✓	49.5	35.7	34.4	40.3	71.7	54.8	53.0	55.3	34.6	31.9	30.8	28.4	55.0	51.2	48.6	45.6
TANGO	✓	45.5	29.2	31.8	38.0	64.6	45.4	46.9	49.7	30.9	26.9	27.6	24.8	50.8	45.1	44.1	41.1
CEN	✓	50.1	36.6	34.5	40.3	70.2	54.9	52.4	56.3	33.8	31.3	30.1	27.9	53.9	50.4	47.7	44.9
Average		48.4	33.8	33.5	39.6	68.9	51.7	50.8	53.8	33.1	30.0	29.5	27.0	53.3	48.9	46.8	43.9
Median		49.5	35.7	34.4	40.3	70.2	54.8	52.4	55.3	33.8	31.3	30.1	27.9	53.9	50.4	47.7	44.9
gpt-neox-20b-entity	✗	57.0	36.3	35.1	35.8	77.0	53.1	50.0	51.2	30.1	28.2	24.2	22.4	48.2	45.6	39.1	35.7
Δ Average		8.6	2.4	1.6	-3.7	8.1	1.4	-0.8	-2.6	-3.0	-1.8	-5.2	-4.6	-5.1	-3.3	-7.7	-8.2
Δ Median		7.4	0.5	0.8	-4.5	6.7	-1.7	-2.4	-4.1	-3.7	-3.1	-5.8	-5.5	-5.8	-4.7	-8.6	-9.3
gpt-neox-20b-pair	✗	58.5	29.5	34.4	32.2	82.0	40.8	48.0	41.4	35.8	27.9	26.8	22.6	56.6	44.1	41.1	33.4
Δ Average		10.1	-4.3	0.9	-7.4	13.1	-10.9	-2.8	-12.4	2.7	-2.1	-2.7	-4.4	3.3	-4.8	-5.7	-10.5
Δ Median		9.0	-6.2	0.1	-8.1	11.8	-14.0	-4.4	-13.9	2.0	-3.3	-3.3	-5.2	2.6	-6.3	-6.6	-11.5

Multi-step	Train	ICEWS14								ICEWS18							
		H@1				H@3				H@1				H@3			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
RE-NET	✓	45.5	30.8	31.7	37.0	62.0	46.6	47.6	51.7	30.4	26.1	26.8	24.7	50.8	44.0	44.1	40.8
RE-GCN	✓	42.5	31.2	31.7	35.8	63.6	47.6	46.1	48.5	31.3	26.9	28.6	25.0	50.6	44.8	45.0	41.0
CyGNet	✓	46.6	27.1	31.4	34.2	66.2	43.3	46.9	49.4	28.7	22.7	25.2	21.6	49.0	41.1	42.8	37.8
Average		44.8	29.7	31.6	35.7	63.9	45.8	46.9	49.9	30.1	25.2	26.8	23.7	50.1	43.3	43.9	39.9
Median		45.5	30.8	31.7	35.8	63.6	46.6	46.9	49.4	30.4	26.1	26.8	24.7	50.6	44.0	44.1	40.8
gpt-neox-20b-entity	✗	41.5	28.0	28.7	31.1	56.9	41.4	41.3	44.0	21.9	18.1	17.9	15.5	33.0	29.9	28.4	24.6
Δ Average		-3.3	-1.7	-2.9	-4.6	-7.0	-4.4	-5.6	-5.8	-8.2	-7.2	-9.0	-8.2	-17.1	-13.4	-15.5	-15.3
Δ Median		-3.9	-2.8	-3.0	-4.7	-6.7	-5.2	-5.6	-5.4	-8.5	-8.0	-8.9	-9.1	-17.5	-14.1	-15.6	-16.2
gpt-neox-20b-pair	✗	44.8	22.3	29.5	28.6	62.8	31.2	40.3	35.8	29.0	21.0	23.0	18.5	46.7	33.3	34.1	27.5
Δ Average		-0.1	-7.4	-2.1	-7.0	-1.2	-14.7	-6.6	-14.1	-1.2	-4.3	-3.8	-5.2	-3.4	-10.0	-9.9	-12.3
Δ Median		-0.7	-8.5	-2.2	-7.2	-0.8	-15.5	-6.6	-13.7	-1.4	-5.1	-3.7	-6.1	-3.9	-10.6	-10.0	-13.2

Table 5: **Performance (Hits@K)** comparison between supervised models and ICL for single-step (top) and multi-step (bottom) prediction, grouped by the number of **number of unique relations** as confounder. The first group consists of supervised models, whereas the second group consists of ICL models, *i.e.,* GPT-NeoX with a history length of 100. The green and red colors represent where LLM is outperforming and underperforming the average performance of the supervised models.



(a) ICEWS14 Single-step (b) ICEWS18 Single-step



(a) ICEWS14 Multi-step (b) ICEWS18 Multi-step

Figure 3: **Hits@1 difference** between the average performance of ICL and the average performance of supervised models, grouped by the **time interval** confounder, for single-step (top) and multi-step (bottom) prediction.