Temporal Knowledge Graph Forecasting Without Knowledge Using In-context Learning

Dong-Ho Lee*, Kian Ahrabian*, Woojeong Jin, Fred Morstatter, Jay Pujara
Department of Computer Science and Information Sciences Institute, University of Southern California

Motivation

Which team will win the Super Bowl in 2023?

```
2000: [Superbowl, Champion, St Louis]
2001: [Superbowl, Champion, Baltimore]
2002: [Superbowl, Champion, New England]
2003: [Superbowl, Champion, Tampa Bay]
...
2019: [Superbowl, Champion, New England]
2020: [Superbowl, Champion, Kansas City]
2021: [Superbowl, Champion, Tampa Bay]
2022: [Superbowl, Champion, Los Angeles]
2023: [Superbowl, Champion,
```

Model	# Params.	Prediction	G.T Rank
EleutherAI gpt-j-6b	6B	Los Angeles	3
EleutherAI gpt-neox-20b	20B	Kansas City	1
OpenAI text-ada-001	350M	Carolina	>5
OpenAI text-babbage-001	1.3B	Indianapolis	>5
OpenAI text-curie-001	6.7B	New England	>5
OpenAI text-davinci-003	175B	TBD	>5
OpenAI gpt-3.5-turbo	-	Sorry, I cannot predict future events.	

Can large language models **forecast** the missing fact using **in-context learning (ICL)**?

TKG reasoning using ICL

(Superbowl, Champion, ?, 2023)

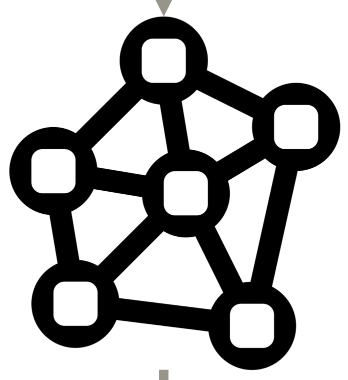
1. History Modeling

Entity vs. Pair
Uni- vs. Bi-direction

Entity: contains "Superbowl"
Pair: contains both "Superbowl, Champion"

Uni-: "Superbowl" placed in "subject"
Bi-: "Superbowl" placed in "subject" or "object"

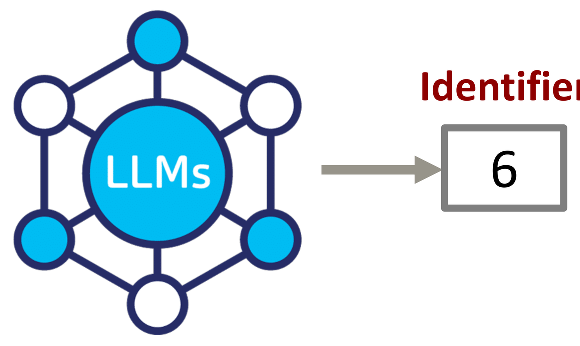
Knowledge Graph



2. Prompt Construction

Lexical vs. Index

Category	Prompt
Lexical $\mathcal{L}(\cdot)$	2000: [Superbowl, Champion, 0. St Louis] 2001: [Superbowl, Champion, 1. Baltimore] ... Identifier 2023: [Superbowl, Champion,
Index $\mathcal{I}(\cdot)$	2000: [0, 0, 0. 0] 2001: [0, 0, 1. 1] ... Identifier 2023: [0, 0,



3. Response Generation

Experimental Setup

Dataset	# Ents	# Rels	# of Facts			Interval
			Train	Valid	Test	
WIKI	12,554	24	539,286	67,538	63,110	1 year
YAGO	10,623	10	161,540	19,523	20,026	1 year
ICEWS14	6,869	230	74,845	8,514	7,371	1 day
ICEWS18	23,033	256	373,018	45,995	49,995	1 day
ACLED-CD22	243	6	1,788	216	222	1 day

Dataset is divided based on time where *train* < *valid* < *test*

Timestamps

Train: 1~100
Valid: 101~200
Test: 201~250

Single-step

Infer 201 using 1~200
Infer 202 using 1~201
Infer 203 using 1~202

Multi-step

Infer 201 using 1~200
Infer 202 using 1~200 + *prev preds*
Infer 203 using 1~200 + *prev preds*

Experimental Results

Single-Step	Train	YAGO			WIKI			ICEWS14			ICEWS18			ACLED-CD22		
		H@1	H@3	H@10	H@1	H@3	H@10	H@1	H@3	H@10	H@1	H@3	H@10	H@1	H@3	H@10
RE-GCN	✓	0.787	0.842	0.884	0.747	0.817	0.846	0.313	0.473	**0.626**	**0.223**	**0.367**	**0.525**	**0.446**	**0.545**	**0.608**
xERTE	✓	0.842	0.902	**0.912**	0.703	0.785	0.801	0.330	0.454	0.570	0.209	0.335	0.462	0.320	0.445	0.497
TLogic	✓	0.740	0.789	0.791	**0.786**	**0.860**	**0.870**	**0.332**	**0.476**	0.602	0.204	0.336	0.480	0.009	0.045	0.094
TANGO	✓	0.590	0.646	0.677	0.483	0.514	0.527	0.272	0.408	0.550	0.191	0.318	0.462	0.327	0.482	0.599
Timetraveler	✓	**0.845**	**0.908**	**0.912**	0.751	0.820	0.830	0.319	0.454	0.575	0.212	0.325	0.439	0.240	0.315	0.457
GPT-NeoX (Entity)	✗	0.784	0.891	**0.927**	0.694	0.804	0.844	**0.324**	**0.460**	**0.565**	0.192	**0.313**	**0.414**	**0.324**	**0.492**	**0.604**
GPT-NeoX (Pair)	✗	**0.787**	**0.892**	0.926	**0.721**	**0.812**	**0.847**	0.297	0.408	0.482	**0.196**	0.307	0.402	0.317	0.440	0.566

Multi-Step	Train	YAGO			WIKI			ICEWS14			ICEWS18			ACLED-CD22		
		H@1	H@3	H@10	H@1	H@3	H@10	H@1	H@3	H@10	H@1	H@3	H@10	H@1	H@3	H@10
RE-GCN	✓	**0.717**	**0.776**	0.817	0.594	0.648	0.678	**0.278**	**0.421**	**0.575**	**0.195**	**0.326**	**0.475**	**0.421**	0.464	0.502
RE-Net	✓	0.534	0.613	0.662	0.472	0.507	0.530	**0.278**	0.408	0.549	0.184	0.314	0.461	0.238	0.445	0.563
CyGNet	✓	0.613	0.742	**0.834**	0.525	0.624	0.675	0.266	0.402	0.545	0.166	0.295	0.444	0.408	**0.500**	**0.588**
TLogic	✓	0.631	0.706	0.715	**0.613**	**0.663**	**0.682**	0.265	0.395	0.531	0.155	0.272	0.412	0.009	0.045	0.094
GPT-NeoX (Entity)	✗	0.686	**0.793**	**0.840**	0.543	0.622	**0.655**	**0.247**	**0.363**	**0.471**	0.136	0.224	0.321	**0.319**	**0.417**	**0.500**
GPT-NeoX (Pair)	✗	**0.688**	**0.793**	0.839	**0.570**	**0.625**	0.652	0.236	0.324	0.395	**0.155**	**0.245**	**0.331**	0.289	0.410	0.464

ICL using (**Entity**, **Uni-direction**, **Index**)-formatted prompt shows similar performance to supervised graph representation learning

Does ICL use heuristics?

Single-Step	ICEWS14			ICEWS18		
	H@1	H@3	H@10	H@1	H@3	H@10
frequency	0.243	0.387	0.532	0.141	0.265	0.409
recency	0.228	0.387	0.536	0.120	0.242	0.403
GPT-NeoX (Entity)	**0.324**	**0.460**	**0.565**	0.192	**0.313**	**0.414**
GPT-NeoX (Pair)	0.297	0.408	0.482	**0.196**	0.307	0.402

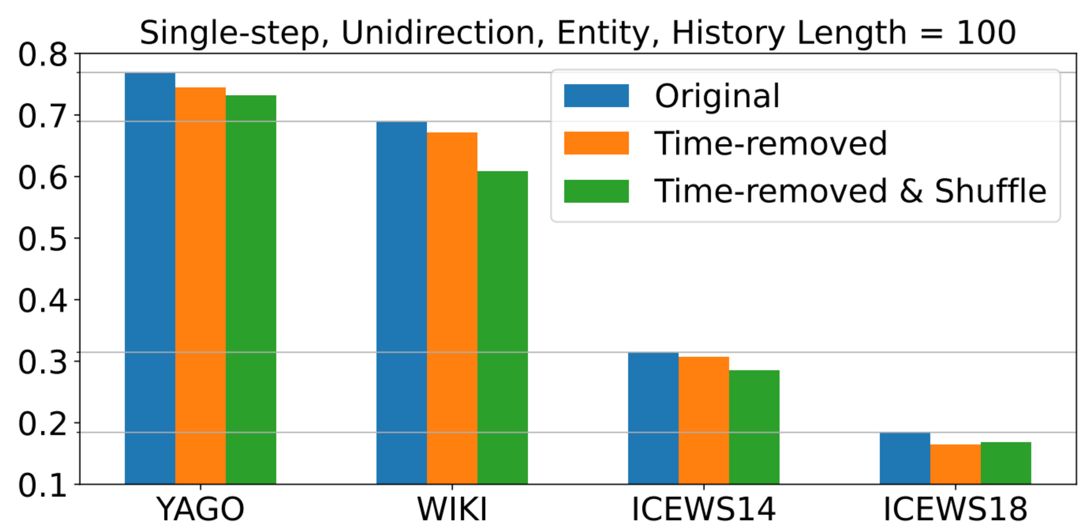
(a) Single-step

Multi-Step	ICEWS14			ICEWS18		
	H@1	H@3	H@10	H@1	H@3	H@10
frequency	0.222	0.349	0.460	0.121	0.207	0.307
recency	0.151	0.268	0.423	0.074	0.149	0.266
GPT-NeoX (Entity)	**0.247**	**0.363**	**0.471**	0.136	0.224	0.321
GPT-NeoX (Pair)	0.236	0.324	0.395	**0.155**	**0.245**	**0.331**

(b) Multi-step

NO, ICL does not rely on specific biases (i.e., frequency, recency). It learns more sophisticated patterns from historical data.

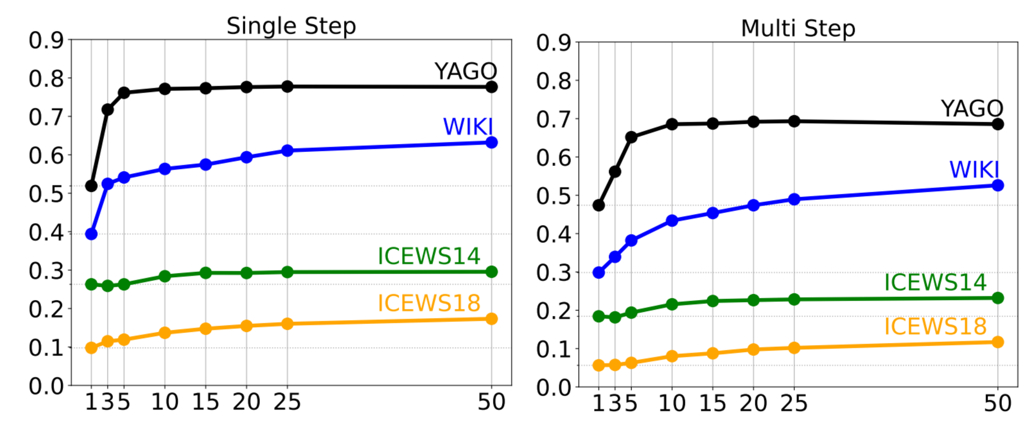
Does ICL use time?



YES, ICL forecasts the next event by comprehending the sequential order of events.

https://github.com/usc-isi-i2/isi-tkg-icl

History length scaling



Model size scaling