NARRATIVE-OF-THOUGHT: Improving Temporal Reasoning of Large Language Models via Recounted Narratives

Xinliang Frederick Zhang[1], Nick Beauchamp[2], and Lu Wang[1]

[1]Computer Science and Engineering, University of Michigan
[2]Political Science and Network Science Institute, Northeastern University

EMNLP 2024

Introduction: Temporal Reasoning

Preamble: Temporal reasoning is essential for humans to perceive the world, understand daily communications, and interpret the temporal aspects of experiences (Allen, 1983; Nebel and Bürckert, 1995).

Introduction: Temporal Reasoning

Preamble: Temporal reasoning is essential for humans to perceive the world, understand daily communications, and interpret the temporal aspects of experiences (Allen, 1983; Nebel and Bürckert, 1995).

Background:

- The recent advent of LLMs has gathered substantial attention to reasoning tasks, while few LLMs exist to handle temporal reasoning well.
- This task is inherently complex, mingled with implicit logical inference and the necessity for profound world knowledge.
- Existing research mainly focuses on a simple relation extraction task OR a perplexing commonsense understanding task.

Introduction: Temporal Reasoning

Preamble: Temporal reasoning is essential for humans to perceive the world, understand daily communications, and interpret the temporal aspects of experiences (Allen, 1983; Nebel and Bürckert, 1995).

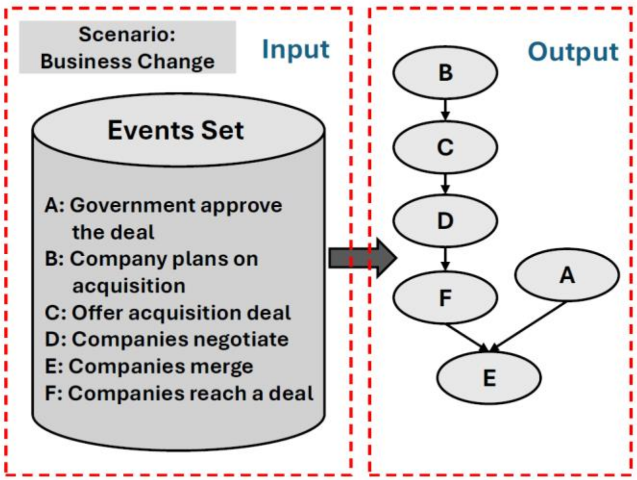
Background:

- The recent advent of LLMs has gathered substantial attention to reasoning tasks, while few LLMs exist to handle temporal reasoning well.
- This task is inherently complex, mingled with implicit logical inference and the necessity for profound world knowledge.
- Existing research mainly focuses on a simple relation extraction task OR a perplexing commonsense understanding task.

Research objectives: Uncover and improve the inherent, global temporal reasoning capabilities of LLMs.

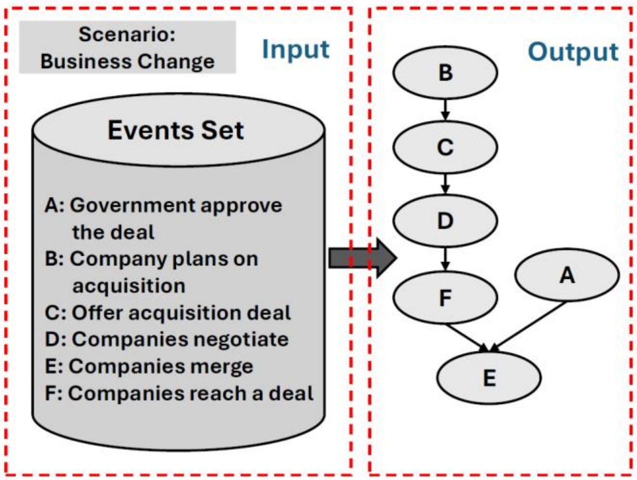
Task: Temporal Graph Generation (TGG)

Task formulation: Given a high-level goal T (e.g., business change) and a set of events V, the objective is to produce a temporal graph G(V, E) where a directed edge in E reveals the temporal order between events.



Task: Temporal Graph Generation (TGG)

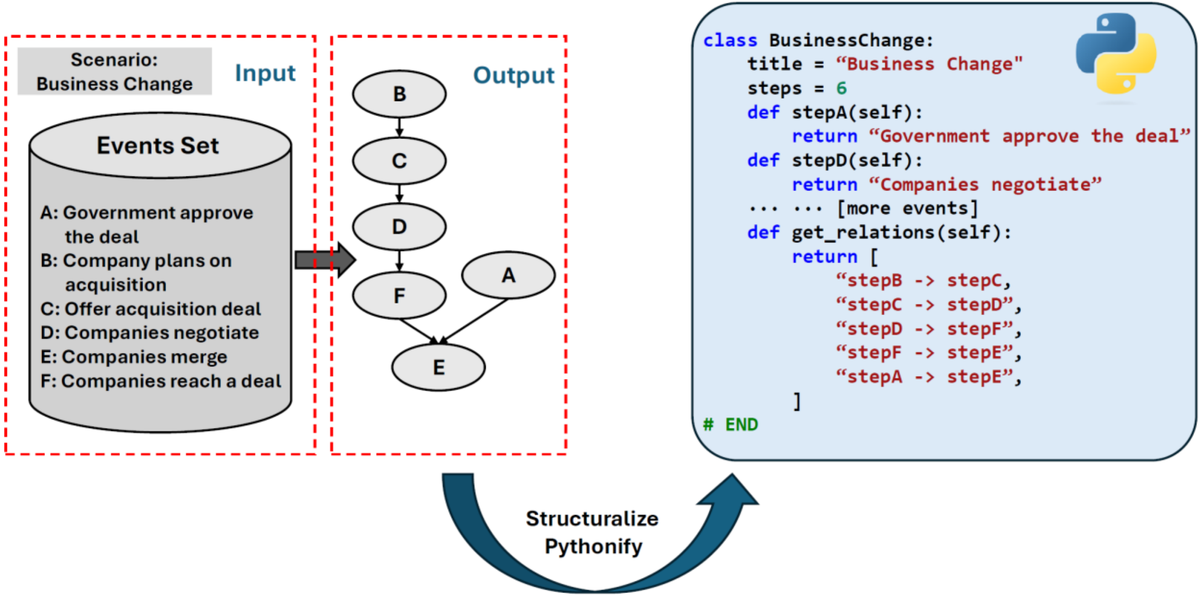
Task formulation: Given a high-level goal T (e.g., business change) and a set of events V, the objective is to produce a temporal graph G(V, E) where a directed edge in E reveals the temporal order between events.



Finding: Small, open-weight LLMs (<10B parameters) lag behind large, proprietary LLMs by 25 F1 points.

Method: Narrative-of-thought (NoT)

1. Given a scenario and a set of events, NoT first converts the input into a Python class.



Method: Narrative-of-thought (NoT)

1. Given a scenario and a set of events, NoT first converts the input into a Python class.
2. NoT guides LLMs to produce a temporally grounded narrative by arranging events in the correct temporal order, leveraging LLMs' intrinsic temporal knowledge.

```
class BusinessChange:
    def stepE(self):
        return “companies merge”
    def stepA(self):
        return “government approve
the deal”
    ··· ··· [more events]
    #Let's think about a
narrative ···
    def get_narrative(self):
        #TODO
        return “This is a report
about ‘business change’. First,
companies plan on an acquisition.
Then, they offer an acquisition
deal to the other company. The
other company accepts the deal and
the two companies start
negotiating the terms of the deal.
After they reach an agreement,
they submit the deal to the
government for approval. Once the
government approves the deal, the
companies can merge. By adhering
to the provided temporal
information, the desired goal is
achieved.”
```

Method: Narrative-of-thought (NoT)

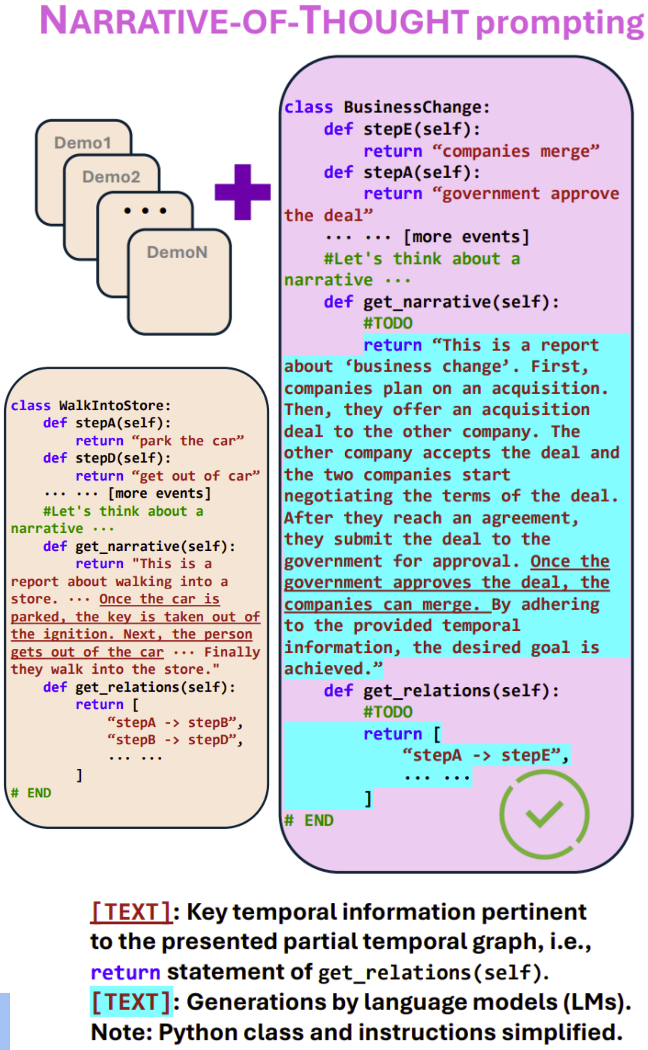
1. Given a scenario and a set of events, NoT first converts the input into a Python class.
2. NoT guides LLMs to produce a temporally grounded narrative by arranging events in the correct temporal order, leveraging LLMs’ intrinsic temporal knowledge.
3. Based on the recounted temporal relations articulated in the narrative, LLMs are instructed to sort events into a temporal graph.

```
class BusinessChange:
    def stepE(self):
        return “companies merge”
    def stepA(self):
        return “government approve
the deal”
    ··· ··· [more events]
    #Let's think about a
narrative ···
    def get_narrative(self):
        #TODO
        return “This is a report
about ‘business change’. First,
companies plan on an acquisition.
Then, they offer an acquisition
deal to the other company. The
other company accepts the deal and
the two companies start
negotiating the terms of the deal.
After they reach an agreement,
they submit the deal to the
government for approval. Once the
government approves the deal, the
companies can merge. By adhering
to the provided temporal
information, the desired goal is
achieved.”
    def get_relations(self):
        #TODO
        return [
            “stepA -> stepE”,
            ··· ···
        ]
# END
```

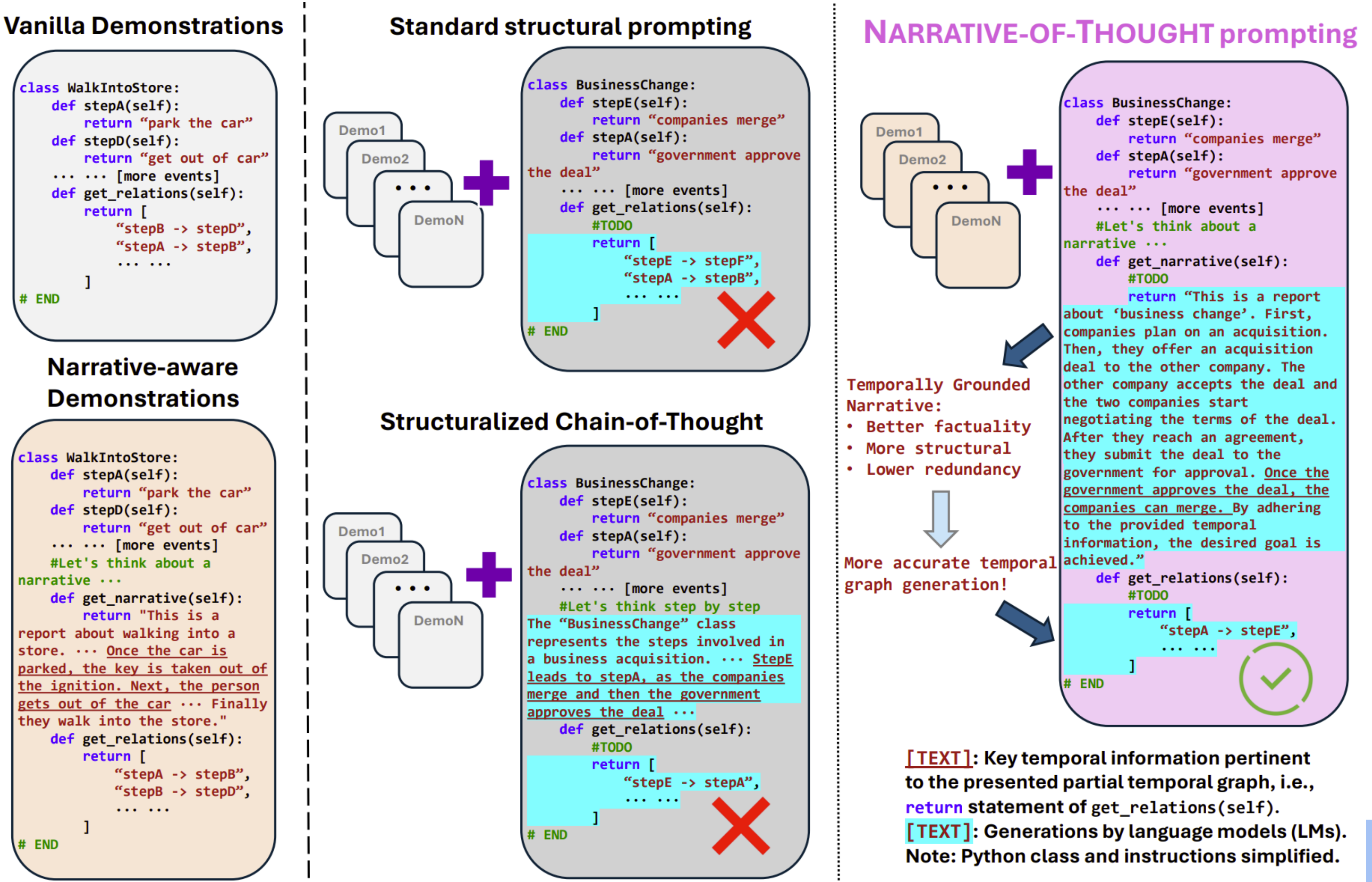
Method: Narrative-of-thought

1. Given a scenario and a set of events, NoT first converts the input into a Python class.
2. NoT guides LLMs to produce a temporally grounded narrative by arranging events in the correct temporal order, leveraging LLMs' intrinsic temporal knowledge.
3. Based on the recounted temporal relations articulated in the narrative, LLMs are instructed to sort events into a temporal graph.

* We further improve NoT by introducing high-quality reference narratives as part of few-shot demonstrations.



Narrative-of-Thought overview and comparison



Experimental Setup

Dataset: ProScript (Sakaguchi et al., 2021), Schema-11 evaluation set (Dror et al., 2023), and WikiHow Script corpus (Lyu et al., 2021).

	#scenarios	#events	Max #events	#temporal links	Event length	%Non-linear	Domain
ProScrpt (Sakaguchi et al.)	2,077	7.46	9	6.95	4.64	39%	Daily
Schema-11 (Dror et al.)	11	7.91	11	7.18	3.48	27%	News
WikiHow Script (Lyu et al.)	291	8.37	20	7.37	9.63	0%	Daily

Base LLMs:

- **Open-weights:** MISTRAL-7B (Jiang et al., 2023), GEMMA-7B (Mesnard et al., 2024), and LLAMA3-8B (AI@Meta, 2024).
- **Propriertary:** GPT-3.5 and GPT-4 (OpenAI, 2023).

Evaluation Metric: We compare both semantic and structural similarities between ground-truth temporal graph and machine-generated ones. We also report Pair-wise Consistency between two generated graphs by the same model.

- **Semantic similarity:** we report edge-wise precision (P), recall (R) and F1.
- **Structural similarity:** We adopt Graph Edit Distance (GED; Abu-Aisheh et al., 2015) and Graph Statistics.

Experimental Results (Selected)

Method	Proscript				Schema-11				WikiHow Script				Avg.	
	F1↑	GED↓	$k(\mathcal{G})$	Cons.↑	F1↑	GED↓	$k(\mathcal{G})$	Cons.↑	F1↑	GED↓	$k(\mathcal{G})$	Cons.↑	F1↑	GED↓
Baselines														
Random	14.0	1.47	1.00	7.8	19.4	3.91	1.00	7.8	14.2	0.06	1.00	8.8	15.9	1.81
GPT-3.5 (0-shot)*	18.4	2.25	1.06	38.6	30.1	4.48	1.27	30.2	17.2	2.80	1.11	40.8	21.9	3.18
GPT-3.5	43.4	1.71	1.07	38.8	62.8	3.30	1.36	50.2	31.0	1.58	1.10	35.4	45.7	2.20
GPT-4	63.9	1.64	1.02	61.4	44.1	7.97	0.64	46.3	43.0	1.71	1.04	48.5	50.3	3.77
LLAMA3-8B (AI@Meta, 2024)														
Standard Prompting	25.1	2.39	1.18	19.9	28.3	4.42	1.24	19.9	20.6	1.17	1.07	21.2	24.7	2.66
Chain-of-Thought	30.1	2.06	**1.00**	23.3	37.3	5.79	0.85	23.5	22.6	**0.99**	**1.02**	**24.3**	30.0	2.95
NOT (no reference)	35.5	1.88	**1.00**	25.3	52.6	**3.18**	1.12	35.0	25.4	**0.99**	**1.02**	20.9	37.8	**2.02**
NOT (alphabetical meta)	**39.5**	1.87	1.01	**28.8**	59.0	3.72	1.12	39.1	26.3	1.01	1.03	22.5	41.6	2.20
NOT (descriptive meta)	38.7	**1.86**	1.01	28.4	**61.5**	3.57	**1.09**	**45.6**	**26.5**	1.04	1.03	22.3	**42.2**	2.16

Note: Results of Gemma and Mistral refer to our paper. Results of fine-tuning also refer to our paper.

Experimental Results (Selected)

Method	Proscript				Schema-11				WikiHow Script				Avg.	
	F1↑	GED↓	$k(\mathcal{G})$	Cons.↑	F1↑	GED↓	$k(\mathcal{G})$	Cons.↑	F1↑	GED↓	$k(\mathcal{G})$	Cons.↑	F1↑	GED↓
Baselines														
Random	14.0	1.47	1.00	7.8	19.4	3.91	1.00	7.8	14.2	0.06	1.00	8.8	15.9	1.81
GPT-3.5 (0-shot)*	18.4	2.25	1.06	38.6	30.1	4.48	1.27	30.2	17.2	2.80	1.11	40.8	21.9	3.18
GPT-3.5	43.4	1.71	1.07	38.8	62.8	3.30	1.36	50.2	31.0	1.58	1.10	35.4	45.7	2.20
GPT-4	63.9	1.64	1.02	61.4	44.1	7.97	0.64	46.3	43.0	1.71	1.04	48.5	50.3	3.77
LLAMA3-8B (AI@Meta, 2024)														
Standard Prompting	25.1	2.39	1.18	19.9	28.3	4.42	1.24	19.9	20.6	1.17	1.07	21.2	24.7	2.66
Chain-of-Thought	30.1	2.06	**1.00**	23.3	37.3	5.79	0.85	23.5	22.6	**0.99**	**1.02**	**24.3**	30.0	2.95
NoT (no reference)	35.5	1.88	**1.00**	25.3	52.6	**3.18**	1.12	35.0	25.4	**0.99**	**1.02**	20.9	37.8	**2.02**
NoT (alphabetical meta)	**39.5**	1.87	1.01	**28.8**	59.0	3.72	1.12	39.1	26.3	1.01	1.03	22.5	41.6	2.20
NoT (descriptive meta)	38.7	**1.86**	1.01	28.4	**61.5**	3.57	**1.09**	**45.6**	**26.5**	1.04	1.03	22.3	**42.2**	2.16

1) Small LLMs **struggle** with temporal reasoning even with few-shot examples.
2) CoT is also ineffective at temporal reasoning, in line with existing findings (Chu et al., 2023).
3) GPT-4 sometimes falls off the throne due to **additional alignment**, when answering sensitive queries.
4) NoT is a powerful tool to assist small LLMs to **catch up with or even surpass GPT-3.5**, and presents **strong compatibility** with various base LLMs. The average **F1 improvements are between 16%-71%.**
5) Temporally grounded **narratives** are significant in improving LLMs' temporal reasoning process.
6) AI systems are far from mastering temporal reasoning, **trailing** the human baseline by **30 F1 points**.

Further Analyses

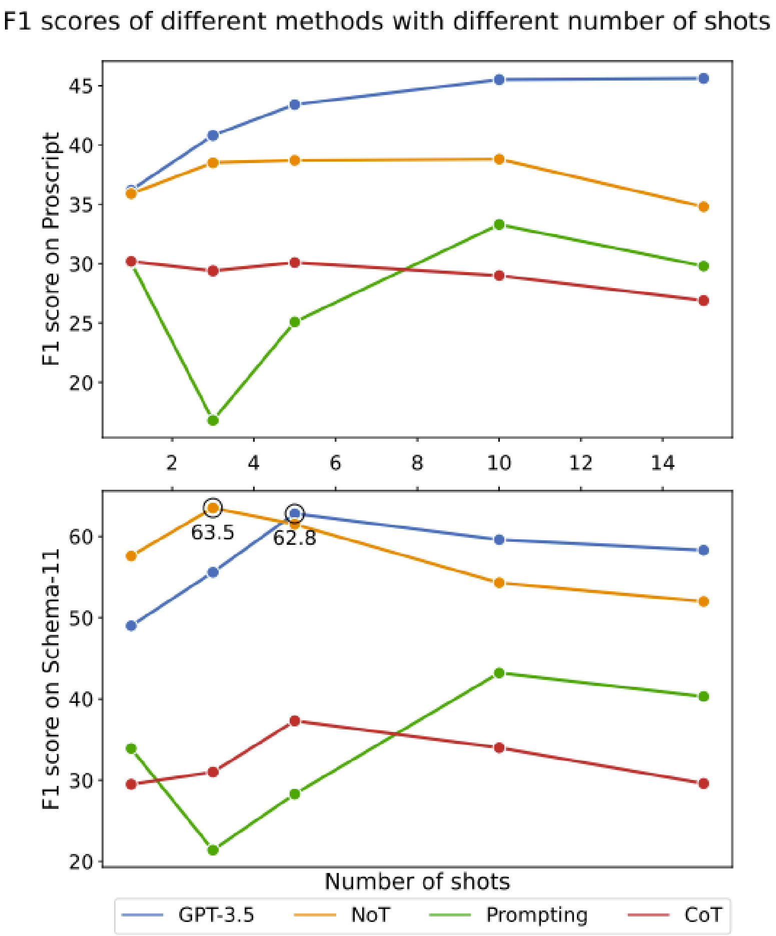
RQ1: Does the number of shots matter?

RQ2: What characteristics define effective reference narratives?

RQ3: How faithful is the temporal graph to intermediate narratives?

Further Analyses

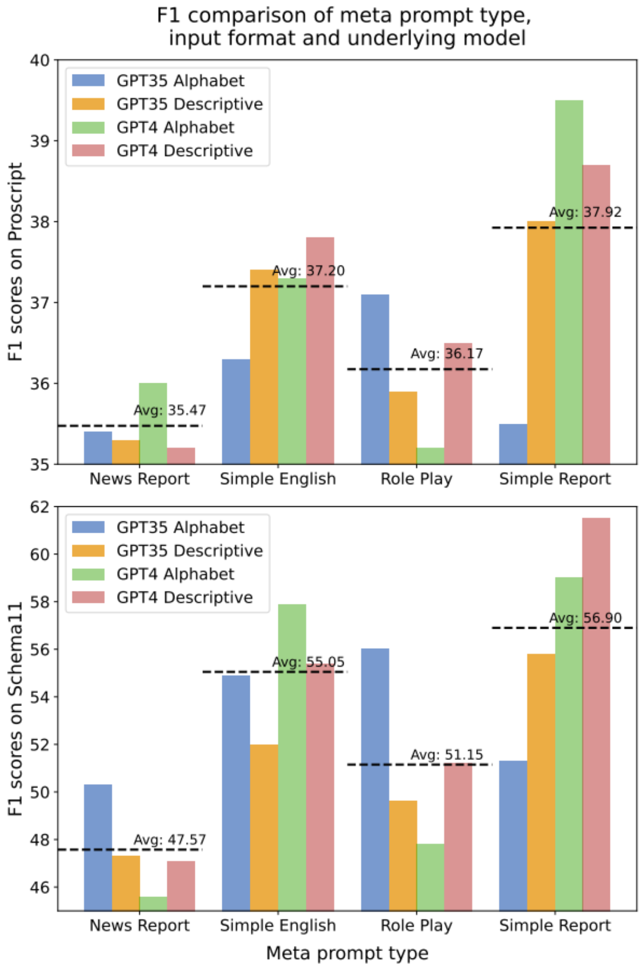
RQ1: Does the number of shots matter?
Ans: The performance generally reaches its peak around the range of **5-10 shots.**



Further Analyses

RQ1: Does the number of shots matter?
Ans: The performance generally reaches its peak around the range of 5-10 shots.

RQ2: What characteristics define effective reference narratives?
Ans: We identify three key characteristics for quality reference narratives: **conciseness**, **simplicity** and **factuality**.



Further Analyses

RQ1: Does the number of shots matter?
Ans: The performance generally reaches its peak around the range of 5-10 shots.

RQ2: What characteristics define effective reference narratives?
Ans: we identify three key characteristics for quality reference narratives: conciseness, simplicity and factuality.

RQ3: How faithful is the temporal graph to intermediate narratives?
Ans: We find a medium-to-high self-faithfulness of **72.8%** where the generated narrative and the temporal graph is **aligned** in terms of the temporal order of events.

Thanks!

Paper

Codebase & Data

Codebase and dataset are available at https://github.com/launchnlp/NoT.

Contact: xlfzhang@umich.edu

This work supported by AFOSR and UM Advanced Research Computing.