What do Large Language Models Learn about Scripts?

Abhilasha Sancheti[12], and Rachel Rudinger[1]

[1]University of Maryland, College Park

[2]Adobe Research

Scan Me

What are Scripts? Why are they important?

What are Scripts? Why are they important?

Scripts are structured commonsense knowledge in the form of *event sequences* that characterize commonplace scenarios, such as, *eating at a restaurant* (Schank and Abelson, 1975)

What are Scripts? Why are they important?

Scripts are structured commonsense knowledge in the form of *event sequences* that characterize commonplace scenarios, such as, *eating at a restaurant* (Schank and Abelson, 1975)

Scripts are fundamental pieces of commonsense knowledge that humans share and assume to be tacitly understood by each other

What are Scripts? Why are they important?

Scripts are structured commonsense knowledge in the form of *event sequences* that characterize commonplace scenarios, such as, *eating at a restaurant* (Schank and Abelson, 1975)

Scripts are fundamental pieces of commonsense knowledge that humans share and assume to be tacitly understood by each other

Script Knowledge, whether implicit or explicit, has been recognized as important for language understanding tasks such as causal structure of events (Miikkulainen, 1995 and Mueller, 2004)

Research Questions

Research Questions

1

Is explicit script knowledge present and accessible through pre-trained generative language models?

Research Questions

1. Is explicit script knowledge present and accessible through pre-trained generative language models?
2. Can we induce script knowledge in pre-trained language models?

Our Contributions

Our Contributions

We present an analysis of the **extent of script knowledge accessible through LMs** using probing techniques, in a **zero-shot** setting, via the task of generating **full ESDs from natural language prompts**

Our Contributions

We present an analysis of the **extent of script knowledge accessible through LMs** using probing techniques, in a **zero-shot** setting, via the task of generating **full ESDs from natural language prompts**

We **propose script induction framework (SIF)** that can generate ESDs for held-out and novel scenarios drawn from a different distribution

Our Contributions

We present an analysis of the **extent of script knowledge accessible through LMs** using probing techniques, in a **zero-shot** setting, via the task of generating **full ESDs from natural language prompts**

We **propose script induction framework (SIF)** that can generate ESDs for held-out and novel scenarios drawn from a different distribution

We present **automatic and manual evaluation of the generated ESDs**, establishing the viability of our framework and paving way for future research in this direction

Script Generation Task

Scenario Baking a cake

Event Sequence Description (ESD) 1. gather ingredients 2. mix cake mix, eggs and water in bowl 3. pour into pan 4. turn on oven 5. put in oven and bake at specified temperature 6.remove cake from oven to cool 7. turn off oven 8. mix frosting 9. frost cake 10. serve cake 11. refrigerate any leftovers.

Input to LM Here is a sequence of events that happen when you bake a cake :

Natural Language Prompt **Scenario**

Probing LMs for Script Knowledge

RQ1: Is explicit script knowledge present and accessible through pre-trained generative language models?

- Experiment with large versions of GPT2, BART, and T5

Prompt Beginnings	Continuations
here is a sequence of events that happen while baking a cake:	None
these are the things that happen when you bake a cake:	1.
describe baking a cake in small sequences of short sentences:	1. get a cake mix
here is an ordered sequence of events that occur when you bake a cake:	1. get a cake mix 2. gather together other ingredients

Probing LMs: Qualitative Outputs

- No useful output was generated from BART and T5 in the probing experiment

Probing LMs: Qualitative Outputs

- No useful output was generated from BART and T5 in the probing experiment
- Some scenario relevant events were generated by GPT2

Scenario: BAKING A CAKE 

1	**these are the things that happen when you bake a cake:** 3 tablespoons (30ml) of unsalted butter, at room temperature 2 tablespoons (30ml) of granulated sugar 2 tablespoons (30ml) of brown sugar, sifted 1 large egg, at room temperature 1 teaspoon vanilla extract 3 large egg yolks 1 cup (350ml) buttermilk To the bowl of an electric mixer fitted with a paddle attachment, add the butter, sugar, brown sugar, egg yolks, and vanilla extract and beat until light and fluffy. Add the buttermilk, then continue beating until fully incorporated. Sift the flour, baking powder, and salt into the bowl of the mixer and beat
2	**here is an ordered sequence of events that occur when you bake a cake:** 1. The cake is baked. 2. The cake is cooled. 3. The cake is served at the table. 4. The dessert is placed on the table. 5. The cake is placed in a baking pan. 6. The cake is transferred to the oven. 7. The cake is put to the middle of the oven and baked. 8. The cake is left in the oven until it is done. 9. The cake is removed from the oven and placed on a plate. 10. The cake is transferred to a cake rack. 11. The cake is placed on a cake plate.

Probing LMs: Qualitative Outputs

- No useful output was generated from BART and T5 in the probing experiment
- Some scenario relevant events were generated by GPT2
- Some events were generated with incorrect ordering

Scenario: BAKING A CAKE 

1	**these are the things that happen when you bake a cake:** 3 tablespoons (30ml) of unsalted butter, at room temperature 2 tablespoons (30ml) of granulated sugar 2 tablespoons (30ml) of brown sugar, sifted 1 large egg, at room temperature 1 teaspoon vanilla extract 3 large egg yolks 1 cup (350ml) buttermilk To the bowl of an electric mixer fitted with a paddle attachment, add the butter, sugar, brown sugar, egg yolks, and vanilla extract and beat until light and fluffy. Add the buttermilk, then continue beating until fully incorporated. Sift the flour, baking powder, and salt into the bowl of the mixer and beat
2	**here is an ordered sequence of events that occur when you bake a cake:** 1. The cake is baked. 2. The cake is cooled. 3. The cake is served at the table. 4. The dessert is placed on the table. 5. The cake is placed in a baking pan. 6. The cake is transferred to the oven. 7. The cake is put to the middle of the oven and baked. 8. The cake is left in the oven until it is done. 9. The cake is removed from the oven and placed on a plate. 10. The cake is transferred to a cake rack. 11. The cake is placed on a cake plate.

Probing LMs: Qualitative Outputs

- No useful output was generated from BART and T5 in the probing experiment
- Some scenario relevant events were generated by GPT2
- Some events were generated with incorrect ordering
- Generated ESD is an anecdote or narration rather than procedural description

Scenario: BAKING A CAKE 

describe baking a cake in small sequences of short sentences: "I want my cake to be soft and moist." (And, you know, don't do this: "If I use a pastry bag, I can get it back to me in less than two minutes.") You may be looking for the most simple recipe for cake, but baking requires skill and a lot of practice. You don't need to learn how to make a cake recipe. You don't even need to know the difference between the cake and the sponge. All you need is a great recipe and a couple of ovens and an oven thermometer. 2. Use a mixer. Mixers are really, really good. (A great idea is to invest in one of those mini, little, portable ones.

Probing LMs: Qualitative Outputs

- No useful output was generated from BART and T5 in the probing experiment
- Some scenario relevant events were generated by GPT2
- Some events were generated with incorrect ordering
- Generated ESD is an anecdote or narration rather than procedural description
- Further conditioning with numeral to guide LM to generate a procedure

Probing LMs: Qualitative Outputs

- No useful output was generated from BART and T5 in the probing experiment
- Some scenario relevant events were generated by GPT2
- Some events were generated with incorrect ordering
- Generated ESD is an anecdote or narration rather than procedural description
- Further conditioning with numeral to guide LM to generate a procedure
- A procedure is generated with coarse-grained events with many events missing

Scenario: BAKING A CAKE 

here is a sequence of events that happen while baking a cake: 1. The cake goes into the oven. 2. The cake cools down in the oven, because the water vapor from the oven condenses onto the top of the cake. 3. The cake is cooled and the frosting set (the sugar crystals are not allowed to fall off). 4. The frosting is set (a little more water vapor condenses onto the top of the frosting). 5. The frosting is set (more water vapor condenses onto the top of the frosting). 6. The cake cools down (the water vapor cools off). 7. The cake comes out of the oven and it is ready to eat. How is this cake not a cake? The first step is that the cake is a cake.
here is an ordered sequence of events that occur when you bake a cake: 1. You have a cake. 2. You have a batch of flour (or other ingredient, such as water, sugar or corn syrup) 3. You have your flour prepared 4. You have your flour mixture 5. You have your flour mixture pre-floured 6. You have your batter prepared 7. You have your batter made 8. You have your batter placed on a baking sheet 9. You have your batter removed from the oven 10. You have your cake cooled in the tray (you must place your cake in the tray before you take it out of the oven) 11. You take your cake out of the tray and remove it from the oven 12. You eat your cake, or leave it out to cool.

Probing LMs: Qualitative Outputs

- No useful output was generated from BART and T5 in the probing experiment
- Some scenario relevant events were generated by GPT2
- Some events were generated with incorrect ordering
- Output is an anecdote or narration rather than procedural description
- Further conditioning with numeral to guide LM to generate a procedure
- A procedure is generated with coarse-grained events with many events missing
- To generate fine-grained events, conditioning is done with some events

Probing LMs: Qualitative Outputs

- No useful output was generated from BART and T5 in the probing experiment
- Some scenario relevant events were generated by GPT2
- Some events were generated with incorrect ordering
- Generated ESD is an anecdote or narration rather than procedural description
- Further conditioning with numeral to guide LM to generate a procedure
- A procedure is generated with coarse-grained events with many events missing
- To generate fine-grained events, conditioning is done with some events
- Some duplicate or irrelevant events were generated

Scenario: BAKING A CAKE 

these are the things that happen when you bake a cake: 1. get a cake mix 2. gather together other ingredients that you'll need for your cake (coconut oil, powdered sugar, eggs, etc.) 3. preheat oven to 350 4. pour on flour and mix by hand 5. pour in egg and mix by hand 6. add in powdered sugar and mix by hand 7. add in coconut oil and mix by hand 8. let it all sit for a minute 9. add in your flour and mix by hand 10. add in your sugar and mix by hand 11. add in your flour and mix by hand 12. add in your flour and mix by hand 13. add in your flour and mix by hand 14. add in your flour and mix by hand 15. add in your egg and mix by hand 16. add in your salt

---

describe baking a cake in small sequences of short sentences: 1. get a cake mix 2. gather together other ingredients 3. start to bake 4. bake a cake 5. take it to the oven 6. the cake is done 7. is it done? 8. is it still hot? 9. what is that crust? 10. what does the cake look like? If we think of a sequence of a few hundred or so words, we can take the basic idea of this example and say that we could write an equivalent, yet shorter, sentence: 1. Get a cake mix 2. Get together the ingredients 3. Bake a cake 4. Take it to the oven 5. The cake is done 6. Is it done? 7. is it hot? 8. what is that crust? 9. what does the cake look like

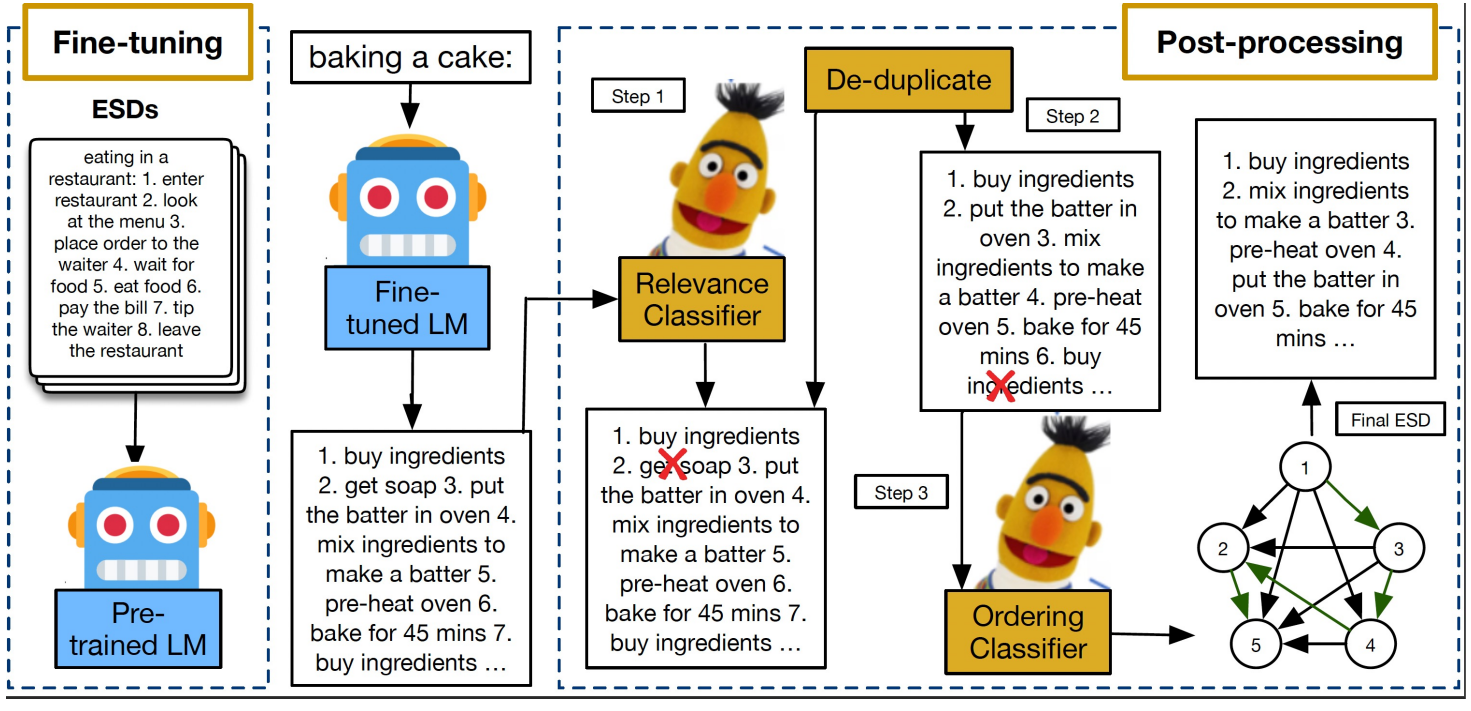
SIF: Script Induction Framework

RQ2: Can we induce script knowledge in pre-trained language models?

SIF: Script Induction Framework

RQ2: Can we induce script knowledge in pre-trained language models?

- 2-staged LM-agnostic script induction framework

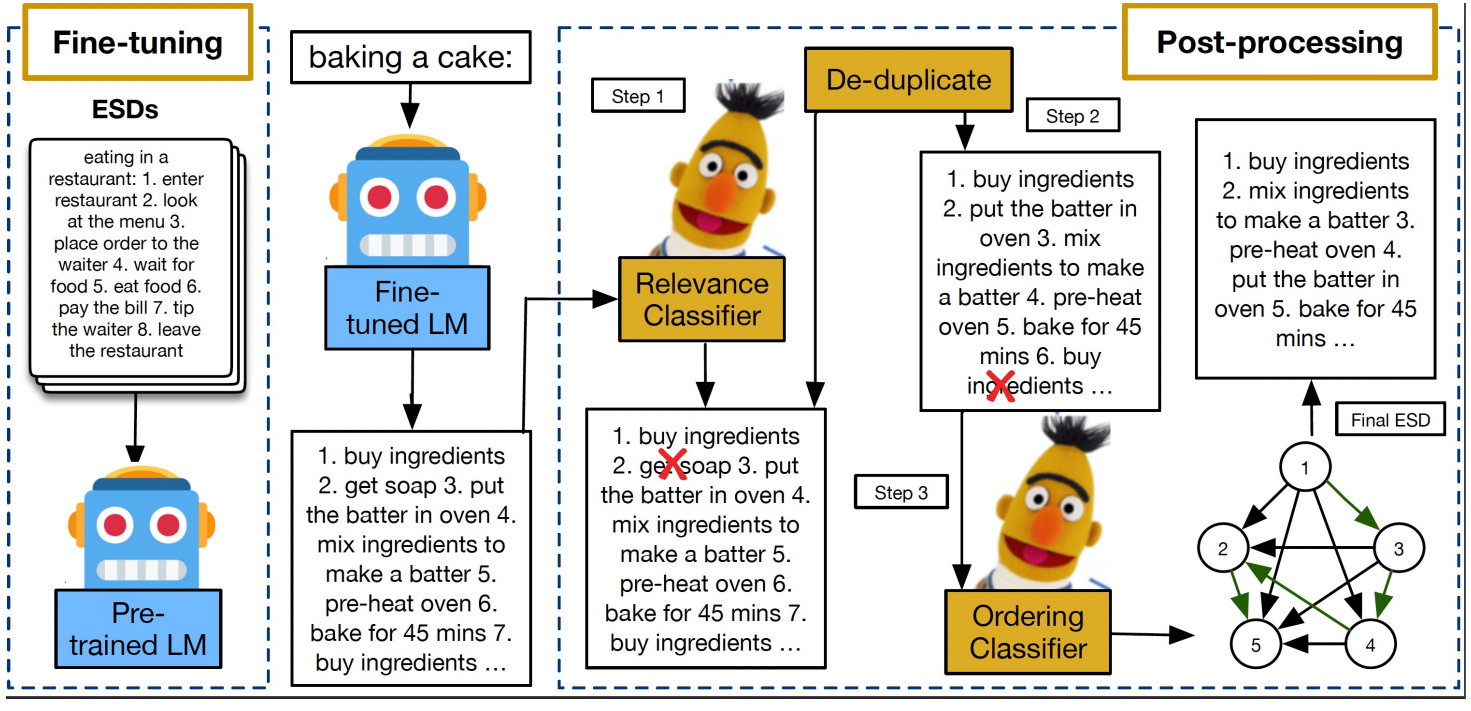


SIF: Script Induction Framework

RQ2: Can we induce script knowledge in pre-trained language models?

- 2-staged LM-agnostic script induction framework
- **Stage 1**: Fine-tuning pre-trained LM on ESDs from DeScript (Wanzare et al., 2016) using different prompt

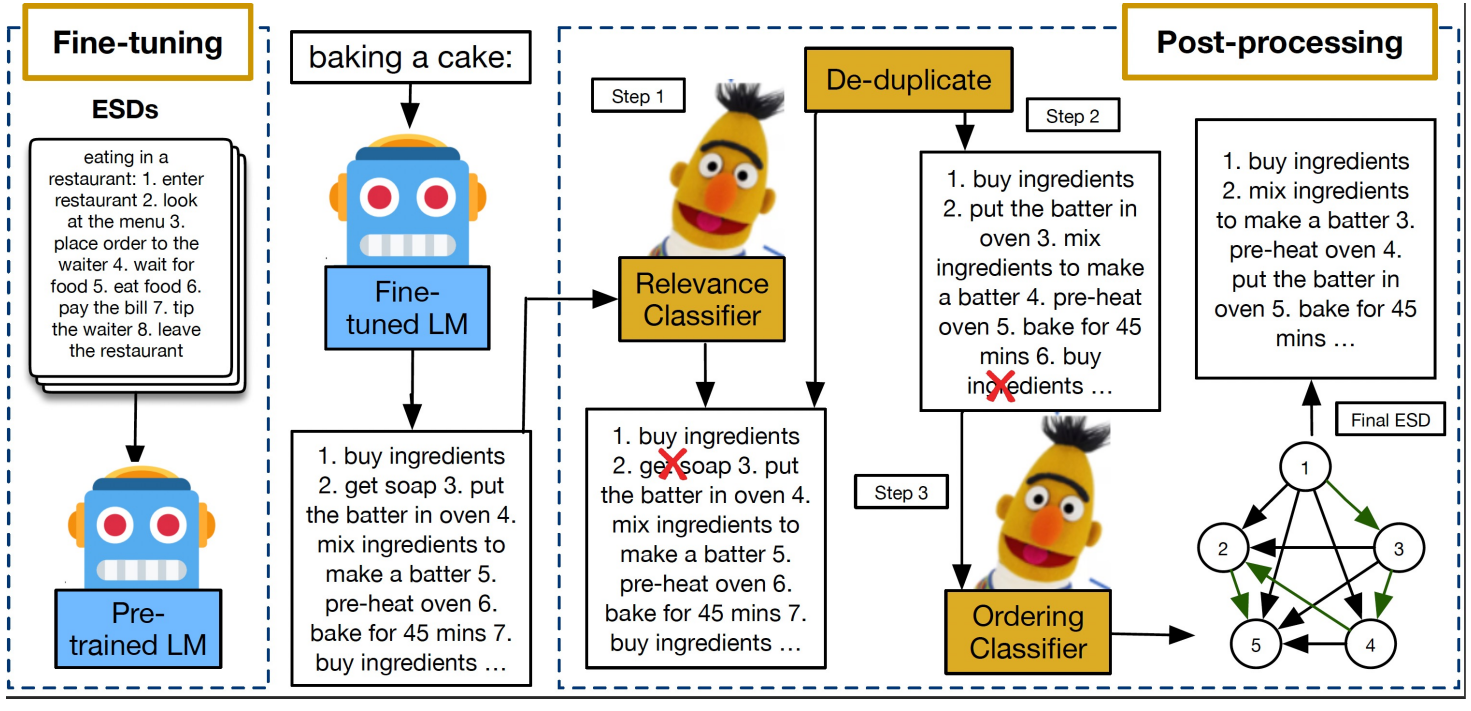
SEQUENCE here is a sequence of events that happen while baking a cake: 1. e_1 2. e_2
EXPECT these are the things that happen when you bake a cake: 1. e_1 2. e_2
ORDERED here is an ordered sequence of events that occur when you bake a cake: 1. e_1 2. e_2
DESCRIBE describe baking a cake in small sequences of short sentences: 1. e_1 2. e_2
DIRECT baking a cake: 1. e_1 2. e_2
TOKENS ⟨SCR⟩ baking a cake ⟨ESCR⟩: 1. e_1 2. e_2
ALLTOKENS ⟨SCR⟩ baking a cake ⟨ESCR⟩: ⟨BEVENT⟩ e_1 ⟨EEVENT⟩ ⟨BEVENT⟩ e_2 ⟨EEVENT⟩



SIF: Script Induction Framework

RQ2: Can we induce script knowledge in pre-trained language models?

- 2-staged LM-agnostic script induction framework
- **Stage 1**: Fine-tuning pre-trained LM on ESDs from DeScript (Wanzare et al., 2016) using different prompt
- **Stage 2:** Post-processing generated ESDs
  - Irrelevant events removal
  - Event de-duplication
  - Temporal order correction



SIF Evaluation

- We use SIF to induce script knowledge in GPT2, BART, and T5

SIF Evaluation

- We use SIF to induce script knowledge in GPT2, BART, and T5
- ESDs generated using SIF are evaluated using BLEU (Papineni et al., 2002) metric given multiple reference ESDs for held-out scenarios

SIF Evaluation

- We use SIF to induce script knowledge in GPT2, BART, and T5
- ESDs generated using SIF are evaluated using BLEU (Papineni et al., 2002) metric given multiple reference ESDs for held-out scenarios

Models	TOKENS	EXPECT	SEQUENCE	ALLTOKENS	DESCRIBE	DIRECT	ORDERED
(1) Zero-shot	03.1 (5.2)	03.6 (5.5)	05.4 (2.8)	03.1 (5.2)	03.2 (3.6)	03.9 (5.1)	06.2 (6.6)
(2) GPT2-$L_{SCRATCH}$	17.2 (3.1)	19.3 (3.7)	16.8 (2.9)	18.6 (4.5)	17.6 (2.6)	14.4 (3.9)	17.7 (3.2)
(3) BART-FT	15.5 (6.0)	20.8 (3.5)	19.6 (3.5)	19.7 (9.2)	19.2 (3.9)	18.0 (6.6)	11.7 (4.8)
(4) GPT2-FT	30.7 (5.1)	31.3 (5.5)	32.4 (6.3)	30.7 (6.6)	32.3 (5.9)	31.4 (5.8)	31.0 (4.8)
(5) BART-**SIF**	16.8 (5.1)	21.1 (4.2)	19.9 (3.7)	20.5 (11.1)	20.0 (3.8)	19.6 (7.2)	13.7 (5.0)
(6) GPT2-**SIF**	**33.6** (5.4)	**33.9** (5.6)	**35.2** (6.9)	**32.5** (6.9)	**34.2** (5.3)	**33.6** (5.7)	**33.2** (5.5)

SIF Evaluation

- We use SIF to induce script knowledge in GPT2, BART, and T5
- ESDs generated using SIF are evaluated using BLEU (Papineni et al., 2002) metric given multiple reference ESDs for held-out scenarios

Models	TOKENS	EXPECT	SEQUENCE	ALLTOKENS	DESCRIBE	DIRECT	ORDERED
(1) Zero-shot	03.1 (5.2)	03.6 (5.5)	05.4 (2.8)	03.1 (5.2)	03.2 (3.6)	03.9 (5.1)	06.2 (6.6)
(2) GPT2-$L_{SCRATCH}$	17.2 (3.1)	19.3 (3.7)	16.8 (2.9)	18.6 (4.5)	17.6 (2.6)	14.4 (3.9)	17.7 (3.2)
(3) BART-FT	15.5 (6.0)	20.8 (3.5)	19.6 (3.5)	19.7 (9.2)	19.2 (3.9)	18.0 (6.6)	11.7 (4.8)
(4) GPT2-FT	30.7 (5.1)	31.3 (5.5)	32.4 (6.3)	30.7 (6.6)	32.3 (5.9)	31.4 (5.8)	31.0 (4.8)
(5) BART-**SIF**	16.8 (5.1)	21.1 (4.2)	19.9 (3.7)	20.5 (11.1)	20.0 (3.8)	19.6 (7.2)	13.7 (5.0)
(6) GPT2-**SIF**	**33.6** (5.4)	**33.9** (5.6)	**35.2** (6.9)	**32.5** (6.9)	**34.2** (5.3)	**33.6** (5.7)	**33.2** (5.5)

- SIF significantly outperforms fine-tuning baselines

SIF Evaluation

- We use SIF to induce script knowledge in GPT2, BART, and T5
- ESDs generated using SIF are evaluated using BLEU (Papineni et al., 2002) metric given multiple reference ESDs for held-out scenarios

Models	TOKENS	EXPECT	SEQUENCE	ALLTOKENS	DESCRIBE	DIRECT	ORDERED
(1) Zero-shot	03.1 (5.2)	03.6 (5.5)	05.4 (2.8)	03.1 (5.2)	03.2 (3.6)	03.9 (5.1)	06.2 (6.6)
(2) GPT2-$L_{SCRATCH}$	17.2 (3.1)	19.3 (3.7)	16.8 (2.9)	18.6 (4.5)	17.6 (2.6)	14.4 (3.9)	17.7 (3.2)
(3) BART-FT	15.5 (6.0)	20.8 (3.5)	19.6 (3.5)	19.7 (9.2)	19.2 (3.9)	18.0 (6.6)	11.7 (4.8)
(4) GPT2-FT	30.7 (5.1)	31.3 (5.5)	32.4 (6.3)	30.7 (6.6)	32.3 (5.9)	31.4 (5.8)	31.0 (4.8)
(5) BART-**SIF**	16.8 (5.1)	21.1 (4.2)	19.9 (3.7)	20.5 (11.1)	20.0 (3.8)	19.6 (7.2)	13.7 (5.0)
(6) GPT2-**SIF**	**33.6** (5.4)	**33.9** (5.6)	**35.2** (6.9)	**32.5** (6.9)	**34.2** (5.3)	**33.6** (5.7)	**33.2** (5.5)

- SIF significantly outperforms fine-tuning baselines
- Script knowledge is best accessible through GPT2 than other LMs

SIF Evaluation

- We use SIF to induce script knowledge in GPT2, BART, and T5
- ESDs generated using SIF are evaluated using BLEU (Papineni et al., 2002) metric given multiple reference ESDs for held-out scenarios

Models	TOKENS	EXPECT	SEQUENCE	ALLTOKENS	DESCRIBE	DIRECT	ORDERED
(1) Zero-shot	03.1 (5.2)	03.6 (5.5)	05.4 (2.8)	03.1 (5.2)	03.2 (3.6)	03.9 (5.1)	06.2 (6.6)
(2) GPT2-$L_{SCRATCH}$	17.2 (3.1)	19.3 (3.7)	16.8 (2.9)	18.6 (4.5)	17.6 (2.6)	14.4 (3.9)	17.7 (3.2)
(3) BART-FT	15.5 (6.0)	20.8 (3.5)	19.6 (3.5)	19.7 (9.2)	19.2 (3.9)	18.0 (6.6)	11.7 (4.8)
(4) GPT2-FT	30.7 (5.1)	31.3 (5.5)	32.4 (6.3)	30.7 (6.6)	32.3 (5.9)	31.4 (5.8)	31.0 (4.8)
(5) BART-**SIF**	16.8 (5.1)	21.1 (4.2)	19.9 (3.7)	20.5 (11.1)	20.0 (3.8)	19.6 (7.2)	13.7 (5.0)
(6) GPT2-**SIF**	**33.6** (5.4)	**33.9** (5.6)	**35.2** (6.9)	**32.5** (6.9)	**34.2** (5.3)	**33.6** (5.7)	**33.2** (5.5)

- SIF significantly outperforms fine-tuning baselines
- Script knowledge is best accessible through GPT2 than other LMs
- Performance across LMs is sensitive to prompt formulation and scenario

Manual Evaluation and Error Analysis

- We manually evaluate generated ESDs at three levels
  - **Individual events**: % of events relevant (**R**) to a scenario (652 ESDs)
  - **Pairwise events**: % of consecutive event pairs correctly ordered (**O,** 582 pairs)
  - **Overall ESD**: degree to which important events are missing on 4-point Likert scale (**M,** 140 ESDs)

Manual Evaluation and Error Analysis

- We manually evaluate generated ESDs at three levels
  - **Individual events**: % of events relevant (**R**) to a scenario (652 ESDs)
  - **Pairwise events**: % of consecutive event pairs correctly ordered (**O,** 582 pairs)
  - **Overall ESD**: degree to which important events are missing on 4-point Likert scale (**M,** 140 ESDs)

Variants	BLEU↑	Manual Evaluation		
		R↑	O↑	M↓
TOKENS	19.2/**22.8**	77.2/**84.3**	72.3/**89.3**	**2.6**/**2.6**
EXPECT	22.8/**26.0**	81.9/**82.7**	74.5/**86.5**	**3.0**/**3.0**
SEQUENCE	27.8/**33.4**	73.3/**83.2**	74.0/**87.5**	**2.5**/**2.5**
ALLTOKENS	33.5/**35.0**	83.5/**85.7**	82.7/**89.5**	**2.6**/**2.6**
DESCRIBE	27.1/**28.6**	80.7/**86.3**	83.9/**85.9**	**2.8**/**2.8**
DIRECT	30.9/**34.1**	81.2/**84.2**	**88.5**/86.1	**2.6**/**2.6**
ORDERED	**31.9**/31.5	84.9/**86.2**	78.6/**86.8**	**2.6**/**2.6**

Boldface: Best between FT/SIF ; Underline: Best across all variants

Manual Evaluation and Error Analysis

- A model can miss significant events, even though it can generate many relevant ones
- We observe repeated paraphrases (4.6% across all prompt variants) of the same event
  - e.g., ‘pour some milk in the pot’ and ‘pour the milk into the coffee pot’ in MAKING COFFEE scenario
- 23.9% of the irrelevant events (13.5% across all prompt variants) are incoherent
  - e.g., ‘take the flat to the bathroom’ for CLEANING A FLAT scenario
- 11.4% mixed events
  - e.g., ‘sit in front of coffee shop’ for MAKING COFFEE scenario
- 61.4% unrelated events
  - e.g., ‘add shampoo’ for WASHING DISHES scenario
- 3.3% ungrammatical events

Conclusion

We presented an analysis of the **extent of script knowledge accessible through LMs** using probing techniques, in a **zero-shot** setting, via the task of generating **full ESDs from natural language prompts**

We **proposed script induction framework (SIF)** that can generate ESDs for held-out and novel scenarios drawn from a different distribution

We presented **automatic and manual evaluation of the generated ESDs**, establishing the viability of our framework and paving way for future research in this direction

GPT2 has incomplete understanding of scripts (irrelevant, missing, repeated and unordered events)

SIF is LM-agnostic and mitigates errors observed in probing experiments (irrelevant, repeated & unordered events)

SIF outperforms FT-based and other baselines as shown via improved BLEU scores and manual evaluation

References

1. Roger C Schank and Robert P Abelson. 1975. Scripts, plans, and knowledge. In IJCAI
2. Risto Miikkulainen. 1995. Script-based inference and memory retrieval in subsymbolic story processing. Applied Intelligence
3. Erik T Mueller. 2004. Understanding script-based stories using commonsense reasoning. Cognitive Systems Research
4. Lilian DA Wanzare, Alessandra Zarcone, Stefan Thater, and Manfred Pinkal. 2016. A crowdsourced database of event sequence descriptions for the acquisition of high-quality script knowledge.
5. Kishore Papineni, Salim Roukos, Todd Ward, and WeiJing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics

Thanks!

Contact: sancheti@umd.edu