Probing Script Knowledge from Pre-Trained Models

Zijia Jin[¶], **Xingyu Zhang**[♮], **Mo Yu**[♣], **Lifu Huang**[♠]
[¶]New York University, [♮]Xi'an Jiaotong University, [♣]WeChat AI, [♠]Virginia Tech
[¶]zj2076@nyu.edu, [♮]xy.zhang@stu.xjtu.edu.cn,
[♣]moyumyu@tencent.com, [♠]lifuh@vt.edu

Abstract

Script knowledge is critical for humans to understand the broad daily tasks and routine activities in the world. Recently researchers have explored the large-scale pre-trained language models (PLMs) to perform various script related tasks, such as story generation, temporal ordering of event, future event prediction and so on. However, it's still not well studied in terms of how well the PLMs capture the script knowledge. To answer this question, we design three probing tasks: *inclusive sub-event selection*, *starting sub-event selection* and *temporal ordering* to investigate the capabilities of PLMs with and without fine-tuning. The three probing tasks can be further used to automatically induce a script for each main event given all the possible sub-events. Taking BERT as a case study, by analyzing its performance on script induction as well as each individual probing task, we conclude that the stereotypical temporal knowledge among the sub-events is well captured in BERT, however the inclusive or starting sub-event knowledge is barely encoded.

1 Introduction

A script is a structure that describes a stereotyped sequence of events that happen in a particular scenario (Schank and Abelson, 1975, 2013). It allows human to keep track of the states and procedures that are necessary to complete various tasks from daily lives to scientific processes. Taking the task of *Eating in a Restaurant* as an example. A classic example script for this task may consist of a chain of subevents, such as *Enter→Order→Eat→Pay (and Tip)→Leave*. The script knowledge has shown benefit to many downstream applications, such as story generation (Li et al., 2013, 2018; Guan et al., 2019; Zhai et al., 2019; Lin et al., 2022), machine reading comprehension (Tian et al., 2020; Ostermann et al., 2018; Sugawara et al., 2018), commonsense reasoning (Ding et al., 2019; Huang et al., 2019; Bauer and Bansal, 2021) and so on.

Recent large-scale pre-trained language models (PLMs) (Devlin et al., 2019; Liu et al., 2019; Radford et al., 2019; Raffel et al., 2019) have shown competitive performance on many natural language processing tasks. Abundant studies have demonstrated that these models either directly capture certain types of syntactic (Goldberg, 2019; Clark et al., 2019; Htut et al., 2019; Rosa and Mareček, 2019), factual (Petroni et al., 2019a, 2020; Bouraoui et al., 2020; Wang et al., 2020) and commonsense knowledge (Zhou et al., 2020; Rajani et al., 2019; Lin et al., 2020) during the pre-training or acquire inductive capability to more efficiently induce such knowledge from natural language text (Pandit and Hou, 2021; Bosselut et al., 2019). However, as another important type of cognitive and schematic knowledge describing human routine activities, scripts are not yet well probed in the language models by prior studies.

To investigate how well the pre-trained language models have captured the script knowledge, in this work, we design three probing tasks and language model prompting methods to probe the script knowledge from PLMs, and further leverage the language model prompting methods to induce the scripts given the main events. Specifically, we aim to answer the following two research questions:

Whether and what script knowledge is captured by the pre-trained language models. To answer this question, we design three sub-tasks to probe the script knowledge, including **inclusive sub-event selection** (i.e., whether a sub-event is included or excluded in a main event or task), **starting sub-event selection** (i.e., which sub-event is the start of the script for a particular main event), and **sub-event temporal ordering** (i.e., predicting a temporal before or after relation between two sub-events). On these sub-tasks, we explore both template-based and soft prompting methods to query the knowledge from pre-trained language models. By investigating their performance gaps to

Proceedings of the Workshop on Unimodal and Multimodal Induction of Linguistic Structures (UM-IoS), pages 87 - 93

the fine-tuning results, we find that both the inclusive and starting sub-event selection sub-tasks have relatively poorer performance than that of temporal ordering, which is likely due to the lack of relevant objectives to encourage the models to capture such knowledge during pre-training, and further suggests future research directions to enhance the PLMs to better capture the script knowledge.

How to better generate the scripts from these pre-trained models. With the language model prompting methods, we can select the inclusive sub-events of a particular script, the starting sub-event and subsequent events by predicting the temporal order among all the inclusive sub-events, which can ultimately generate a sequence of events as the script of a main event. Thus, we further design a benchmark dataset to fine-tune the models for the three sub-tasks and evaluate their performance on generating the whole scripts for various main events from diverse domains and topics.

The contributions of this work can be summarized as follows:

- We are the first to formulate the sub-tasks and set up benchmark datasets to probe the script knowledge from pre-trained language models.
- We are the first to research on the generation and evaluation of the whole scripts from pre-trained language models.

2 Related Work

Script Knowledge The definition of Script Knowledge was first proposed in 1981 (Feigenbaum et al., 1981), which aims to detect the relation between two events. Chambers and Jurafsky (2008) created the first unsupervised data-driven method based on point-wise mutual information (PMI) to automatically extract narrative event chains. Recently, researchers explored deep neural networks, especially large-scale pre-train language models to predict the temporal relation between two events (Pustejovsky et al., 2003; Chambers, 2013; Ferraro and Durme, 2016; Reimers et al., 2016) or generate the future event (Pichotta and Mooney, 2014; Jans et al., 2012; Zhang et al., 2020). Comparing with these studies, our work focuses more on investigating how well the PLMs encode or capture the script knowledge from pre-training and their bottleneck, suggesting possible directions for future research.

Language Model Probing Probing is a popular way to detect what knowledge is encoded in PLMs. At first, probing method is designed for detect morphology knowledge(Belinkov et al., 2017) ,syntactic knowledge (Peters et al., 2018) and semantic knowledge(Tenney et al., 2019). Then researchers began to pay more attention to more complex knowledge like commonsense knowledge. The two main standard approaches in probing commonsense knowledge is building classifiers(Hewitt and Liang, 2019) or filling text in the gap(Petroni et al., 2019b). In our study, we extend the accuracy based methods and designed a series of downstream tasks specific to Scripts Knowledge.

3 Method

3.1 Script Knowledge Probing

Our first goal is to probe the script knowledge from pre-trained language models. To do so, we divide the script knowledge into three categories: the *Inclusive* and *starting* relation between each sub-event and main event, indicating whether the sub-event should be included in or the start of the script of a particular main event, and the *temporal* relation (i.e., *Before* or *After*) among the sub-events. To probe these knowledge from PLMs, we design the following tasks.

Task 1: Inclusive Sub-event Selection As Figure 1 shows, given a main event, e.g., *"Clean laundry"*, and a candidate sub-event, e.g., *"Gather dirty clothes."*, we aim to have the language model to determine whether the sub-event belongs to the script of the target main event. To do so, we use `[MASK]` to connect them into a whole sequence and use a PLM to encode the sequence into contextual representations. In order to predict the *Inclusive* relation, we apply a linear function (i.e., a MLM head) to project the `[MASK]` into a probability distribution over the whole vocabulary of the PLM. By exploring many candidate tokens from the target vocabulary to represent each relation, we finally select "include" to denote the *Inclusive* relation and "except" for *Exclusive*.

Task 2: Starting Sub-event Selection Given a main event and a set of sub-events that are predicted to belong to the script of the main event, we aim to select the most probable sub-event as the start of the script. We formulate it as a sequence classification problem. We concatenate the main event and each sub-event candidate with a prompt "*start with*", e.g., *Taking bus start with finding bus stop*, and use a MLP layer to predict a score indicating how likely

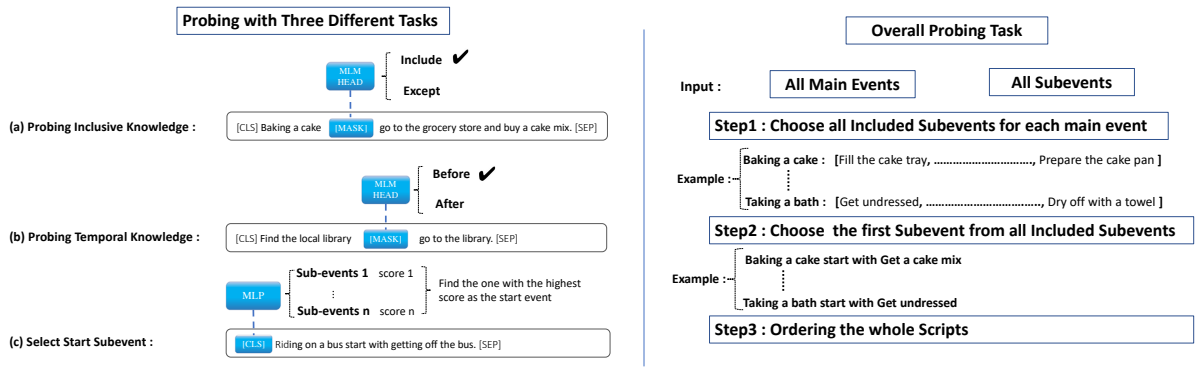

Figure 1: Overview of the probing approaches for (1) Inclusive Sub-event Selection, (2) Starting Sub-event Selection, and (3) Sub-event Temporal Ordering. And an overall evaluation stage for generating scripts with main events and subevents as input.

the sub-event is the start of the script of the main event, based on the contextual representation of the [CLS]. As a result, we use the sub-event with the highest score as the first sub-event. We design a margin based loss function to encourage the score of the positive start sub-event to be higher than others.

$$L(s^*, s_i) = \sum_{\tilde{s}_i \in \tilde{S}} max(score(\tilde{s}_i) + m - score(s^*), 0)$$

where s^* represents the positive start sub-event of a particular script and $\tilde{S}$ denotes the set of other sub-events from the same script. The margin m is a hyper-parameter, which is set as 1.0 in our experiment. During inference, given a set of candidate sub-events, we compare their scores and select the one with the highest score as the starting sub-event.

Task 3: Sub-event Temporal Ordering This probing task is to show the capability of the PLMs on correctly organizing the sub-events into a temporally ordered event sequence. To do so, we design a new language model probing approach following (Petroni et al., 2019c). As shown in Figure 1, given two subevents, e.g., *"put clothes in dryer."* and *"turn on dryer."* , we use [MASK] to connect them into a sequence and use a PLM to encode it. The temporal relation is predicted by comparing the probability of tokens "*before*" and "*after*" based on the contextual representation of [MASK].

3.2 Script Induction with PLMs

The second goal in this work is to design a simple yet effective approach to automatically induce scripts based on PLMs. Given a particular main event and a set of candidate sub-events, to induce the script for the target main event, we design a pipeline approach consisting of three steps: (1) selecting a subset of inclusive sub-events from all the candidates; (2) determining the starting sub-event; and (3) ordering all the inclusive sub-events by predicting the temporal relation between each pair of them. These three steps correspond to the three approaches designed for script knowledge probing.

4 Experiment Setup

We take BERT-base-uncased (Devlin et al., 2019) as the target PLM to investigate how well it encodes the script language via the three probing tasks. We combine three script datasets, including DeScript (Wanzare et al., 2016), OMICS (Gupta and Kochenderfer, 2004) and Stories (Trinh and Le, 2018), where each main event is annotated with 7 to 122 scripts written by different crowd-sourcing workers. We sample 60 main events as the evaluation set, 39 main events as the development set and use the remaining 98 main events for training. For the main events in training and development sets, we keep all the scripts, while for each main event in the evaluation set, we only keep the longest script as the target. Table 1 shows the statistics of each dataset.

Datasets	# Main Events	# Scripts
Training	98	4,685
Development	39	1,791
Test	60	60

Table 1: Data statistics for training, development and evaluation Sets.

To create the training samples for the *inclusive sub-event selection* task, for each script, we use all the ground truth subevents as positive samples and randomly choose 100 times of negative samples

from other main events' scripts. For evaluation, as the inclusive sub-event selection requires a pool of all the possible candidate events, we combine the sub-events of all scripts in the evaluation dataset. To create the training samples for the *start sub-event selection* task, we use the first sub-event of each script as the positive sample and all the remaining sub-events from the same script as the negative samples. During the inference, we select the starting sub-event from the inclusive sub-events predicted by the inclusive sub-event selection approach. We use accuracy as the evaluation metric. Finally, for the temporal ordering task, we create each training sample based on each sub-event together with one of its following sub-events. We randomly shuffle the order of each pair of sub-events and create its corresponding label: *"before"* or *"after"*. To evaluate the quality of the temporal ordering among all the sub-events, we first generate a script based on the predicted temporal order and then use ROUGE-L to evaluate the longest common subsequence between the generated script and the gold script.

We compare the following approaches for each probing task as well as the script induction:

BERT Pre-trained: Directly use the pre-trained BERT model to make the predictions on the evaluation set.

BERT Fine-tuning: Fine-tune BERT with task-specific training data and evaluate those fine-tuned models on the evaluation set.

BERT Ptuning: Following the Ptuning framework (Liu et al., 2021), fine-tune the parameters of both BERT model and prompt tokens.

BERT Ptuning Freeze: Only fine-tune the prompt tokens while freezing the parameters of BERT model.

5 Results and Analysis

5.1 Overall Script Induction

We first show the results of end-to-end script induction given each main event and the pool of all candidate sub-events. As Table 2 shows, without any fine-tuning, BERT-Pretrained can barely induce any reasonable scripts. The high precision and low recall indicates that the bottleneck is likely in correctly selecting the inclusive sub-events for each main event. However, with fine-tuning either on the whole BERT parameters or a few prompt parameters, the script induction performance can be improved significant, demonstrating that the pre-trained BERT actually captures certain level of script knowledge but requires external probes to induce such knowledge from it. Finally, by analyzing of the performance of fine-tuning approaches, we notice a more significant improvement on recall. We conjecture that with fine-tuning, the inclusive sub-event selection is more likely to be improved.

Method	Rouge-L		
	Rec	Prec	F-score
BERT-Pretrained	3.25	22.60	4.81
BERT-Finetuning	37.19	28.07	28.73
BERT-Ptuning	48.70	28.78	32.52
BERT-Ptuning-Freeze	85.16	0.41	0.80

Table 2: Performance of script induction

5.2 Probing on Individual Tasks

We further analyze the capability of BERT on encoding each type of script knowledge based on the three probing tasks. To avoid error propagation, for both starting sub-event selection and temporal ordering, we use the gold inclusive sub-events of each main input as input.

As Table 3 shows, for inclusive sub-event selection, without fine-tuning, both BERT-Pretrained and BERT-Ptuning-Freeze cannot correctly select any inclusive sub-events. This is likely due to the discrepancy between the pre-training objectives of BERT (i.e., MASK language modeling and next sentence prediction) with the objective of inclusive sub-event selection. With fine-tuning, the performance of both BERT-Finetuning and BERT-Ptuning is improved significantly, which is aligned with our assumption in Section 5.1. Starting sub-event selection is hard to all the approaches, which is likely due to two reasons: one is the limited training samples, and the other is that though we formulate each sub-task as mask prediction to better induce the knowledge from BERT, the pattern "*Main_Event starts with Sub_Event*" is less likely to appear in the unlabeled corpus than other patterns, such as "*Main_Event includes Sub_Event*" and "*Event_A before/after Event_B*". Finally, all the approaches show consistently descent performance on temporal ordering, no matter whether BERT is fine-tuned or not, demonstrating that BERT has well captured the relations among the events with stereotypical temporal orders, possibly

Method	Inclusive Subevent Selection			Starting Subevent Selection	Temporal Ordering
	Rec	Prec	F-score	Accuracy	Rouge-L F1
BERT-Pretrained	7.44	0.64	1.17	18.33	63.79
BERT-Finetuning	33.83	44.71	38.51	21.66	62.87
BERT-Ptuning	31.16	56.24	40.10	20.00	63.62
BERT-Ptuning-Freeze	98.69	0.52	1.03	28.33	66.02

Table 3: Performance on each individual task.

due to the next sentence prediction objective during pre-training.

6 Conclusion

In this work, we investigate the capability of large-scale pre-trained language models (PLMs) on capturing three aspects of script knowledge: *inclusive sub-event knowledge*, *starting sub-event knowledge* and *temporal knowledge* among the sub-events from the same script. These three types of knowledge can be further leveraged to automatically induce a script for each main event given all the possible sub-events. We use BERT as a target PLM. By analyzing its performance on script induction as well as each individual probing task, we achieve the conclusions that the stereotypical temporal knowledge among the sub-events is well captured in BERT, however the inclusive and starting sub-event knowledge are not well encoded.

7 Limitations

In this paper, we design a three-stages method to evaluate PLMs' performance in Scripts Knowledge. Although we design those three tasks with pre-prepared candidates as inputs, a more practical condition in real life needs the PLMs to generate scripts from scratch. We plan to use generate models like GPT in the next paper to solve open-domain scripts generation tasks. Moreover, the datasets we used in this paper mostly focused on daily life which not include much scrips knowledge in other domains.

References

Lisa Bauer and Mohit Bansal. 2021. Identify, align, and integrate: Matching knowledge graphs to commonsense reasoning tasks. *arXiv preprint arXiv:2104.10193*.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*.

Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from bert. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7456–7463.

Nathanael Chambers. 2013. Event schema induction with a probabilistic entity-driven model. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1797–1807. ACL.

Nathanael Chambers and Daniel Jurafsky. 2008. Unsupervised learning of narrative event chains. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 789–797. The Association for Computer Linguistics.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*.

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.

Xiao Ding, Kuo Liao, Ting Liu, Zhongyang Li, and Junwen Duan. 2019. Event representation learning enhanced with external commonsense knowledge. *arXiv preprint arXiv:1909.05190*.

Edward A Feigenbaum, Avron Barr, and Paul R Cohen. 1981. The handbook of artificial intelligence.

Francis Ferraro and Benjamin Van Durme. 2016. A unified bayesian model of scripts, frames and language. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016,*

Phoenix, Arizona, USA, pages 2601–2607. AAAI Press.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Jian Guan, Yansen Wang, and Minlie Huang. 2019. Story ending generation with incremental encoding and commonsense knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6473–6480.

Rakesh Gupta and Mykel J. Kochenderfer. 2004. Common sense data acquisition for indoor mobile robots. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence, July 25-29, 2004, San Jose, California, USA*, pages 605–610. AAAI Press / The MIT Press.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 2733–2743. Association for Computational Linguistics.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R Bowman. 2019. Do attention heads in bert track syntactic dependencies? *arXiv preprint arXiv:1911.12246*.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. *arXiv preprint arXiv:1909.00277*.

Bram Jans, Steven Bethard, Ivan Vulic, and Marie-Francine Moens. 2012. Skip n-grams and ranking functions for predicting script events. In *EACL 2012, 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France, April 23-27, 2012*, pages 336–344. The Association for Computer Linguistics.

Boyang Li, Stephen Lee-Urban, George Johnston, and Mark Riedl. 2013. Story generation with crowdsourced plot graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 27.

Zhongyang Li, Xiao Ding, and Ting Liu. 2018. Constructing narrative event evolutionary graph for script event prediction. *arXiv preprint arXiv:1805.05081*.

Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. 2020. Birds have four legs?! numersense: Probing numerical commonsense knowledge of pre-trained language models. *arXiv preprint arXiv:2005.00683*.

Li Lin, Yixin Cao, Lifu Huang, Shuang Li, Xuming Hu, Lijie Wen, and Jianmin Wang. 2022. Inferring commonsense explanations as prompts for future event generation. *arXiv preprint arXiv:2201.07099*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

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*.

Simon Ostermann, Ashutosh Modi, Michael Roth, Stefan Thater, and Manfred Pinkal. 2018. Mcscript: A novel dataset for assessing machine comprehension using script knowledge. *arXiv preprint arXiv:1803.05223*.

Onkar Pandit and Yufang Hou. 2021. Probing for bridging inference in transformer language models. *arXiv preprint arXiv:2104.09400*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. *arXiv preprint arXiv:2005.04611*.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019a. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019b. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019c. Language models as knowledge bases? 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 2019, Hong Kong, China, November 3-7, 2019*, pages 2463–2473. Association for Computational Linguistics.

Karl Pichotta and Raymond J. Mooney. 2014. Statistical script learning with multi-argument events. In *Proceedings of the 14th Conference of the European*

Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden, pages 220–229. The Association for Computer Linguistics.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. 2003. The timebank corpus. In *Corpus linguistics*, volume 2003, page 40. Lancaster, UK.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. *arXiv preprint arXiv:1906.02361*.

Nils Reimers, Nazanin Dehghani, and Iryna Gurevych. 2016. Temporal anchoring of events for the timebank corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Rudolf Rosa and David Mareček. 2019. Inducing syntactic trees from bert representations. *arXiv preprint arXiv:1906.11511*.

Roger C Schank and Robert P Abelson. 1975. Scripts, plans, and knowledge. In *IJCAI*, volume 75, pages 151–157.

Roger C Schank and Robert P Abelson. 2013. *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Psychology Press.

Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. What makes reading comprehension questions easier? *arXiv preprint arXiv:1808.09384*.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Zhixing Tian, Yuanzhe Zhang, Kang Liu, Jun Zhao, Yantao Jia, and Zhicheng Sheng. 2020. Scene restoring for narrative machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3063–3073.

Trieu H. Trinh and Quoc V. Le. 2018. A simple method for commonsense reasoning. *CoRR*, abs/1806.02847.

Chenguang Wang, Xiao Liu, and Dawn Song. 2020. Language models are open knowledge graphs. *arXiv preprint arXiv:2010.11967*.

Lilian D. A. Wanzare, Alessandra Zarcone, Stefan Thater, and Manfred Pinkal. 2016. A crowdsourced database of event sequence descriptions for the acquisition of high-quality script knowledge. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

Fangzhou Zhai, Vera Demberg, Pavel Shkadzko, Wei Shi, and Asad Sayeed. 2019. A hybrid model for globally coherent story generation. In *Proceedings of the Second Workshop on Storytelling*, pages 34–45.

Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020. Reasoning about goals, steps, and temporal ordering with wikihow. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4630–4639. Association for Computational Linguistics.

Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. 2020. Evaluating commonsense in pre-trained language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9733–9740.

A Appendix

A.1 Examples of Errors

In this section, we'd like to use a couple of examples of errors to show that what kind of information are usually being missed by PLMs. We choose 2 scripts as inputs and test BERT's(Without Finetuning) ability to choose the right candidates and order them.