Entity Tracking via Effective Use of Multi-Task Learning Model and Mention-guided Decoding

Janvijay Singh[♣] **Fan Bai**[♣] **Zhen Wang**[♢]
♣ School of Interactive Computing, Georgia Institute of Technology
♢ Department of Computer Science and Engineering, The Ohio State University
iamjanvijay@gatech.edu
fan.bai@cc.gatech.edu
wang.9215@osu.edu

Abstract

Cross-task knowledge transfer via multi-task learning has recently made remarkable progress in general NLP tasks. However, entity tracking on the procedural text has not benefited from such knowledge transfer because of its distinct formulation, i.e., tracking the event flow while following structural constraints. State-of-the-art entity tracking approaches either design complicated model architectures or rely on task-specific pre-training to achieve good results. To this end, we propose **MEET**, a **M**ulti-task learning-**e**nabled **e**ntity **T**racking approach, which utilizes knowledge gained from general domain tasks to improve entity tracking. Specifically, MEET first fine-tunes T5, a pre-trained multi-task learning model, with entity tracking-specialized QA formats, and then employs our customized decoding strategy to satisfy the structural constraints. MEET achieves state-of-the-art performances on two popular entity tracking datasets, even though it does not require any task-specific architecture design or pre-training.[1]

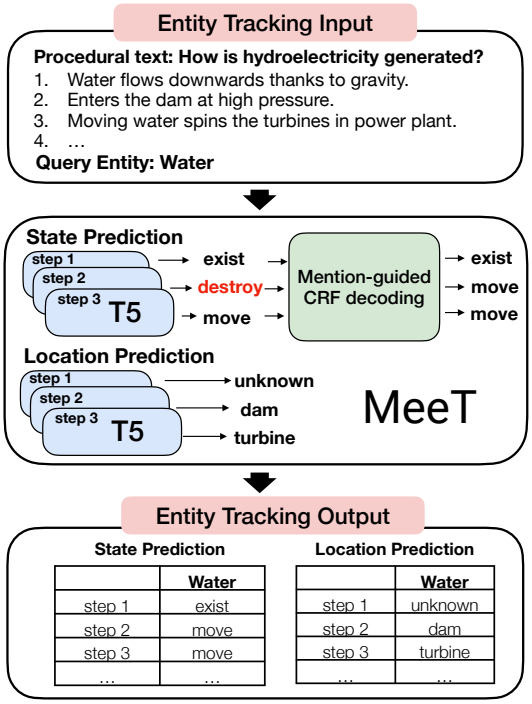

Figure 1: Overview of MEET (**M**ulti-task learning-**e**nabled **e**ntity **T**racking). MEET utilizes the multi-task learning in T5 to boost entity tracking performance, with a customized decoding strategy addressing the structural constraints in state prediction (e.g., "move" cannot happen after "destroy").

1 Introduction

Pre-trained language models have revolutionized the NLP field in recent years (Devlin et al., 2019; Liu et al., 2019; Brown et al., 2020) and also become more versatile with the novel encoder-decoder architecture (Raffel et al., 2020; Lewis et al., 2020), which allows them to handle different types of NLP tasks without further architectural changes. This versatility inherently facilitates cross-task knowledge transfer via multi-task learning (Raffel et al., 2020; Aribandi et al., 2022), and thus helps push the boundary of many popular NLP tasks such as question answering (Khashabi et al., 2020) and semantic parsing (Xie et al., 2022). However, entity tracking, which tracks the states and locations of an entity throughout the procedural text, like scientific processes or recipes, has not been impacted by this multi-task learning wave for two main reasons. First, entity tracking requires the model to make step-wise predictions while satisfying structural constraints (e.g., an entity cannot be "moved" after being "destroyed" in the previous steps). This requirement is usually tackled by designing task-specific architectures (Gupta and Durrett, 2019b; Tang et al., 2020; Huang et al., 2021), and those generic multi-task models with the encoder-decoder architecture cannot address it easily. Second, understanding procedural text requires domain-specific knowledge, which usu-

[1]Our code and data are available at https://github.com/iamjanvijay/MeeT.

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 1255–1263
May 2-6, 2023

ally does not exist in general domain tasks that multi-task learning models are trained on, so it is not clear how effective the knowledge transfer will be given this domain gap (Zhang et al., 2021; Bai et al., 2021; Shi et al., 2022).

In this paper, we study how entity tracking can benefit from the current multi-task learning paradigm and present **MEET**, a **M**ulti-task learning-**e**nabled **e**ntity **T**racking approach. This approach includes two parts. The first part fine-tunes T5 (Raffel et al., 2020), a model that has been pre-trained on a diverse set of NLP tasks and has shown great cross-task generalizability. Here, we design entity tracking-specialized QA formats to accommodate the need to make step-specific predictions, while facilitating effective knowledge transfer from T5. The second part resolves conflicted state predictions under structural constraints. We use a customized offline CRF inference algorithm, where the main idea is to emphasize the predictions of steps, in which the query entity is explicitly mentioned, because the fine-tuned model performs better in those cases (Table 5). On two benchmark datasets, ProPara (Dalvi et al., 2018) and Recipes (Bosselut et al., 2018), our MEET outperforms previous state-of-the-art methods, which require extra domain-specific pre-training or data augmentation. We verify the importance of multi-task learning in T5 and our proposed decoding strategy through careful analyses and ablation studies.

To sum up, our contributions are three-fold: (1) Our work is the first to explore cross-task knowledge transfer for entity tracking on procedural text; (2) Our proposed approach, MEET, effectively uses the off-the-shelf pre-trained multi-task learning model T5 with a customized decoding strategy, and thus achieves state-of-the-art performance on two benchmark datasets; (3) Our comprehensive analyses verify the benefits of multi-task learning on entity tracking.

2 Related Work

Tracking the progression of an entity within procedural text, such as cooking recipes (Bosselut et al., 2018) or scientific protocols (Tamari et al., 2021; Le et al., 2022; Bai et al., 2022), is challenging as it calls for a model to understand both superficial and intrinsic dynamics of the process. Recent work on entity tracking can be divided into two lines. One focuses on designing task-specific fine-tuning architectures to ensure that the model makes step-grounded predictions while following the structural constraints. For instance, Rajaby Faghihi and Kordjamshidi (2021) introduce time-stamp embeddings into RoBERTa (Liu et al., 2019) to encode the index of the query step. Gupta and Durrett (2019b) frame entity tracking as a structured prediction problem and use a CRF layer to promote global consistency under those structural constraints. In our case, we show that, with QA formulation, simply appending the index of the query step to the question and indexing the procedure produces step-specific predictions. Moreover, we propose a customized offline CRF-decoding strategy for structural constraints to compensate for the fact that it is hard to jointly train T5, our backbone LM, with a CRF layer, like in previous methods.

The other line of work focuses on domain-specific knowledge transfer (Zhang et al., 2021; Bai et al., 2021; Shi et al., 2022; Ma et al., 2022). Concretely, LEMON (Shi et al., 2022) achieves great performance by performing in-domain pre-training on 1 million procedural paragraphs. CGLI (Ma et al., 2022) shows that adding high-quality pseudo-labeled data (generated via self-training) during fine-tuning can also boost the model performance. In contrast, our work explores how entity tracking can benefit from out-of-domain knowledge via using off-the-shelf pre-trained multi-task learning models.

3 Method

In this section, we present **MEET**, a **M**ulti-task learning-**e**nabled **e**ntity **T**racking approach. Here, we first review the problem definition, and then lay out the details of MEET.

3.1 Problem Definition

Entity tracking aims at monitoring the status of an entity throughout a procedure. The input of this task contains two items: 1) a procedural paragraph P, composed of a sequence of sentences $\{s_1, s_2, ..., s_T\}$; and 2) a procedure-specific query entity e. Given the input, our goal is to predict the state and location of the query entity at each timestamp of the procedure (see an example from the ProPara dataset in Figure 1).

3.2 MEET

MEET includes two parts, task-specific fine-tuning with our proposed QA formats and the mention-

guided conflict-resolve decoding.

Task-spefic Fine-tuning We formulate the two sub-tasks of entity tracking, state prediction and location prediction, as multi-choice and extractive QA problems respectively (see §4.2 for comparison with other task formulations), and fine-tune T5 to make independent predictions for every step in the procedure. Given a query entity e and procedure P, to predict the entity state at step t, the input sequence is formatted as the concatenation of the template question "*What is the state of* e *in step* t*?*", candidate states (e.g., *create*, *move* and *destroy*), and the full procedure with step index prepended. The output is just one of the candidate states. For location prediction, the input sequence is the concatenation of the question "*Where is* e *located in step* t*?*" and the indexed procedure, with the snippet "*Other locations: none, unknown.*" appended. This is because entity locations sometimes are not explicitly mentioned in the procedure. The output is a text span, indicating the location of the query entity after step t. Examples of both tasks can be found in Appendix A.

Conflict-resolve Decoding Entity tracking places unique structural constraints on state predictions (e.g., *move* cannot happen after *destroy*). Similar to Gupta and Durrett (2019a), we run an offline CRF-decoding method (Viterbi decoding) to resolve conflicting state predictions. We initialize CRF transition scores T with the transition statistics in the training data, following Ma et al. (2022). For example, $T(p, q)$, the transition score between state p and q, is $log(1/10)$ if there is only one $p \Rightarrow q$ transition out of 10 transitions starting with the state p. We set the scores of all unseen transitions to $-inf$. As for CRF emission scores, we use the state prediction logits from T5. In contrast with previous methods, which treat each step equally, we weigh the emission scores differently, depending on whether the query entity e is **explicitly mentioned** in the step:

$$U_i' = \begin{cases} \tau_{exp} \cdot U_i, & \text{if } e \text{ is mentioned in step } i, \\ \tau_{imp} \cdot U_i, & \text{otherwise} \end{cases}$$

where U_i' represents the emission score of step i after weighing, and τ_{exp} and τ_{imp} are hyper-parameters, determined by the grid search on the dev set. The intuition behind our approach is that, as the fine-tuned model performs better on "explicitly mentioned" steps (Table 5), leaning toward those steps during decoding via controlled weights will result in more accurate predictions.[2]

Model	P	R	F1
DYNAPRO (Amini et al., 2020)	75.2	58.0	65.5
TSLM† (Faghini et al., 2021)	68.4	68.9	68.6
KOALA† (Zhang et al., 2021)	77.7	64.4	70.4
LEMON†(Shi et al., 2022)	74.8	69.8	72.2
CGLI† (Ma et al., 2022)	75.7	**70.0**	72.7
MEET (ours)	**80.3**	67.1	**73.1**

Table 1: Test set performance on ProPara. † indicates that the backbone language model has been further pre-trained on either in-domain corpus or auxiliary tasks. MEET performs on par with SOTA models without pre-finetuning on any in-domain corpus.

4 Experiments

Datasets We experiment with two benchmark datasets of entity tracking: ProPara (Dalvi et al., 2018) and Recipes (Bosselut et al., 2018). ProPara contains 488 scientific process-based procedural paragraphs (Figure 1), and Recipes includes 866 cooking recipes. Note that previous work experiments with different splits of the Recipes dataset; in this paper, we follow the split of Zhang et al. (2021)[3] as it is used in most of the recent work (Huang et al., 2021; Shi et al., 2022). More dataset details are presented in Appendix B.

Evaluation ProPara performances are evaluated in two levels: *sentence-level*[4] (Dalvi et al., 2018) and *document-level*[5] (Tandon et al., 2018). Here, we focus on the *document-level* evaluation because it provides a comprehensive assessment of the model's understanding of the overall procedure and serves as the basis for the ProPara leaderboard rankings. The *document-level* evaluation is conducted by comparing the input/output entities and their transformations in the procedure with the gold answers. Further details regarding two evaluations and the result of the *sentence-level* evaluation can be found in Appendix C. For Recipes, following

[2] After hyper-parameter tuning, the optimal values for τ_{exp} and τ_{imp} are 0.6 and 0.7 respectively.

[3] https://github.com/ytyz1307zzh/KOALA/issues/4

[4] https://github.com/Mayer123/CGLI/blob/main/src/evalQA.py

[5] https://github.com/allenai/aristo-leaderboard/blob/master/propara/evaluator

previous work (Zhang et al., 2021; Shi et al., 2022), we evaluate the location changes of each ingredient throughout the recipe.[6]

Baselines For ProPara, we compare MEET with the top five approaches on its leaderboard. Among these five approaches, DYNAPRO (Amini et al., 2020), TSLM (Rajaby Faghihi and Kordjamshidi, 2021), and CGLI (Ma et al., 2022) design task-specific fine-tuning architecture using off-the-shelf LMs while KOALA (Zhang et al., 2021) and LEMON (Shi et al., 2022) develop in-domain LMs for procedural text. For Recipes, as mentioned previously, we compare MEET with methods that experiment on the same data split of Zhang et al. (2021). We refer readers to the corresponding paper of each baseline for further details.

Implementation Details Our approach MEET is implemented using Huggingface Transformers (Wolf et al., 2020). Given the limited computational resources, we choose T5-large as the backbone of our MEET. The fine-tuning process employs the AdamW optimizer with a learning rate of 1×10^{-4} and a batch size of 16. To resolve any potential conflict between state prediction and location prediction, we apply the rules designed in Ma et al. (2022) to integrate the output from both tasks.

4.1 Results

We present the test set results of ProPara and Recipes in Table 1 and Table 2, respectively. Our MEET outperforms the competitive baseline CGLI (Ma et al., 2022) on the ProPara dataset with state-of-the-art performance despite the fact that CGLI uses extra pseudo-labeled training data (generated by self-training) for data augmentation. On Recipes, MEET surpasses the previous best-performing method LEMON (Shi et al., 2022) by a substantial margin of 4.9 F_1. It is noteworthy that the cooking recipes in the Recipes dataset were collected from the web,[7] which may have been included in the C4 corpus[8] used for pre-training T5 and thus potentially contributes to the advantage of our MEET on Recipes.

Model	P	R	F_1
NCET (Gupta and Durrett, 2019b)	56.5	46.4	50.9
IEN (Tang et al., 2020)	58.5	47.0	52.2
KOALA (Zhang et al., 2021)	60.1	52.6	56.1
REAL (Huang et al., 2021)	55.2	52.9	54.1
LEMON (Shi et al., 2022)	56.0	67.1	61.1
MEET (ours)	**64.2**	**78.0**	**66.0**

Table 2: Test set results on Recipes. MEET achieves the state-of-the-art performance, outperforming the previous SOTA LEMON by 4.9 F_1.

4.2 Analysis & Ablation Study

Multi-task Learning To investigate the impact of T5's multi-task learning process on entity tracking, we experiment with two variants of T5 as the backbone of MEET: 1) T5-v1.1,[9] a T5-like LM (with slight architecture changes) whose pre-training does not include any supervised tasks; 2) T5-v1.1$_{QA\text{-}FT}$, the resulting LM after fine-tuning T5-v1.1 on the three QA datasets,[10] which T5 is pre-trained on. The performance of the three LMs (T5-large size) on the ProPara dev set is presented in the top section of Table 3. We can see that T5 outperforms T5-v1.1 by a large margin, verifying that multi-task learning on out-of-domain non-entity-tracking tasks can benefit entity tracking. In addition, the advantage of T5 over T5-v1.1$_{QA\text{-}FT}$ indicates that knowledge transfer can cross the task boundaries with T5's encoder-decoder architecture.

Task Formulation We compare our QA formulation with two other task formulations, proposed in recent work, for T5. The first formulation is called "*step-input*" (Gupta and Durrett, 2019a; Amini et al., 2020), where each pair of the query entity e and procedure step t is formulated as one instance. Here, the state prediction is formulated as a classification problem, where the entity name is appended to the input, and no candidate answers are provided. Moreover, the procedure is trimmed until step t to specify the step index in the input. The second formulation is called "*process-input*" (Zhang et al., 2021; Gupta and Durrett, 2019b), where the model predicts entity states or locations in all steps in one instance. The input is the concatenation of entity e and the full procedure, and the model decodes

[6] https://drive.google.com/drive/folders/1PYGLe7hSoCYfpKmpPumeTy6jmPyONGz4

[7] http://www.ffts.com/recipes.htm

[8] https://www.tensorflow.org/datasets/catalog/c4

[9] https://huggingface.co/docs/transformers/model_doc/t5v1.1

[10] The three datasets include MultiRC (Khashabi et al., 2018), ReCoRD (Zhang et al., 2018), and BoolQ (Clark et al., 2019)

	P	R	F1
MEET (ours)	77.3	71.1	74.1
Multi-task Learning			
T5-v1.1	76.6	64.9	70.3
T5-v1.1$_{QA\text{-}FT}$	76.3	65.8	70.7
Task Formulation			
Process-level	89.3	30.2	45.1
Step-level	76.7	61.3	68.1
Decoding Strategy & Model Size			
CRF-normal	75.0	72.8	73.8
T5-base	76.8	68.2	72.2

Table 3: Analysis and ablation study on ProPara (dev set results). Top: Comparison of different backbone LMs to investigate the impact of multi-task learning. Middle: Comparison of different task formulations. Bottom: Ablation on decoding strategy and model size. Multi-task learning leads to a better entity tracking model, especially with the QA formulation and mention-guided decoding.

entity states and locations in all steps sequentially. The results of two new formulations are presented in the middle of Table 3. Our proposed QA formulation outperforms the other two formulations by a large margin. Detailed analyses of formulation comparison can be found in Appendix D.

Decoding Strategy & Model Size The ablation study on decoding strategy and model size is shown at the bottom section of Table 3. Clearly, our proposed "mention-guided" decoding strategy, as well as using a larger LM as the backbone, contribute to the success of MEET.

5 Conclusion

We presented MEET, a T5-based entity tracking approach. This approach includes our newly proposed QA fine-tuning formats and a customized decoding strategy so that it can effectively encode the flow of events in the procedural text while following structural constraints. The state-of-the-art performances on two benchmark datasets demonstrate the effectiveness of MEET, and further analyses verify that multi-task learning on out-of-domain tasks can be beneficial for entity tracking.

Limitations

This paper demonstrates that multi-task learning on a combination of general domain datasets can effectively improve the model's understanding of the procedural text. However, the precise source dataset responsible for this improvement remains uncertain, making it an avenue for future research to investigate more efficient knowledge transfer through the identification of the most pertinent source dataset. Moreover, the pipeline structure of MEET may limit its practical utilization. As such, future work could consider incorporating our proposed mention-guided decoding strategy into the end-to-end training of the multi-task learning model.

References

Aida Amini, Antoine Bosselut, Bhavana Dalvi, Yejin Choi, and Hannaneh Hajishirzi. 2020. Procedural reading comprehension with attribute-aware context flow. In *Automated Knowledge Base Construction*.

Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q. Tran, Dara Bahri, Jianmo Ni, Jai Gupta, Kai Hui, Sebastian Ruder, and Donald Metzler. 2022. Ext5: Towards extreme multi-task scaling for transfer learning. In *International Conference on Learning Representations*.

Fan Bai, Alan Ritter, Peter Madrid, Dayne Freitag, and John Niekrasz. 2022. SynKB: Semantic search for synthetic procedures. In *Proceedings of the The 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 311–318, Abu Dhabi, UAE. Association for Computational Linguistics.

Fan Bai, Alan Ritter, and Wei Xu. 2021. Pre-train or annotate? domain adaptation with a constrained budget. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5002–5015, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi. 2018. Simulating action dynamics with neural process networks. In *Proceedings of the 6th International Conference for Learning Representations (ICLR)*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Bhavana Dalvi, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. 2018. Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension. 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 1595–1604, New Orleans, Louisiana. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Aditya Gupta and Greg Durrett. 2019a. Effective use of transformer networks for entity tracking. 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 759–769, Hong Kong, China. Association for Computational Linguistics.

Aditya Gupta and Greg Durrett. 2019b. Tracking discrete and continuous entity state for process understanding. In *Proceedings of the Third Workshop on Structured Prediction for NLP*, pages 7–12, Minneapolis, Minnesota. Association for Computational Linguistics.

Hao Huang, Xiubo Geng, Jian Pei, Guodong Long, and Daxin Jiang. 2021. Reasoning over entity-action-location graph for procedural text understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5100–5109, Online. Association for Computational Linguistics.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. 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 252–262, New Orleans, Louisiana. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Nghia T. Le, Fan Bai, and Alan Ritter. 2022. Few-shot anaphora resolution in scientific protocols via mixtures of in-context experts. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2693–2706, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1340–1350, Florence, Italy. Association for Computational Linguistics.

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*.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Eric Nyberg, and Alessandro Oltramari. 2022. Coalescing global and local information for procedural text understanding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1534–1545, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Masaaki Nagata, Katsuki Chousa, and Masaaki Nishino. 2020. A supervised word alignment method based on cross-language span prediction using multilingual BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 555–565, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Hossein Rajaby Faghihi and Parisa Kordjamshidi. 2021. Time-stamped language model: Teaching language

models to understand the flow of events. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4560–4570, Online. Association for Computational Linguistics.

Qi Shi, Qian Liu, Bei Chen, Yu Zhang, Ting Liu, and Jian-Guang Lou. 2022. Lemon: Language-based environment manipulation via execution-guided pre-training.

Ronen Tamari, Fan Bai, Alan Ritter, and Gabriel Stanovsky. 2021. Process-level representation of scientific protocols with interactive annotation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2190–2202, Online. Association for Computational Linguistics.

Niket Tandon, Bhavana Dalvi, Joel Grus, Wen-tau Yih, Antoine Bosselut, and Peter Clark. 2018. Reasoning about actions and state changes by injecting commonsense knowledge. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 57–66, Brussels, Belgium. Association for Computational Linguistics.

Jizhi Tang, Yansong Feng, and Dongyan Zhao. 2020. Understanding procedural text using interactive entity networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7281–7290, Online. Association for Computational Linguistics.

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.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. *EMNLP*.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *ArXiv*, abs/1810.12885.

Zhihan Zhang, Xiubo Geng, Tao Qin, Yunfang Wu, and Daxin Jiang. 2021. Knowledge-aware procedural text understanding with multi-stage training. In *Proceedings of the Web Conference 2021*, WWW '21, page 3512–3523, New York, NY, USA. Association for Computing Machinery.

Wanjun Zhong, Duyu Tang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. A heterogeneous graph with factual, temporal and logical knowledge for question answering over dynamic contexts. *CoRR*, abs/2004.12057.

Dataset	Statistics	Train	Dev	Test	Total
Recipes	# procedures	693	86	87	866
	Avg. steps / proc.	8.8	8.9	9.0	8.8
	Avg. entities / proc.	8.6	8.8	8.5	8.6
ProPara	# procedures	391	43	54	488
	Avg. steps / proc.	6.8	6.7	6.9	6.8
	Avg. entities / proc.	3.8	4.1	4.4	3.9

Table 4: Statistics of Recipes and ProPara.

	P	R	F1
implicit	37.4	23.2	28.3
explicit	**68.3**	**72.4**	**70.2**

Table 5: MEET's sentence-level performance (before applying offline CRF) on *implicit* and *explicit* steps (where the query entity is explicitly mentioned). Clearly, MEET makes more accurate predictions on *explicit* steps.

A Fine-tuning Formats for T5

A.1 State Prediction (Multi-choice QA)

Input:

```
What is the state of water in step 2?
(a) create (b) ... (f) move
step 1: Water flows downawards thanks
to gravity. step 2: Enters the dam at
high pressure. step 3: The moving water
spins the turbines in the power plant ...
step 6: The water leaves the dam at the
bottom.
```

Output:

```
move
```

A.2 Location Prediction (Extractive QA)

Input:

```
Where is water located in step 2?
step 1: Water flows downwards thanks to
gravity. step 2: Enters the dam at high
pressure. step 3: The moving water spins
the turbines in the power plant ... step
6: The water leaves the dam at the bottom.
Other locations: none, unknown.
```

Output:

```
dam
```

B Dataset

For ProPara (Dalvi et al., 2018), following Ma et al. (2022), the state prediction task includes six candidate states (`Outside_Before`, `Create`, `Destroy`, `Move`, `Exist` and `Outside_After`). For Recipes (Bosselut et al., 2018), each ingredient has two possibles states (`Exist` or `Absence`) in each step of the recipe. Full data statistics on two datasets are presented in Table 4.

C Evaluation

Sentence-level evaluation This evaluation measures the following questions for each target *entity*:

- **Cat-1:** Is *entity* created (destroyed, moved) in the process?
- **Cat-2:** When (step #) is *entity* created (destroyed, moved)?
- **Cat-3:** Where (location) is *entity* created (destroyed, moved to/from)?

Further, the F_1 scores of the three questions are aggregated with micro/macro averages.

Document-level evaluation It measures the four questions below for each paragraph:

- What are the *input* entities to the process?
- What are the *output* entities of the process?
- What entity *conversions* occur, when (step #), and where (location)?
- What entity *movements* occur, when, and where?

The macro average of the F_1 scores of these four questions will be used as the final score.

Table 6 provides a comprehensive comparison of past work on the ProPara dataset, including both document-level and sentence-level evaluations.

D Analysis of Formulation Comparison

When compared with the "*step-input*" formulation, the QA formulation allows the model to have the full context, and may take better advantage of LM's pre-training scheme (Li et al., 2019; Nagata et al., 2020). The "*process-input*" formulation works the worst in this comparison. With qualitative analyses, we find that it suffers from error propagation due to its autoregressive decoding, so future work may explore incorporating structural decoding (Tandon et al., 2018) into T5.

Model	Document-level			Sentence-level				
	P	**R**	**F1**	**Cat-1**	**Cat-2**	**Cat-3**	**macro**	**micro**
NCET (Gupta and Durrett, 2019b)	67.1	58.5	62.5	73.7	47.1	41.0	53.9	54.0
IEN (Tang et al., 2020)	69.8	56.3	62.3	71.8	47.6	40.5	53.3	53.0
DYNAPRO (Amini et al., 2020)	75.2	58.0	65.5	72.4	49.3	44.5	55.4	55.5
ProGraph (Zhong et al., 2020)	67.3	55.8	61.0	67.8	44.6	41.8	51.4	51.5
TSLM (Faghini et al., 2021)	68.4	68.9	68.6	78.8	56.8	40.9	58.8	58.4
KOALA (Zhang et al., 2021)	77.7	64.4	70.4	78.5	53.3	41.3	57.7	57.5
REAL (Huang et al., 2021)	**81.9**	61.9	70.5	78.4	53.7	42.4	58.2	57.9
LEMON (Shi et al., 2022)	74.8	69.8	72.2	**81.7**	58.3	43.3	61.1	60.7
CGLI (Ma et al., 2022)	75.7	**70.0**	72.7	80.8	60.7	46.8	**62.8**	**62.4**
MEET (ours)	80.3	67.1	**73.1**	77.5	**61.0**	**49.6**	62.7	**62.4**

Table 6: Document-level and sentence-level evaluation results on ProPara test set.