Coalescing Global and Local Information for Procedural Text Understanding

Kaixin Ma[†], **Filip Ilievski**[§], **Jonathan Francis**[†¶],
Eric Nyberg[†], **Alessandro Oltramari**[¶]
[†]Language Technologies Institute, Carnegie Mellon University
[§]Information Sciences Institute, University of Southern California
[¶]Human-Machine Collaboration, Bosch Research Pittsburgh
{kaixinm, jmf1, ehn}@cs.cmu.edu, ilievski@isi.edu, alessandro.oltramari@us.bosch.com

Abstract

Procedural text understanding is a challenging language reasoning task that requires models to track entity states across the development of a narrative. A complete procedural understanding solution should combine three core aspects: local and global views of the inputs, and global view of outputs. Prior methods considered a subset of these aspects, resulting in either low precision or low recall. In this paper, we propose **C**oalescing **G**lobal and **L**ocal **I**nformation (CGLI), a new model that builds *entity-* and *timestep-aware* input representations (local input) considering the *whole* context (global input), and we *jointly* model the entity states with a structured prediction objective (global output). Thus, CGLI simultaneously optimizes for both precision and recall. We extend CGLI with additional output layers and integrate it into a story reasoning framework. Extensive experiments on a popular procedural text understanding dataset show that our model achieves state-of-the-art results; experiments on a story reasoning benchmark show the positive impact of our model on downstream reasoning. We release our code here: https://github.com/Mayer123/CGLI

1 Introduction

Understanding the causal links of events in procedures is a key aspect of intelligence, facilitating human narration and dialogue. For instance, understanding why story B is plausible and why story A is not (Figure 1) requires procedural understanding of the causes of John leaving his notebook at home, as opposed to him taking out his notebook from his bag: writing in a notebook is counterfactual in the former case, and intuitive in the latter. Understanding stories requires procedural models that can reason consistently about event implications, and do so at different granularities. For a model to decide whether a story is plausible, it has to track the entity states over time, understand the effects of

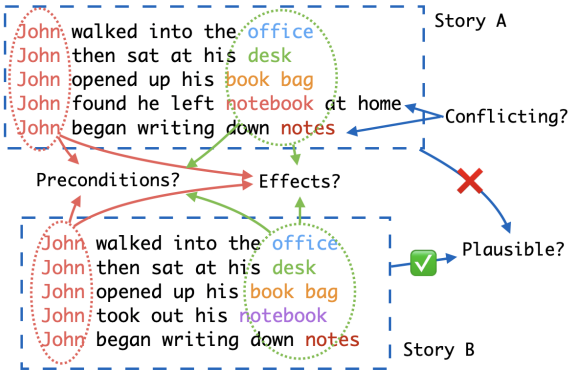

Figure 1: An example story of understanding task. Given two stories, the task is to judge which story is more plausible, find the conflicting sentence pair in the implausible story, and predict entity states at every step.

the described actions (green arrows), and consider the preconditions for a given action (pink arrows). Meanwhile, the model must reconcile the causes and effects of all events described in the story, to provide a globally consistent interpretation.

While procedural reasoning research reports steady progress in recent years (Rajaby Faghihi and Kordjamshidi, 2021; Gupta and Durrett, 2019a; Zhang et al., 2021), story understanding and procedural reasoning have rarely been considered together (Storks et al., 2021). Works have attended only to complementary aspects of the procedural reasoning problem, e.g., Gupta and Durrett (2019a) build entity-centric context representations and ignoring timestep-wise representation modeling; and Rajaby Faghihi and Kordjamshidi (2021) later proposed a timestep-specific model providing unique context encoding at every step to enable modeling flexibility. However, these methods predict independent step-wise entity states, thus compromising the dependency of outputs *across* different steps—yielding high recall but low precision. Global-output methods (Gupta and Durrett, 2019b; Zhang et al., 2021) explicitly leverage the strong

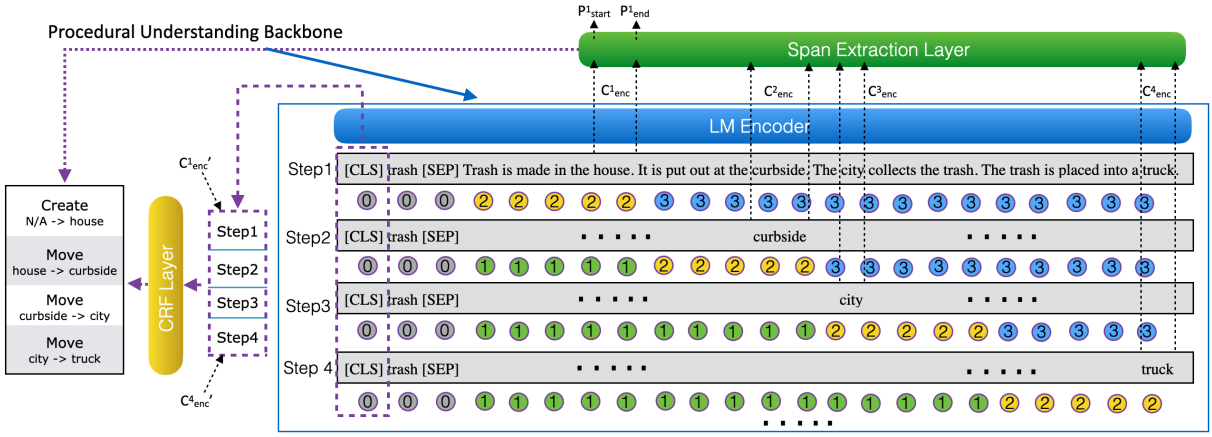

Figure 2: An illustration of CGLI. At every step, the LM encodes the full paragraph with different timestep ids (colored circles with numbers). The span extraction layer yields a location span for every entity at every step and this span sequence is combined with action sequence produced by a CRF layer to form the final predictions.

dependency across steps by jointly modeling the entity actions from all steps, but these methods only have one context encoding for all entities and steps, thus providing sub-optimal input representations—yielding high precision but low recall.

In this paper, we propose **C**oalescing **G**lobal and **L**ocal **I**nformation (CGLI): a new model for procedural text understanding that makes global decisions in consideration of entity-, timestep-centric, and global views of the input. To do so, our model builds a separate input view for every entity, at every step, while providing the whole context. Meanwhile, CGLI represents the entity actions across steps jointly with a structured prediction objective, thus achieving high consistency between different steps. The contributions of this paper are:

1. A novel procedural understanding method, CGLI, which produces global outputs of narrative procedures based on a unified view of the input, combining both local (entity-centric, timestep-specific) and global (document-wide) views—thus optimizing precision and recall, simultaneously.

2. A story understanding framework, which builds upon our procedural understanding model, to enable story understanding with explicit and explainable understanding of event procedures, captured through entity precondition and effect states.

3. An extensive evaluation of CGLI against strong baselines on a procedural task, `ProPara` (Dalvi et al., 2018), and recent story understanding task, `TRIP` (Storks et al., 2021). Our experiments show the positive impact of our method, through achieving state-of-the-art results, while ablation studies measure the impact of its individual components.

2 Task Definition

Procedural text understanding. The task input consists of an n-sentence paragraph $P = \{s_1, s_2, ...s_n\}$, and k entities $\{E_1, E_2, ...E_k\}$. The goal is to predict precondition state $S^p_{i,t}$ and effect state $S^e_{i,t}$, for every entity at every step, as well as the action $A_{i,t}$ performed by the entity at every step; $i \in \{1, 2, ..k\}$, $t \in \{1, 2, ...n\}$. The effect state at $t-1$ is the same as precondition state at step t, i.e., $S^e_{i,t-1} = S^p_{i,t}$, hence S_i is a sequence of length $n+1$. Following prior work (Mishra et al., 2018), $A_{i,t} \in$ {Create, Exist, Move, Destroy}, $S^e_{i,t} \in$ {*non-existence*, *unknown-location*, *location*}, and for *location*, a span of text in P needs be identified for the prediction. Action $A_{i,t}$ describes the entity state changes from precondition to effect, thus it can be inferred from the state sequence S_i, and vice versa—e.g., if $S^p_{i,1} =$ *non-existence* and $S^e_{i,1} =$ *location*, then $A_{i,1} =$ Create.

Procedural story understanding. The input to the procedural story understanding task consists of two parallel stories, P_1, $P_2 = \{s_1, s_2, ...s_n\}$, each consisting of n sentences and differing only in one of the sentences. Following Storks et al. (2021), the task is to identify which story is more plausible, identify the conflicting pair of sentences (s_{c1} and s_{c2}) in the implausible story, and list the preconditions S^e_i and effects S^p_i of all entities at every step of a story. Here, multiple attributes are considered for precondition and effect states. Unlike in the procedural text understanding task, the story completion task does not require that the effect state at step $t-1$ should match the precondition state at step t, i.e., S^e_{it-1} and S^p_{it} are not necessarily equal.

3 CGLI: Coalescing Global and Local Information

In this section, we describe the input representation, the architecture, and the training details of our model, as illustrated in Figure 2.

Input representation. To allow greater modeling flexibility and enable span extraction for entity location-prediction, we build a unique input representation for every entity at each step (*local view*), and we provide it access to the entire context (*global view*). Given an entity, we create a pseudo question Q 'where is {entity}' (*entity-aware*), and concatenate it with the full paragraph P, resulting in C = [CLS] Q [SEP] P [SEP]. We map C using the embedding layer of a language model (LM), resulting in C_{emb}. We then combine C_{emb} with timestep embeddings to mark the current step of interest (*timestep-aware*), following (Rajaby Faghihi and Kordjamshidi, 2021). In particular, each input token is assigned a timestep ID where {0=padding, 1=past, 2=current, 3=future}, forming $T \in \mathbb{R}^m$, where m is the number of tokens. The timestep sequence is projected through another embedding layer $Timestep \in \mathbb{R}^{4\times d}$. The sum of C_{emb} and $Timestep(T)$, denoted with $C'_{emb} \in \mathbb{R}^{d\times m}$, is then processed by the LM encoder layers, where d is the hidden layer dimension of the LM encoder. Formally:[1]

$$C_{emb} = \text{Embed}(C) \quad (1)$$
$$C'_{emb} = C_{emb} + Timestep(T) \quad (2)$$
$$C_{enc} = \text{LM Encoder}(C'_{emb}) \quad (3)$$

Location prediction. Given the LM encoded representation $C_{enc} \in \mathbb{R}^{d\times m}$, we extract the start and end indices of the location span:

$$P_{Start} = \text{Softmax}(W_s C_{enc}) \quad (4)$$
$$P_{End} = \text{Softmax}(W_e C_{enc}), \quad (5)$$

where $W_s, W_e \in \mathbb{R}^d$. For unknown locations and non-existing states, we extract the [CLS] token as the span, analogous to how unanswerable questions are usually handled (Rajpurkar et al., 2018).

In-batch Conditional Random Field. For entity state/action modeling, we jointly predict the entity actions across all steps (*global output*). We first group the encoded representation C^t_{enc} of the same entity at different time steps t in one batch chronologically, yield $C^N_{enc} \in \mathbb{R}^{d\times m\times(n+1)}$. Then we extract the [CLS] token embedding to represent the entity state of every step $C^{N'}_{enc} \in \mathbb{R}^{d\times(n+1)}$. We concatenate the entity state representation of every two consecutive steps to represent the actions between these two-state pairs. The result $D^N_{enc} \in \mathbb{R}^{2d\times n}$ is mapped to the emission scores $\phi \in \mathbb{R}^{a\times n}$, where a is the number of possible actions.

$$D^t_{enc} = \text{Concat}(C^{t'}_{enc}, C^{(t+1)'}_{enc}) \quad (6)$$
$$\phi = W_a^T(tanh(W_d^T D^N_{enc})) \quad (7)$$

where $W_d \in \mathbb{R}^{2d\times d}$, $W_a \in \mathbb{R}^{d\times a}$. The entity action sequence $A \in \mathbb{R}^n$ is modeled by a conditional random field (CRF):

$$P(A|\phi,\psi) \propto \exp(\sum_{t=1}^{n} \phi_t(A_t) + \psi(A_{t-1}, A_t)), \quad (8)$$

with the CRF layer's transition scores $\psi \in \mathbb{R}^{a\times a}$.

Prior initialization. Previous methods (Gupta and Durrett, 2019b; Zhang et al., 2021) initialize the CRF transition scores randomly and update them during training. This allows transition between any pair of actions. However, certain transitions between entity actions are nonsensical, e.g., an entity cannot be destroyed if it has not been created, and a destroyed entity cannot move. Learning such constraints may be possible if we have sufficient data, which is not the case for the tasks we are considering. Thus, we propose to directly impose commonsense constraints on the model's transition scores, because these conditions are universally true and can be used to reduce the model's search space. Specifically, we set an entity action transition score to *-inf* if it has not been seen in the training data, otherwise we estimate the initial score of a transition based on its frequency in the training data: $\psi^{uv} = log(\frac{Num(u,v)}{Num(u)})$, where ψ^{uv} is the log probability of transition from action u to action v, $Num(u,v)$ is the transition count from u to v in data, $Num(u)$ is the count of u in data.

Training and inference. We jointly optimize the location and the entity action prediction losses during training:

$$\mathcal{L}_{loc} = -\frac{1}{n}\sum_{n}^{t=0}(log(P^{y^t_s}_{Start}) + log(P^{y^t_e}_{End})) \quad (9)$$
$$\mathcal{L}_{action} = -log(P(A|\phi,\psi)) \quad (10)$$
$$\mathcal{L} = \mathcal{L}_{loc} + \mathcal{L}_{action}, \quad (11)$$

[1] To model the precondition state of step 1, we also build an input sequence for step 0.

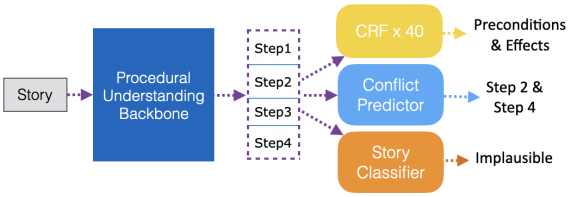

Figure 3: An illustration of integrating CGLI into a story understanding framework. The story is encoded in the same way as shown in Figure 2, producing a sequence of step representations, i.e., a batch of [CLS] vectors. These vectors serve as input to different output layers to model the three task objectives: plausibility (orange), conflict sentence detection (blue), and entity state prediction (yellow).

where y_s^t and y_e^t are the ground-truth start and end indices at step t. During inference, we use Viterbi decoding to produce the most likely entity action sequence and use the span extractor for the most likely location at every step. We combine the action sequence and location predictions to deterministically infer all precondition and effect states.

Data augmentation. Procedural text understanding requires dense annotation of entity states per step, making it challenging and expensive to collect large data. To address data sparsity, we propose a data augmentation method that could effectively leverage the unannotated paragraphs to enhance model's performance. In particular, we first train a model on the gold training set and then apply it to label the unannotated paragraphs, resulting a set of noisy examples. We then mix these examples with gold training data to train a second model.

4 Story Understanding with CGLI

We integrate CGLI into a story understanding framework with minimum modifications following the task definition, and the overall model is shown in Figure 3. As the story understanding tasks do not require location extraction, we remove the span extraction layer, which makes the input representation of step 0 obsolete. Given that the continuity of effects to preconditions between consecutive steps does not hold in this task, we directly use $C_{enc}^{N'} \in \mathbb{R}^{d \times n}$ instead of $D_{enc}^{N} \in \mathbb{R}^{2d \times n}$ in the in-batch CRF. Given B number of attributes for precondition and effect states, we apply an in-batch CRF module for each attribute. Specifically, we apply equations 7 and 8 for every attribute, yielding $2B$ such modules in total. To detect conflicting sentences, we concatenate every pair of sentence representations, and pass it through a linear layer to find the conflicting pair. For story classification, we take the mean of sentence representations for story representation, and pass it through a linear layer for binary classification. Formally,

$$C_{confl} = \text{vstack}(\text{Concat}(C_{enc}^{t'}, C_{enc}^{j'})) \quad (12)$$

$$P_{confl} = \text{Softmax}(W_{confl} C_{confl}) \quad (13)$$

$$C_{plau} = \text{Mean}(C_{enc}^{N'}) \quad (14)$$

$$P_{plau} = \text{Softmax}(W_{plau}^{T} C_{plau}), \quad (15)$$

where $C_{confl} \in \mathbb{R}^{2d \times \frac{n(n-1)}{2}}$, $j \in \{t+1, ...n\}$, $W_{confl} \in \mathbb{R}^{2d}$, $C_{plau} \in \mathbb{R}^{d}$, $W_{plau} \in \mathbb{R}^{d \times 2}$. During training, we jointly optimize all three task objectives:

$$\mathcal{L}_{plau} = -log(P_{plau}^{y_p}) \quad (16)$$

$$\mathcal{L}_{confl} = \begin{cases} -log(P_{confl}^{y_c}) & \text{if } y_p = 0 \\ 0 & otherwise \end{cases} \quad (17)$$

$$\mathcal{L}_{att} = -log(P(S^p|\phi^p, \psi^p)) - log(P(S^e|\phi^e, \psi^e)) \quad (18)$$

$$\mathcal{L} = \mathcal{L}_{plau} + \mathcal{L}_{confl} + \frac{1}{B} \sum_{B}^{b=0} \mathcal{L}_{att}^{b} \quad (19)$$

where y_p=0 if the story is not plausible and y_p=1 if the story is plausible, and y_c denotes the conflict sentence pair index. Note that in our setup, each entity can produce a prediction for conflict sentence pair and story plausibility. At inference time, we take the average of all entities' logits to get the final predictions for these two objectives.

5 Experimental Setup

Benchmarks. We evaluate procedural understanding on `ProPara` (Mishra et al., 2018)[2], which contains 488 human-written paragraphs from the natural science domain. The paragraphs are densely annotated by crowd workers, i.e., for every entity, its existence and location are annotated for every step. Additional 871 unannotated paragraphs are also provided by ProPara. We use these for data augmentation.

We test story understanding on `TRIP` (Storks et al., 2021), which contains crowdsourced plausible and implausible story pairs. In each pair, the plausible story label and the conflicting sentence pair label in implausible story are annotated. TRIP annotates 20 attributes with predefined set of possible values. The annotations are given for all entities at every timestep of the two stories.

[2] `ProPara` is covered under Apache 2.0 License.

Table 1: ProPara test set results. Modeling: E=entity, T=timestep-specific, GC=global context, GO=global outputs.

	Modeling				Sentence-level					Document-level		
Model	E	T	GC	GO	Cat1	Cat2	Cat3	Macro^{avg}	Micro^{avg}	P	R	F1
ProLocal (Dalvi et al., 2018)	Y	Y	N	N	62.7	30.5	10.4	34.5	34.0	**81.7**	36.8	50.7
ProGlobal (Dalvi et al., 2018)	Y	Y	Y	N	63.0	36.4	35.9	45.1	45.4	61.7	48.8	51.9
ProStruct (Tandon et al., 2018)	Y	Y	N	Y	-	-	-	-	-	74.3	43.0	54.5
KG-MRC (Das et al., 2018)	N	Y	N	N	62.9	40.0	38.2	47.0	46.6	64.5	50.7	56.8
NCET (Gupta and Durrett)	N	N	Y	Y	73.7	47.1	41.0	53.9	54.0	67.1	58.5	62.5
IEN (Tang et al., 2020)	N	N	Y	Y	71.8	47.6	40.5	53.3	53.0	69.8	56.3	62.3
DynaPro (Amini et al., 2020)	Y	Y	N	N	72.4	49.3	44.5	55.4	55.5	75.2	58.0	65.5
TSLM (2021)	Y	Y	Y	N	78.8	56.8	40.9	58.8	58.4	68.4	68.9	68.6
KOALA (Zhang et al., 2021)	N	N	Y	Y	78.5	53.3	41.3	57.7	57.5	77.7	64.4	70.4
CGLI (Ours)	Y	Y	Y	Y	80.3	60.5	**48.3**	**63.0**	**62.7**	74.9	**70.0**	72.4
CGLI (Ours) + Data Augmentation	Y	Y	Y	Y	**80.8**	**60.7**	46.8	62.8	62.4	75.7	**70.0**	**72.7**

Table 2: Statistics of the datasets.

Dataset	Stats	Train	Dev	Test
ProPara	#Paragraphs	391	43	54
	#Ents/Para	3.8	4.1	4.4
	#Sents/Para	6.7	6.7	6.9
TRIP	#Paragraphs	1169	474	504
	#Ents/Para	7.0	8.1	8.3
	#Sents/Para	5.1	5.0	5.1

We provide datasets splits details in Table 2. For TRIP, we only report the unique story statistics in Table 2. Note that Storks et al. (2021) have upsampled some of the plausible stories to match the number of implausible stories.

Evaluation metrics. Following previous work, we report both sentence-level metrics[3] and document-level metrics[4] on ProPara. Sentence-level evaluation computes accuracy over three questions: whether the entity created (moved/destroyed) in the process (*Cat1*), and if so, when (*Cat2*) and where (*Cat3*).[5] Document-level metrics compute F1 scores of the identified inputs (entities that exist before the process begins and are destroyed in the process), outputs (entities that do not exist before but exist after the process), conversions (instances where some entities are converted to other entities), and moves (location changes of entities).

For TRIP, we follow the original work and report the following metrics: *accuracy* of classifying the plausible story, *consistency* of finding the conflicting sentence pairs when the story classification is correct, and *verifiability*, which evaluates the prediction of the entities' effects at s_{c1} and the entities' preconditions at s_{c2}. We also report the average F1-score for preconditions and effects across the 20 attributes to better understand the model's procedural understanding ability.

Baselines. For ProPara, we directly report baseline results from the official leaderboard. For TRIP, we report the results from the best model released by Storks et al. (2021).

Training details. For ProPara, we define two additional action types to represent the entity transitions, namely Out-of-Create, Out-of-Destroy similar to (Zhang et al., 2021). Hence, the total size of the action space is six. For evaluation, these two types would be mapped to NONE transition, and they are defined to help the model differentiate the NONE types during training, i.e., if the entity has not being created or if it has been destroyed. To facilitate model's learning on location predictions, we initialized our model with a RoBERTa-Large (Liu et al., 2019) model pretrained on SQuAD 2.0 (Rajpurkar et al., 2018). We run our model five times with different random seeds and report the maximum scores in Table 1 and average scores with a 95% confidence interval in Table 3 and Table 4. For TRIP, we directly initialize the model with RoBERTa-Large. On ProPara we train models for 20 epochs and 6 epochs with data augmentation to let the model receive the similar number of updates. We train models for 10 epochs on TRIP. Except for training epochs, we use the same set of hyperparameters in all of our experiments: learning rate 1e-5, batch size 1, gradient accumulation 2. We used Transformers (Wolf et al., 2020) library [6] for all of our experiments and all of our models

[3] https://github.com/allenai/propara/tree/master/propara/evaluation

[4] https://github.com/allenai/aristo-leaderboard/tree/master/propara

[5] Cat2 and Cat3 only apply to entities that satisfy Cat1.

[6] Covered under Apache 2.0 License.

have about 360M parameters.

Computing infrastructure. We run our experiments on a single Nvidia A6000 GPU or a single Nvidia Titan RTX GPU. For ProPara, each experiment takes about 1.5 hours to finish. For TRIP, each experiment takes about 9 hours to finish.

6 Results and Analysis

6.1 Procedural text understanding

CGLI significantly outperforms all previous baselines on ProPara, achieving state-of-the-art results (Table 1). With data augmentation, our model achieves further improvement on document level. For each baseline, we indicate whether it considers entity-centric information (E), timestep-centric (T), global context (GC), and global output (GO). We note that models that adopt global output usually have much higher precision than recall on document level. On the other hand, TSLM is very good on recall, which is expected given its focus on entity and timestep input modeling.[7] CGLI is able to achieve both strong precision and recall, showing the benefit of global reasoning over both entity- and timestep-specific global inputs in a single model.

We break down the results on ProPara by the document-level question types defined in §5 and compare our best model with the best results reported by TSLM and KOALA. The precision and recall per question type are shown in Figure 4. Consistent with the overall results, KOALA is particularly strong on precision for all types and TSLM is much better on recall. CGLI is able to maintain a balance between those two extremes and achieve overall better results. All three models perform similarly when predicting the inputs and the outputs of a procedure. Yet, CGLI achieves much higher performance on transitional questions regarding entity conversions and moves, which are notably harder to predict. These results suggest that the gains of CGLI over previous works are mostly due to hard-to-answer categories.

6.2 Story understanding

Our method outperforms the baseline method on the TRIP dataset by a very large margin on all metrics, especially on consistency where we observe nearly 400% relative improvement over the baseline (Table 4). This may seem surprising as both our model and the baseline use the same LM backbone. After further analysis of the baseline model, we notice three sub-optimal design decisions. First, the baseline detects conflicting sentence pairs via binary classification for every sentence, independently, without considering pairs of sentences. As a result, for 47.6% of examples in TRIP test set, the baseline model predicted either less or more than two sentences as conflicting, thus getting a score of 0 on consistency. Second, the baseline uses the same encoded representations to directly model both story classification and conflicting pair detection objectives. Without using task-specific output projection layers, the model may be hard to optimize. Third, the baseline did not provide global input view to the model, i.e., each sentence is encoded independently.

6.3 Ablation studies

Impact of modeling aspects To understand the contribution of each of the four modeling aspects we identified for the procedural text understanding, we ablate each of them in CGLI.

No GO is done by removing the CRF layer and directly training the model with cross-entropy loss over the emission probability $\phi \in \mathbb{R}^{n \times a}$ defined in §3. During inference, we predict the action at each timestep independently by taking the argmax over the emission probability instead of viterbi decoding. **No GC** is achieved by allowing the model model to access up to t sentences at every timestep $t \in \{1, 2, 3, ...n\}$, i.e. the model has no access to future sentences. For **No T**, we remove the timestep embeddings such that each entity would have identical encoded context representations across timesteps. For **No E**, we no longer provide the pseudo question with the entity name in the input §3, such that all entities in the same paragraph would have the same encoded context representations.

The results are shown in the bottom half of Table 3. Removing either T or E leads to drastic drop in the F1 score. This is not surprising because the model would have no clue how to distinguish different timesteps or different entities, respectively. We found that the model predict most of entity actions to be NONE, leading to extremely high precision and low recall. Removing GO also leads to a large drop in F1 score, which is actually similar to TSLM's performance, a model that lacks GO. This shows that modeling the global dependency

[7] This pattern may not always hold for other models due to other modeling differences, e.g., LSTM vs. BERT.

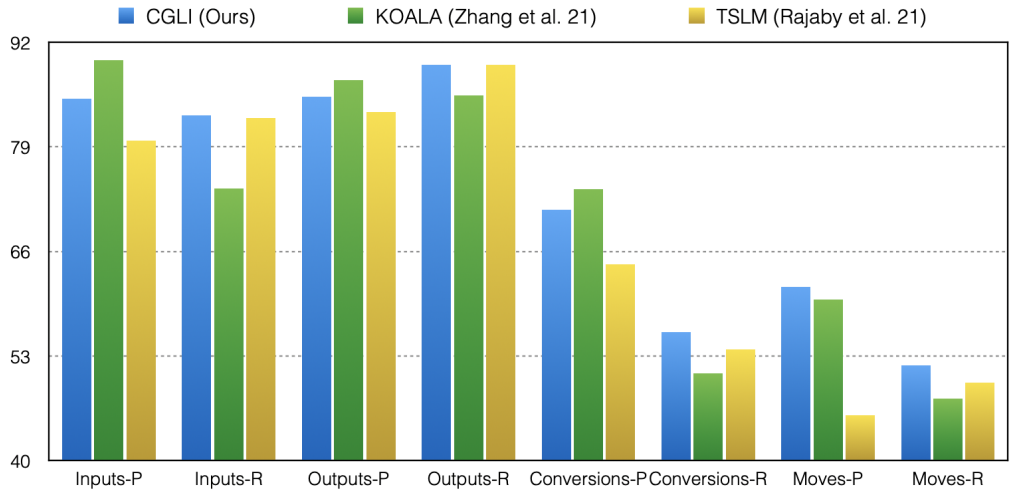

Figure 4: Document-level evaluation on ProPara test set, split by precision (P) and recall (R) per category (Inputs, Outputs, Conversions, Moves).

Table 3: Document-level ablation results of proposed model components and modeling aspects on the ProPara.

	Dev set			Test set		
Model	P	R	F1	P	R	F1
CGLI + Data Augmentation	78.5(±1.7)	76.1(±0.8)	**77.3**(±0.8)	75.2(±1.1)	68.8(±0.8)	**71.9**(±0.5)
CGLI	77.3(±1.5)	75.5(±0.7)	76.4(±1.0)	73.0(±1.9)	**69.8**(±1.2)	71.3(±0.9)
No SQuAD2.0	76.5(±1.3)	75.4(±0.9)	75.9(±0.4)	72.5(±2.7)	68.0(±1.3)	70.1(±0.8)
No Prior	75.6(±0.8)	**76.6**(±0.6)	76.1(±0.3)	72.0(±2.1)	68.1(±1.4)	70.0(±1.3)
No GO	75.7(±1.1)	76.1(±1.4)	75.9(±0.5)	70.2(±1.2)	67.3(±1.2)	68.7(±0.8)
No GC	75.5(±1.3)	73.2(±1.0)	74.3(±0.5)	73.2(±2.2)	66.7(±0.6)	69.8(±1.1)
No T	82.3(±0.7)	59.7(±0.4)	69.2(±0.3)	77.2(±1.3)	54.3(±1.0)	63.8(±0.8)
No E	**84.5**(±1.1)	48.6(±0.3)	61.7(±0.2)	**84.9**(±0.7)	40.8(±0.5)	55.1(±0.3)

Table 4: Results on the TRIP dataset. The F1 scores of last two columns are Macro averages of 20 attributes.

Model	Accuracy	Consistency	Verifiability	Precondition F1	Effect F1
TRIP-RoBERTa (Storks et al., 2021)	73.2	19.1	9.1	51.3	49.3
CGLI (Ours)	93.4(±1.5)	76.3(±1.7)	24.8(±1.6)	70.8(±1.8)	74.9(±1.7)
CGLI (Ours) No CRF	**94.1**(±0.7)	**77.3**(±1.0)	**28.0**(±2.5)	**72.1**(±1.6)	**75.6**(±1.6)

is important for procedural understanding. Finally, removing GC also hurts the performance, which is also expected because location spans often only appear in future sentences, thus span extraction layer is at disadvantage in this setting.

Impact of training data To understand the impact of the CGLI components, we ablate SQuAD2.0 pretraining by initializing the model with vanilla RoBERTa-Large model and we ablate prior initialization by randomly initializing the transition probabilities in the CRF layer. The results (upper half of Table 3) show that with data augmentation, CGLI achieves higher overall F1 scores on average and the gains are mostly from precision. Both pretraining on SQuAD2.0 and prior initialization have a positive impact on the CGLI performance.

As the continuity from effect to precondition states no longer holds on the TRIP story understanding task (cf. §2), we investigate the impact of the CRF layers on modeling entity states. We remove the CRF layers for both effects and preconditions, and we directly train CGLI with regular classification objectives, hence entity states at each step are predicted independently (No GO). Table 4 shows that removing CRF improves performance. We hypothesize that this is caused by the implausible stories in the dataset. Since the entity states in the implausible story's conflicting sentences are inconsistent by nature, training the CRF to maximize their probabilities can be confusing for the model. To verify this, we train models with and without CRF on plausible stories only. In this case, the model is only trained to predict entities effects and preconditions. We found that the models have very

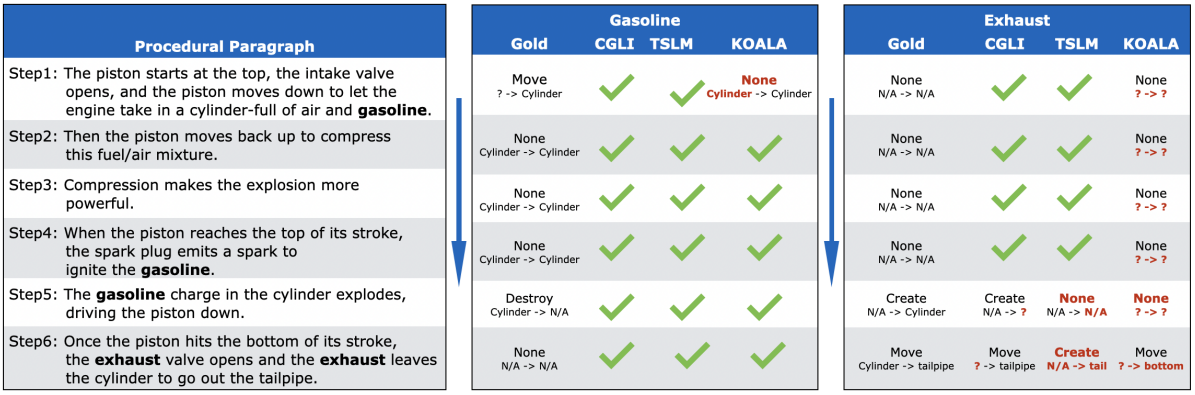

Figure 5: Example predictions on ProPara from three models for two entities. Red font indicate errors.

Table 5: Error Examples on TRIP. The conflicting pairs are marked with *, and the entity of interest with *italic*.

Ann washed her hair in the bathtub.
Ann used the hair dryer to get ready to go out.
Ann applied deodorant to her armpits.
**Ann* put her pants on.
- (Effects, is wet), Pred: False, Gold: Irrelevant
*Ann ironed her *pants* before going out.
- (Preconditions, is wet), Pred: True, Gold: Irrelevant
*John forgot his *notebook* at home.
- (Effects, location), Pred: Moved, Gold: Irrelevant
John sat at his desk.
John opened up his book bag.
* John took out his *notebook*.
- (Preconditions, location),
- Pred: Picked up, Gold: Taken out of container
John began writing down notes.

similar F1 scores with or without CRF (preconditions 74.1 vs 73.7, effects 76.5 vs 76.6). Thus, we conclude that implausible stories are detrimental to CRF training. Moreover, as the effects of the previous step are not a precondition of the current step on TRIP, the outputs from previous steps can hardly contribute to the current prediction, thus CRF has a limited contribution even on the plausible stories.

6.4 Case Studies

We show an example of tracking states for two entities from ProPara with partial outputs from CGLI, TSLM, and KOALA in Figure 5. For gasoline, our model and TSLM both got perfect predictions, but KOALA missed the action at step 1, thus predicting no moves across the process. For exhaust, the sentence in step 6 gives a strong signal for a movement, however, there is no mention of exhaust in the previous steps. Our model is able to infer that *create* needs to come before *move*, thus correctly predicting the actions in steps 5 and 6. However, since TSLM does not have the global output view, it cannot capture such transitions. For KOALA, although it is also able to predict the move and infer that the exhaust should exist before the move, it is unable to predict the create action. We note that for both entities, KOALA is more reluctant to predict actions compared to the other two models. These observations explain why KOALA achieves overall higher precision but lower recall.

We show story reasoning examples from TRIP in Table 5. Since the largest gap in the model performance is between consistency and verifiability, we select examples where our model successfully predicted conflicting sentences but failed to predict entity states. We see that the model still lacks common sense on certain concepts, e.g., forgetting something at home does not result in changing its location, and people usually iron their clothes after they are dry. We also note that some entity states might be hard to distinguish, e.g., the distinction between picking up something versus taking something out of a container only depends on previous location of the object, which might be hard for models to learn from data. These observations suggest that enhancing the model's commonsense reasoning ability is a promising future direction.

7 Related Work

Recent **procedural text understanding** benchmarks including ScoNe (Long et al., 2016), bAbI (Weston et al., 2015), ProcessBank (Berant et al., 2014), ProPara (Mishra et al., 2018), Recipe (Bosselut et al., 2018), and OpenPI (Tandon et al., 2020) have inspired a series of methods. Mishra et al. (2018) propose ProLocal that encodes each step of a procedure separately and ProGlobal that

encodes the full paragraph at every step. KG-MRC (Das et al., 2018) builds a dynamic knowledge graph of entity and location mentions to communicate across time steps. DynaPro (Amini et al., 2020) employs pre-trained LM to jointly predict entity attributes and their transitions. TSLM (Rajaby Faghihi and Kordjamshidi, 2021) formulates procedural understanding as a question answering task, and leverages models pretrained on SQuAD (Rajpurkar et al., 2016) enhanced with a timestamp encoding. Although equipped with various ways to pass information across time steps, these methods still make local predictions thus they may compromise the global dependency of outputs. Another line of work focuses on jointly modeling the entity action sequence, aiming to ensure global structure and consistency. ProStruct (Tandon et al., 2018) aims to find the globally optimal entity action sequence using beam search. Gupta and Durrett (2019b) devise a structured neural architecture NCET, modeled with a CRF, which recurrently updates the hidden representation of each entity at each step. IEN (Tang et al., 2020) builds upon NCET and augments the entity-to-entity attention. KOALA (Zhang et al., 2021) further enhances the NCET by pretraining on Wikipedia and ConceptNet (Speer et al., 2017). The key shortcoming of these global methods is that they rely on entity mentions extracted from a single copy of encoded context shared by all entities and all steps, which limits their modeling capacity. Our proposed method stands out from all previous works by coalescing complementary granularities of procedural text modeling, by building specific and informative input representations while modeling output dependency. Concurrent to our work, Shi et al. (2022) proposed LEMON for language-based environment manipulation. Their focus on model pretraining is orthogonal to CGLI.

There are also numerous recent **story understanding** benchmarks (Mostafazadeh et al., 2016; Qin et al., 2019; Mostafazadeh et al., 2020), and modeling methods (Qin et al., 2020; Guan et al., 2020; Gabriel et al., 2021; Ghosal et al., 2021). The TRIP task (Storks et al., 2021) integrates a procedural understanding component in story understanding to enable consistent and interpretable reasoning over narratives. To our knowledge, we are the first work to bridge the gap of modeling methods between procedural understanding and story comprehension. Other tasks that require reasoning over procedures exist, including defeasible reasoning (Rudinger et al., 2020; Madaan et al., 2021), abductive commonsense inference (Bhagavatula et al., 2019), reasoning over preconditions (Qasemi et al., 2021), script reasoning (Zhang et al., 2020; Sakaguchi et al., 2021) and multimodal script reasoning (Yang et al., 2021; Wu et al., 2021), are typically solved by specialized methods, without separately modeling procedural and causal links. We intend to apply CGLI on these tasks in the future to bridge this gap.

8 Conclusions & Future Work

We proposed CGLI: a novel procedural understanding method that combined global and local information. Recognizing the key role of procedural understanding in downstream tasks, we also integrated CGLI in a story understanding framework. Our experiments showed the benefit of our coalesced method, with the global views providing optimal precision, while the local view boosting its recall, ultimately achieving new state-of-the-art results. We demonstrated that CGLI can help with classifying stories, and identifying the conflicting sentence for inconsistent stories. Future work should investigate how to enhance the commonsense ability of our procedural understanding model, e.g., by injecting commonsense knowledge during finetuning (Chen et al., 2018; Ma et al., 2019) or by pretraining on commonsense knowledge bases (Guan et al., 2020; Ilievski et al., 2021; Ma et al., 2020), and how to apply procedural understanding to other downstream tasks, such as dialogue modelling (Zhou et al., 2021) and planning (Shridhar et al., 2020). Also, it's worth exploring the lightweight-tuning methods (Ma et al., 2021; Vu et al., 2022) to enhance the model's generalization and reduce computation cost.

Acknowledgements

We would like to thank Yonatan Bisk, Aman Madaan, and Ruohong Zhang for helpful discussions and the anonymous reviewers for their valuable suggestions on this paper. Some datasets and models are used by this work, despite their not having specified licenses in their code repositories—for these, we followed previous works and only used them for pure research purposes.

References

Aida Amini, Antoine Bosselut, Bhavana Dalvi Mishra, Yejin Choi, and Hannaneh Hajishirzi. 2020. Procedural reading comprehension with attribute-aware context flow. *CoRR*, abs/2003.13878.

Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, Peter Clark, and Christopher D. Manning. 2014. Modeling biological processes for reading comprehension. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1499–1510, Doha, Qatar. Association for Computational Linguistics.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen tau Yih, and Yejin Choi. 2019. Abductive commonsense reasoning.

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)*.

Jiaao Chen, Jianshu Chen, and Zhou Yu. 2018. Incorporating structured commonsense knowledge in story completion.

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.

Rajarshi Das, Tsendsuren Munkhdalai, Xingdi Yuan, Adam Trischler, and Andrew McCallum. 2018. Building dynamic knowledge graphs from text using machine reading comprehension.

Saadia Gabriel, Chandra Bhagavatula, Vered Shwartz, Ronan Le Bras, Maxwell Forbes, and Yejin Choi. 2021. Paragraph-level commonsense transformers with recurrent memory. In *AAAI*.

Deepanway Ghosal, Navonil Majumder, Rada Mihalcea, and Soujanya Poria. 2021. STaCK: Sentence ordering with temporal commonsense knowledge. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8676–8686, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. 2020. A knowledge-enhanced pretraining model for commonsense story generation. *Transactions of the Association for Computational Linguistics*, 8:93–108.

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.

Filip Ilievski, Jay Pujara, and Hanzhi Zhang. 2021. Story generation with commonsense knowledge graphs and axioms. In *Workshop on Commonsense Reasoning and Knowledge Bases*.

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.

Reginald Long, Panupong Pasupat, and Percy Liang. 2016. Simpler context-dependent logical forms via model projections. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1456–1465, Berlin, Germany. Association for Computational Linguistics.

Kaixin Ma, Jonathan Francis, Quanyang Lu, Eric Nyberg, and Alessandro Oltramari. 2019. Towards generalizable neuro-symbolic systems for commonsense question answering. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 22–32, Hong Kong, China. Association for Computational Linguistics.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Yonatan Bisk, Eric Nyberg, and Alessandro Oltramari. 2020. Knowledge-driven data construction for zero-shot evaluation in commonsense question answering.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Satoru Ozaki, Eric Nyberg, and Alessandro Oltramari. 2021. Exploring strategies for generalizable commonsense reasoning with pre-trained models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5474–5483, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Dheeraj Rajagopal, Peter Clark, Yiming Yang, and Eduard Hovy. 2021. Think about it! improving defeasible reasoning by first modeling the question scenario. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6291–6310, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bhavana Dalvi Mishra, 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. *arXiv preprint arXiv:1805.06975*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Nasrin Mostafazadeh, Aditya Kalyanpur, Lori Moon, David Buchanan, Lauren Berkowitz, Or Biran, and Jennifer Chu-Carroll. 2020. GLUCOSE: GeneraLized and COntextualized story explanations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4569–4586, Online. Association for Computational Linguistics.

Ehsan Qasemi, Filip Ilievski, Muhao Chen, and Pedro Szekely. 2021. Corequisite: Circumstantial preconditions of common sense knowledge.

Lianhui Qin, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark, and Yejin Choi. 2019. Counterfactual story reasoning and generation. 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 5043–5053, Hong Kong, China. Association for Computational Linguistics.

Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena D. Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. 2020. Back to the future: Unsupervised backprop-based decoding for counterfactual and abductive commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 794–805, Online. Association for Computational Linguistics.

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.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Rachel Rudinger, Vered Shwartz, Jena D. Hwang, Chandra Bhagavatula, Maxwell Forbes, Ronan Le Bras, Noah A. Smith, and Yejin Choi. 2020. Thinking like a skeptic: Defeasible inference in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4661–4675, Online. Association for Computational Linguistics.

Keisuke Sakaguchi, Chandra Bhagavatula, Ronan Le Bras, Niket Tandon, Peter Clark, and Yejin Choi. 2021. proScript: Partially ordered scripts generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2138–2149, Punta Cana, Dominican Republic. 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.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

Shane Storks, Qiaozi Gao, Yichi Zhang, and Joyce Chai. 2021. Tiered reasoning for intuitive physics: Toward verifiable commonsense language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4902–4918, Punta Cana, Dominican Republic. 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.

Niket Tandon, Keisuke Sakaguchi, Bhavana Dalvi, Dheeraj Rajagopal, Peter Clark, Michal Guerquin, Kyle Richardson, and Eduard Hovy. 2020. A dataset for tracking entities in open domain procedural text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6408–6417, Online. 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.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou', and Daniel Cer. 2022. SPoT: Better frozen model adaptation through soft prompt transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5039–5059, Dublin, Ireland. Association for Computational Linguistics.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M. Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks.

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.

Te-Lin Wu, Alex Spangher, Pegah Alipoormolabashi, Marjorie Freedman, Ralph Weischedel, and Nanyun Peng. 2021. Understanding procedural knowledge by sequencing multimodal instructional manuals.

Yue Yang, Artemis Panagopoulou, Qing Lyu, Li Zhang, Mark Yatskar, and Chris Callison-Burch. 2021. Visual goal-step inference using wikiHow. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2167–2179, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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)*, pages 4630–4639, Online. Association for Computational Linguistics.

Zhihan Zhang, Xiubo Geng, Tao Qin, Yunfang Wu, and Daxin Jiang. 2021. Knowledge-aware procedural text understanding with multi-stage training. In *WWW '21: The Web Conference 2021, Ljubljana, Slovenia, April 19–23, 2021*.

Pei Zhou, Pegah Jandaghi, Hyundong Cho, Bill Yuchen Lin, Jay Pujara, and Xiang Ren. 2021. Probing commonsense explanation in dialogue response generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4132–4146, Punta Cana, Dominican Republic. Association for Computational Linguistics.