EECS 595 Final Project Proposal - Tiered Reasoning for Intuitive Physics: a Dialogue State Tracking approach

Ikhee Shin
Electrical and Computer Engineering, University of Michigan / ikhee@umich.edu

Abstract

Tiered Reasoning for Intuitive Physics (TRIP) Storks et al. (2021) is recently published. It highlights on underlying reasoning process in addition to end performance. One of the tasks in Storks et al. (2021) is state classification: how does the state of entities change after each sentence of the story? This state changes through the story so, we can apply the Dialogue State Tracking approach. In this project, we want to use a carry-over module to improve state classification.

1 Introduction

Storks et al. (2021) highlights on underlying reasoning process in addition to the end performance of the language model. Storks et al. (2021) claims that a large-scale language model is not good at explaining the result it predicts. To deep dive into this, TRIP has state classification: the model should predict the physical state of entities after each sentence of a story. If the performance of this task is high, it means that the model is explainable. This is very important if the model is applied to real products and the environment.

The objective of Dialogue State Tracking is to track belief states of task-oriented dialogue such as booking hotels. If the domain is booking a hotel, typical belief states are a star, location, and price. Then, the model can use estimated belief states to search entity that satisfies the user's requirement.

Physical state classification in Storks et al. (2021) and Dialogue State Tracking has a few things in common. First, both tasks have a sequence of sentences. TRIP has a story, and the story has sequential sentences with a changing state. Dialogue State Tracking has dialogues between user and system, and after each user utterance, belief states change. Second, states are accumulated for both datasets.

Considering common features of TRIP and Dialogue State Tracking, in this project, we want to apply a carry-over module in Dialogue State Tracking to improve the performance of state classification in TRIP.

2 Related Work

2.1 Tiered Reasoning for Intuitive Physics

TRIP (Storks et al. (2021)) introduced a benchmark dataset that contains not only end task but also reasoning task. It points out that while large-scale pre-trained language models can achieve high end-task performance, they have relatively low performance in the underlying reasoning process.

The TRIP dataset consists of pairs of stories and each story consist of sentences which are actions as Table1. The pair of stories are the same except for one sentence. Given pair of stories, the end task is to determine which story is more plausible. To be plausible, a story should not have conflict. For example, the story B in figure 1 is not plausible because it has conflict. The telephone is unplugged in sentence 2, but it rings in sentence 5.

To understand the underlying reasoning process of the end task, TRIP has 2 additional tasks. The first task is detecting conflicting sentences. For example, the model should notice that the second and 5th sentences in Table 1 have conflict. Another task is state prediction. Each sentence of the story contains actions. So, it modifies the physical states of the attribute (e.g. telephone). State prediction contains precondition, effect, and state prediction.

In the experiment of Storks et al. (2021), a large-scale pre-trained model achieved 71% for the end task, while it has relatively low performance on state prediction.

One important thing about state prediction is that the state does not change frequently. In the story example of Table 1, the state of attribute telephone changes in the second sentence. Except for that, it does not change. Table 3 has statistics of state change. There are two operation types: carry and

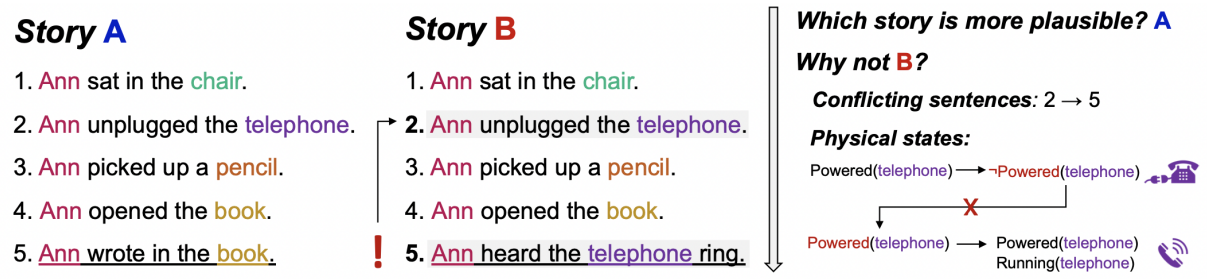

Figure 1: Story example from TRIP

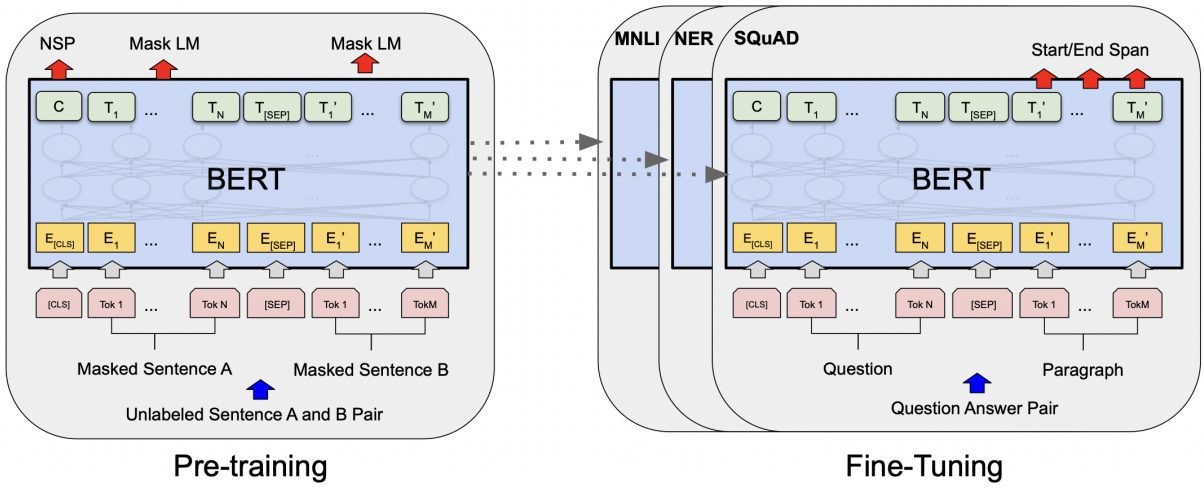

Figure 2: BERT architecture from Devlin et al. (2018)

update. Given attributes, if all states of attributes don't change, it is carried. If at least the state of attribute changes, it is updated. We can see that for 71% of sentences, there are no state changes.

2.2 Language Models

Large scale pretrained language models (Devlin et al. (2018), Liu et al. (2019)) outperforms previous approaches such as recurrent networks (Staudemeyer and Morris (2019), Schuster and Paliwal (1997) and Chung et al. (2014)).

The pre-trained model was first widely used in computer vision. Deng et al. (2009) is the large-scale image dataset with classes annotated. The CNN-based models (He et al. (2015)) are pre-trained using this dataset by predicting classes of images. Then, many works fine-tune this model to their specific task. This approach is useful since usually, there is not enough for the task. By pre-training on a large-scale dataset, the model can learn general features which can be used by a downstream task.

Devlin et al. (2018) used the encoder part of Vaswani et al. (2017) to pretrain the large scale language model. After this work, a lot of variants (Brown et al. (2020), Yang et al. (2019), Clark et al. (2020) and Lan et al. (2019)) are developed and pretrained language models become main stream and it is even applied to image domain (Parmar et al. (2018)).

To pre-train the model using the image dataset, images with annotation are needed. But, Devlin et al. (2018) used self-supervised learning to alleviate this issue. The text data is crawled from the internet (e.g. Wiki) and for training data, some portions of the text are replaced with Mask token and the training objective is to predict the value of these positions. The model should predict tokens of these positions by inspecting context information. This loss is called MLM (Masked Language Model). The importance of MLM loss is re-examined in several works (e.g. Mehri et al. (2020)) where the author claims that using MLM loss during finetuning stage improves the performance.

By MLM loss, pre-trained language models learn semantic information. Similar to pre-trained

User:	I need to book a hotel in the east that has 4 stars.
Hotel	area=east, stars=4
Agent:	I can help you with that. What is your price range?
User:	That doesn't matter if it has free wifi and parking.
Hotel	parking=yes, internet=yes price=dontcare, stars=4, area=east
Agent:	If you'd like something cheap, I recommend Allenbell
User:	That sounds good, I would also like a taxi to the hotel from cambridge
Hotel	parking=yes, internet=yes price=dontcare, area=east, stars=4
Taxi	departure=Cambridge destination=Allenbell

Figure 3: Dialogue example from Budzianowski et al. (2018)

models in image domain, many tasks (Rajpurkar et al. (2016), Sang and Meulder (2003)) finetune pre-trained encoder to specific task as Figure 2. For the TRIP task, only the CLS token is used for finetuning.

2.3 Dialogue State Tracking

The Objective of DST (Dialogue State Tracking) is to track the belief states of given task-oriented dialogue. Tracked belief states can be used by downstream modules (e.g. hotel booking bot, chatbot).

Task-oriented dialogue is a dialogue between user and agent that contains a goal of the user. Figure 3 is a example of task-oriented dialogue from Budzianowski et al. (2018) and Table 2 is corresponding states. The task is booking a hotel and the belief states are the descriptions of the hotel (e.g. star, location). These belief states are called slots. Slots are predefined for each task. And the DST model should fill the value for slots for each sentence of dialogue.

As seen in table2, not all slots are updated in each sentence of dialogue. For example, the first sentence updates two slots while other slots don't change. This is intuition that Gao et al. (2019), Kim et al. (2020) and Heck et al. (2020) applied carry over module. For each slot, the carry-over module determines whether the value of the slot should be updated. If an update is not needed, it carries overs the state of the previous turn. And if an update is needed slot value prediction module predicts the value of the slot.

3 Proposed Approach

Gao et al. (2019) and Kim et al. (2020) applied Carry over module to Dialogue State Tracking task. It is efficient when the state does not change frequently. Table 1 is one example of the TRIP dataset. The second sentence of the story changes the state of the entity telephone. But, after the state is modified in sentence 2, it does not vary. Also, we can notice that most of the state does not change during the story.

Taking advantage of this fact, we can add a Carry Over module that determines whether the state changes or not. Table3 defines the operations of the Carry Over module. Since there are carry and update, the Carry Over module is a binary classification module. If the result of the Carry Over module is updated, the next state is classified.

3.1 Contextual Encoder

We used BERT and RoBERTa encoder as a contextual embedding model. For each attribute, an attribute is concatenated to each sentence of a story. And then, it is fed into the encoder. That is, each sentence is fed into the encoder the number of attribute times. Inputs to the BERT encoder are previous sentence, current sentence, and current state. For example, to predict $S_{t=2}$ in Table 1, inputs are

$$A_2 = \texttt{telephone}$$

Story	State
1. Ann sat in the chair	`Powered(telephone)`: True
2. Ann unplugged the telephone	`Powered(telephone)`: False
3. Ann picked up a pencil	`Powered(telephone)`: False
4. Ann opened the book	`Powered(telephone)`: False
5. Ann heard the telephone ring	`Powered(telephone)`: False

Table 1: Story example from TRIP

Dialogue	State
U: I need to book a hotel in the east that has 4 stars	`area`:east, `stars`:4
A: I can help you with that. What is the price range?	`area`:east, `stars`:4
U: That doesn't matter if it has free wifi and parking	`area`:east, `stars`:4,`parking`:yes,`internet`:yes
A: If you'd like something cheap, I recommend Allenbell	`area`:east, `stars`:4,`parking`:yes,`internet`:yes
U: Than sounds good.	`area`:east, `stars`:4,`parking`:yes,`internet`:yes

Table 2: Dialogue example from Multiwoz

$D_2 =$ `Ann unplugged the telephone`

and target output is

$S_2 =$ `Powered(telephone) : True`

A_t and D_t are tokenized using the tokenizer of BERT. And then, each $A_t - D_t$ is fed into encoder. After that, the Embedding of the CLS token can be used for downstream tasks.

3.2 Loss function

The loss function is the same as the TRIP approach except for carry-over module loss. Carry Over module loss is binary cross-entropy loss.

3.3 Dataset

We used the TRIP dataset. The objective of this project is to improve the state classification module.

4 Evaluation

Table4 is the evaluation result. By applying Carry Over module, the performance of Accuracy increased while that of Consistency and Verifiability decreased.

5 Discussion

Applying Carry Over module to attribute state classification didn't improve Consistency and Verifiability. In TRIP, attribute state is first classified and this information is used to predict precondition and effect. What I expected was by improving attribute state prediction, the performance of precondition and effect prediction also increases. But, it was not. The performance of attribute state prediction might be not directly related to that of precondition and effect.

Kim et al. (2020) reported that improving the carry-over module is crucial since whether to update or not depends on that module. That is, if the result of the carry-over module is wrong, the error is propagated. I used a simple fully connected layer for the carry-over module for this experiment. But, it might be better to use more complicated architecture such as LSTM. Now, not only carry-over but also precondition and effect are predicted sentence by sentence independently. If these are predicted sequentially, the performance might increase.

6 Conclusion

Applying the carry-over module didn't improve the Consistency and Verifiability performance. The size of the dataset is not that huge. So we can try to apply the below methods.

First, we can consider applying the Multi-task learning approach. If there is a similar inference task, we can train the model using both TRIP and task.

Second, we can consider the data augmentation method. this can be text level or story level. Text level (Feng et al. (2021)) augmentation modifies each sentence. For example, we can shuffle each sentence. Story level augmentation (Li et al. (2020)) modifies structure of story. It should be careful since modification in story level can make a given story not plausible.

Operation Type	definition	# Operations
`carry`	All states don't change	119172 (71%)
`update`	At least one state changes	48618 (29%)

Table 3: Operation definition

Model	Carryover	Accuracy (%)	Consistency (%)	Verifiability (%)
RobBERTa	N	74.6	24.8	8.8
RobBERTa	Y	78.6	22.8	7.7
BERT	Y	73.8	13.1	2.8

Table 4: Evaluation

7 Github link

https://github.com/ikhee0119/Verifiable-Coherent-NLU/tree/dev-carry

- experiment with experiment.ipynb

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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. *CoRR*, abs/2005.14165.

Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *CoRR*, abs/1810.00278.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. *CoRR*, abs/2003.10555.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard H. Hovy. 2021. A survey of data augmentation approaches for NLP. *CoRR*, abs/2105.03075.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tür. 2019. Dialog state tracking: A neural reading comprehension approach. *CoRR*, abs/1908.01946.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. Trippy: A triple copy strategy for value independent neural dialog state tracking. *CoRR*, abs/2005.02877.

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient dialogue state tracking by selectively overwriting memory. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 567–582.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

Shiyang Li, Semih Yavuz, Kazuma Hashimoto, Jia Li, Tong Niu, Nazneen Fatema Rajani, Xifeng Yan, Yingbo Zhou, and Caiming Xiong. 2020. Coco: Controllable counterfactuals for evaluating dialogue state trackers. *CoRR*, abs/2010.12850.

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. *CoRR*, abs/1907.11692.

Shikib Mehri, Mihail Eric, and Dilek Hakkani-Tür. 2020. Dialoglue: A natural language understanding benchmark for task-oriented dialogue. *CoRR*, abs/2009.13570.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, and Alexander Ku. 2018. Image transformer. *CoRR*, abs/1802.05751.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *CoRR*, cs.CL/0306050.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Ralf C. Staudemeyer and Eric Rothstein Morris. 2019. Understanding LSTM - a tutorial into long short-term memory recurrent neural networks. *CoRR*, abs/1909.09586.

Shane Storks, Qiaozi Gao, Yichi Zhang, and Joyce Chai. 2021. Tiered reasoning for intuitive physics: Toward verifiable commonsense language understanding.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.