Multi-Task Self-Supervised Learning for Script Event Prediction

Bo Zhou[1,2], Yubo Chen[1,2], Kang Liu[1,2], Jun Zhao[1,2], Jiexin Xu[3], Xiaojian Jiang[3], Jinlong Li[3]
[1]School of Artificial Intelligence, University of Chinese Academy of Sciences
[2]National Laboratory of Pattern Recognition, CASIA, [3]China Merchants Bank
{bo.zhou,yubo.chen,kliu,jzhao}@nlpr.ia.ac.cn,{jiexinx,jiangxiaojian,lucida}@cmbchina.com

ABSTRACT

Most existing approaches to script event prediction rely on manually labeled data heavily, which is often expensive to obtain. To cope with the training data bottleneck, we investigate methods of combining multiple self-supervised tasks, i.e. tasks where models are explicitly trained with automatically generated labels. We propose two self-supervised pre-training tasks: one is End Identification and the other is Contrastive Scoring. Multi-task learning framework is then leveraged to combine these two tasks to jointly train the model. The pre-trained model is then fine-tuned using human-annotated script event prediction training data. Experimental results on the commonly used dataset show that our approach can achieve competitive performance compared to the previous models which are trained with the whole dataset by using just 10% of the training data, and our model trained on the whole dataset outperforms previous models significantly.

CCS CONCEPTS

• **Computing methodologies** → **Information extraction**;

KEYWORDS

Script Event Prediction, Self-Supervised Learning, Multi-Task Learning

ACM Reference Format:
Bo Zhou[1,2], Yubo Chen[1,2], Kang Liu[1,2], Jun Zhao[1,2], Jiexin Xu[3], Xiaojian Jiang[3], Jinlong Li[3]. 2021. Multi-Task Self-Supervised Learning for Script Event Prediction. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3459637.3482150

1 INTRODUCTION

Understanding events in text is important to many natural language processing applications, such as dialog generation [21] and discourse understanding [17]. One of the most challenging tasks in this line is script event prediction [2]. As shown in Figure 1, the purpose of this task is to predict the most suitable subsequent event from a candidate list given a chain of narrative events (context).

Previous work on script event prediction [7, 13, 20] relies on manually labeled data heavily, which is often expensive to obtain and the data size is limited. To cope with the training data bottleneck, we investigate the self-supervised learning method in this paper. Self-supervised learning aims to train a model on auxiliary tasks with automatically generated labels. By pre-training on these tasks, the model can learn useful representations and perform well on downstream supervised tasks with less human-annotated data. Inspired by the successful application of self-supervised learning in NLP [4, 12, 16, 19], we want to design self-supervised tasks which will match the goal of script event prediction as much as possible and then help to improve the performance of the task. Thus, we propose two self-supervised tasks called **E**nd **I**dentification (EI) and **C**ontrastive **S**coring (CS) for script event prediction. The EI task enables the model to judge the actual ending of a narrative text which is also helpful for script event prediction since it aims to choose the most reasonable ending event from a candidate list. The CS task endows the model with the ability to differentiate a proper narrative text ending from an improper ending, which matches the goal of script event prediction.

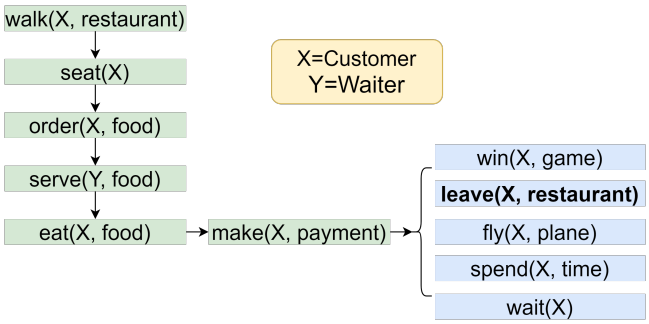

Figure 1: An example of script event prediction. Given a chain of narrative events (context), the purpose of this task is to predict the most suitable (marked in bold) subsequent event from a candidate list.

To help the script event prediction as much as possible, we hope that the discrepancy between the distributions of self-supervised data and gold script event prediction data is as small as possible. As a result, we base our self-supervised tasks on the resources called TimeTravel [18] because it captures a rich set of causal and temporal commonsense relations between daily events, which is similar to script event prediction data. We also point out that other choices of self-supervised data and self-supervised tasks are possible, which we leave as future work.

The first part of TimeTravel comprises narrative stories of five sentences. Given a story in the EI task, we first create other four stories by swapping the fifth sentence with the first to fourth sentences respectively. Then the model is trained to identify the actual ending, i.e. the fifth sentence.

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482150

The second part of TimeTravel consists of original narrative stories and their rewritten versions. Each original story has five sentences and the rewritten one has the same first sentence and different remaining four sentences. In CS task, given an original story and its corresponding rewritten story, we first swap the endings of the two stories to create a negative original story and a negative rewritten story, then the model is trained to make the scores of the original story and its corresponding rewritten story higher than their corresponding negative ones.

We then leverage multi-task learning framework to combine these two tasks to train the model jointly. Then the pre-trained model is fine-tuned with human-annotated script event prediction training data. Our contributions can be summarized as follows:

- We propose two self-supervised tasks for script event prediction to handle the training data bottleneck. To the best of our knowledge, we are the first to investigate self-supervised learning in script event prediction.
- Based on the two self-supervised tasks, we further leverage multi-task framework to combine the two self-supervised tasks.
- Experimental results show that our approach can achieve competitive performance compared to the previous models which are trained with the whole dataset by using 10% of the training data. Our method trained on the whole dataset significantly outperforms previous methods.

2 METHODOLOGY

The proposed method is illustrated in Figure 2. We first jointly train the model by the two self-supervised tasks. Then the model is fine-tuned on the human-annotated script event prediction training data. Next we introduce the two self-supervised tasks and the fine-tuning process in detail.

2.1 Self-Supervised Learning Task

End Identification. Given a story from the first part of TimeTravel $s = (s_1, s_2, s_3, s_4, s_5)$, we first create other four stories by swapping the fifth sentence with the first to fourth sentences respectively:

$$\begin{cases} s^1 = (s_5, s_1, s_2, s_3, s_4) \\ s^2 = (s_1, s_5, s_2, s_3, s_4) \\ s^3 = (s_1, s_2, s_5, s_3, s_4) \\ s^4 = (s_1, s_2, s_3, s_5, s_4) \end{cases} \tag{1}$$

For each story we create these additional four stories, then the model is trained by these stories to identify the fifth sentences. We then treat the EI task as a multi-classification problem of five classes. Given a story s^i, a feature vector $\mathbf{h}_i$ is obtained by sending s_i into an encoder. A linear layer and softmax function will be used on $\mathbf{h}_i$ to obtain a classification probability vector $\mathbf{p}_i$ and cross-entropy is used as the classification loss:

$$\mathcal{L}_{EI} = -\frac{1}{N}\sum_{i=1}^{N} \mathbf{l}_i \cdot \log \mathbf{p}_i \tag{2}$$

where N is the number of stories and $\mathbf{l}_i$ is the label vector for the input story.

Contrastive Scoring. We note that in the second part of TimeTravel, the original story and its rewritten version sometimes have the same fifth sentences. In order to avoid the false negative problem in CS task, we drop the fifth sentences and keep the first four sentences. Thus given an original story $o = (s_1, s_2, s_3, s_4)$ and its rewritten version $r = (s_1, s'_2, s'_3, s'_4)$, we first create a negative story for o by replacing s_4 with s'_4, i.e. $on = (s_1, s_2, s_3, s'_4)$ and a negative story for r by replacing s'_4 with s_4, i.e. $rn = (s_1, s'_2, s'_3, s_4)$. Then o, on, r and rn are input to the encoder to get the corresponding hidden feature vectors. For o and on, we obtain $\mathbf{h}_o$ and $\mathbf{h}_{on}$ and these two vectors are input to the following loss [8] to make the score of o higher than that of on:

$$\mathcal{L}_1 = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{\mathbf{w}\cdot\mathbf{h}_o^i + b}}{e^{\mathbf{w}\cdot\mathbf{h}_o^i + b} + e^{\mathbf{w}\cdot\mathbf{h}_{on}^i + b}} \tag{3}$$

where N is the number of original stories, $\mathbf{w}$ and b are learnable parameters. Similarly, for r and rn, we obtain $\mathbf{h}_r$ and $\mathbf{h}_{rn}$ which are input to a loss the same as equation (3) to get $\mathcal{L}_2$. Finally, we have a loss for CS task:

$$\mathcal{L}_{CS} = \mathcal{L}_1 + \mathcal{L}_2 \tag{4}$$

2.2 Multi-Task Pre-training

Since both the EI and CS tasks match the goal of script event prediction, we hope that combining EI and CS can further help the script event prediction task and thus we leverage multi-task learning to jointly train the model:

$$\mathcal{L}_{pretraining} = \mathcal{L}_{EI} + \mathcal{L}_{CS} \tag{5}$$

We pre-train our model with the two self-supervised tasks EI and CS by minimizing the loss $\mathcal{L}_{pretraining}$.

2.3 Script Event Prediction Fine-tuning

After pre-trained on the two self-supervised tasks of EI and CS, the model is then fine-tuned with human-annotated script event prediction training data. As shown in Figure 1, given a context of event chain $(e_1, e_2, ..., e_{n-1})$, the task is to predict the most suitable subsequent event e_n from a candidate list. Formally, an event e is represented by $v(a_0, a_1, a_2)$ where v is the verb describing the event, a_0 is its subject, a_1 is its direct object and a_2 is its indirect object. Sometimes an event may have less than four components.

When fine-tuning the model, we first represent an event by (a_0, v, a_1, a_2), then we get a sequence of context $(a_0^1, v^1, a_1^1, a_2^1, ..., a_0^{n-1}, v^{n-1}, a_1^{n-1}, a_2^{n-1})$. Each of the five candidate events is also represented as (a_0, v, a_1, a_2) and appended to the context. As shown in Figure 2, these five sequences will be input to the same encoder as in self-supervised pre-training to obtain five feature vectors $\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3, \mathbf{h}_4, \mathbf{h}_5$, then we get a classification probability vector $\mathbf{p}$:

$$\mathbf{p} = \text{softmax}(\mathbf{w}\cdot\mathbf{h}_1 + b, ..., \mathbf{w}\cdot\mathbf{h}_5 + b) \tag{6}$$

where $\mathbf{w}$ and b are the same learnable parameters as equation (3). Then the cross-entropy is used as the classification loss:

$$\mathcal{L}_{finetuning} = -\frac{1}{M}\sum_{i=1}^{M} \mathbf{l}_i \cdot \log \mathbf{p}_i \tag{7}$$

where M is the number of event contexts and $\mathbf{l}_i$ is the label vector for the context indicating which of the candidate events is correct. We fine-tune our model by minimizing the loss $\mathcal{L}_{finetuning}$.

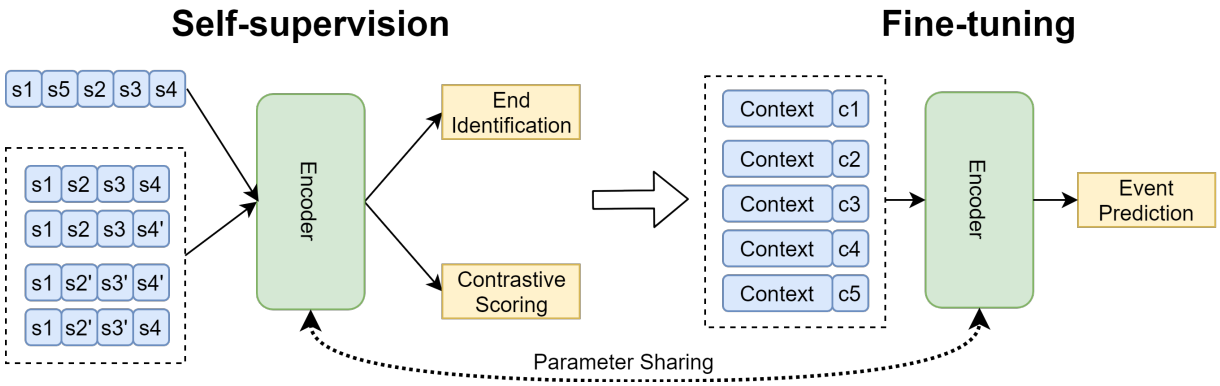

Figure 2: Illustration of our proposed method. The model is first jointly trained by the two self-supervised tasks in the multi-task framework and then fine-tuned using human-annotated script event prediction training data.

3 EXPERIMENT

3.1 Experimental Setup

Dataset. In the script event prediction task, event chains are extracted from the New York Times portion of the Giga-word corpus the same as previous work [7]. We use C&C tools [3] for POS tagging and dependency parsing, and OpenNLP is used for coreference resolution and phrase structure parsing. The detailed data statistics are shown in Table 1.

	Training	Development	Test
Documents	830,643	103,583	103,805
Chains	140,331	10,000	10,000

Table 1: Data statistics of the script event prediction task.

The two self-supervised tasks are based on TimeTravel [18]. After creating another four stories for each story in the first part of TimeTravel, we get 484k stories in total for the EI task. For the CS task, we obtain 113k stories in total from the second part of TimeTravel.

Training Details. We use BERT-base [4] as the encoder. For the self-supervised tasks, the batch size is 40, and 20% of the stories are used for the CS task while 80% of the stories are for the EI task in a mini-batch. The learning rate is 2e-5 and the model is trained for 3 epochs. For fine-tuning the model, the batch size is set to 16 and the learning rate is 2e-5. We use Adam [11] to optimize the model for both self-supervised tasks and the fine-tuning task.

3.2 Overall Performance

We compare our self-supervised model with previous state-of-the-art models in Table 2. Our model outperforms all the previous models and achieves a 62.31% accuracy, which is 3.65% higher than the state-of-the-art. We attribute the success of our model to the two self-supervised tasks which both endow the model with the ability to differentiate a proper ending from an improper ending.

Surprisingly, our model achieves competitive performance compared to the previous models which are trained with the whole dataset by using just 10% of the training data, which shows that our self-supervised method can substantially reduce the need for human-annotated data which is often expensive to obtain.

Models	Accuracy (%)
PMI [2]	30.52
BiGram [9]	29.67
EventComp [7]	49.57
SGNN [13]	52.45
PairLSTM [20]	55.12
SAM-Net [14]	54.48
Lv et al. [15]	58.66
Our self-supervised (10% data)	55.25
Our self-supervised (Whole data)	**62.31**

Table 2: Comparison with previous state-of-the-art models. 10% data means fine-tuning our model on 10% of the training data of script event prediction task and whole data means using all the training data.

Method	Accuracy (%)
BERT-base	58.61
+Contrastive Scoring	61.89
+End Identification	61.64
+Multi-Task	**62.31**

Table 3: Ablation study over the two self-supervised tasks.

3.3 Ablation over the two Self-Supervised Tasks

We study the effect of the two self-supervised tasks in Table 3. Both the EI and CS achieve higher performances than the original model which doesn't conduct self-supervised task. This demonstrates that both our self-supervised tasks are helpful for the script event prediction task. Higher performance is achieved when we jointly train the two self-supervised tasks by multi-task learning, which demonstrates that combining the two tasks can further help the script event prediction task.

3.4 Varying Amounts of Supervised Data

We explore how fine-tuning scales with human-annotated data size, by varying the amount of human-annotated training data the model has access to. We plot Accuracy with respect to the amounts of human-annotated script event prediction data for fine-tuning in Figure 3. Our self-supervised model improves performance considerably when limited gold human-annotated training data is available, but those gains diminish with more high-quality human-annotated data. Using only 30% of the labeled data, our approach already achieves comparable performance than the previous state-of-the-art method using 100% of the human-annotated training data, demonstrating that our self-supervised tasks are particularly useful on small datasets.

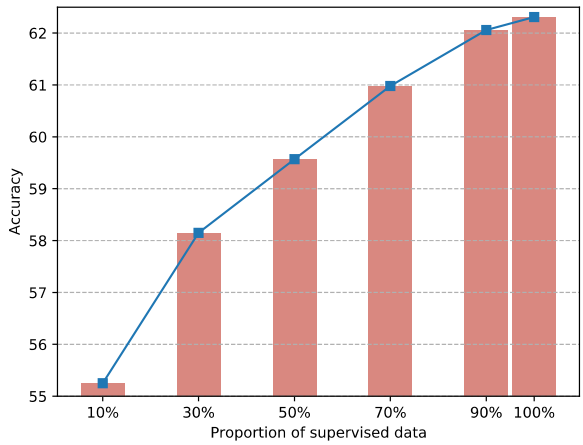

Figure 3: The impact of supervised training data size to script event prediction.

3.5 Error Analysis

The error of our model is mainly caused by two reasons:

The first is that some events in the chain contain too little information to predict the correct subsequent event: an event is represented by four components $v(a_0, a_1, a_2)$, and sometimes an event may have less than four components. For quantitative measurement, we analyze the widely used New York Times corpus for script event prediction task and show the statistics in Figure 4. Events with only two components accounted for 28.88% of the total number of events, and there are even 0.16% of events with only one component. What's worse, some events contain noises due to the automatic construction of the dataset, which makes predicting the correct subsequent event more challenging.

The second kind of error comes from the discrepancy between the distributions of gold script event prediction data and the self-supevised data. Although the two self-supervised tasks match the goal of script event prediction task, the discrepancy between the data distributions of the self-supervised tasks and the supervised task will still introduce some errors.

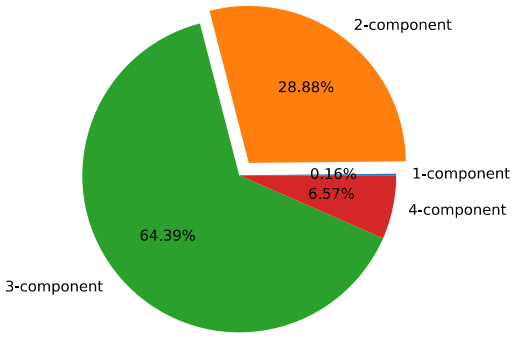

Figure 4: Proportion of events containing different numbers of components in the dataset.

4 RELATED WORK

Script Event Prediction. Script event prediction was first proposed by Chambers and Jurafsky [2], they proposed a statistical model to learn the cooccurrence between events for predicting the subsequent event. Subsequently, Granroth-Wilding and Clark [7] presented a neural network that simultaneously learns word embedding and composition function. In order to integrate order information and event relation, an LSTM based model [20] was proposed. Different from the above work, Li et al. [13] constructed event graph instead of event chains for the task of script event prediction.

Self-Supervised Learning. A variety of self-supervised learning tasks [5, 6, 10] have been proposed in computer vision domain and achieved impressive results. Self-supervised learning is also used in natural language processing domain, for example, to learn word embedding [16] and language model [1]. Tasks such as next sentence prediction [4] and sentence order prediction [12] are proposed to learn better representations that greatly improve the performances of downstream tasks. Motivated by the success of self-supervised learning, we propose two self-supervised tasks to help the model better differentiate a proper ending from an improper ending for script event prediction.

5 CONCLUSION

In this paper, we propose two self-supervised tasks and leverage multi-task learning to jointly train the model for script event prediction to handle the training data bottleneck. Experimental results show that our model trained with 10% of the training data can achieve competitive performance compared to the previous models which are trained with the whole dataset. Further, our model trained on the whole dataset outperforms previous methods (trained with the whole dataset) significantly.

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China (No.2020AAA0106400), and this work is supported by the National Natural Science Foundation of China (No.61976211, No.61806201). This work is also supported by Beijing Academy of Artificial Intelligence (BAAI2019QN0301).

REFERENCES

[1] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The journal of machine learning research* 3 (2003), 1137–1155.

[2] Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*. 789–797.

[3] James R Curran, Stephen Clark, and Johan Bos. 2007. Linguistically motivated large-scale NLP with C&C and Boxer. In *Proceedings of the 45th annual meeting of the Association for Computational Linguistics Companion volume proceedings of the demo and poster sessions*. 33–36.

[4] 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)*. 4171–4186.

[5] Carl Doersch, Abhinav Gupta, and Alexei A Efros. 2015. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*. 1422–1430.

[6] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. 2018. Unsupervised Representation Learning by Predicting Image Rotations. In *International Conference on Learning Representations*.

[7] Mark Granroth-Wilding and Stephen Clark. 2016. What happens next? event prediction using a compositional neural network model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30.

[8] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 297–304.

[9] Bram Jans, Steven Bethard, Ivan Vulić, and Marie Francine Moens. 2012. Skip n-grams and ranking functions for predicting script events. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. 336–344.

[10] Dahun Kim, Donghyeon Cho, Donggeun Yoo, and In-So Kweon. 2018. Learning Image Representations by Completing Damaged Jigsaw Puzzles. In *IEEE Winter Conference on Applications of Computer Vision*. IEEE TPAMI TC, 793–802.

[11] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*.

[12] 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. In *International Conference on Learning Representations*.

[13] Zhongyang Li, Xiao Ding, and Ting Liu. 2018. Constructing narrative event evolutionary graph for script event prediction. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 4201–4207.

[14] Shangwen Lv, Wanhui Qian, Longtao Huang, Jizhong Han, and Songlin Hu. 2019. SAM-Net: Integrating event-level and chain-level attentions to predict what happens next. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 6802–6809.

[15] Shangwen Lv, Fuqing Zhu, and Songlin Hu. 2020. Integrating External Event Knowledge for Script Learning. In *Proceedings of the 28th International Conference on Computational Linguistics*. 306–315.

[16] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2*. 3111–3119.

[17] Allen Nie, Erin Bennett, and Noah Goodman. 2019. DisSent: Learning sentence representations from explicit discourse relations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 4497–4510.

[18] 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)*. 5046–5056.

[19] Shaolei Wang, Wangxiang Che, Qi Liu, Pengda Qin, Ting Liu, and William Yang Wang. 2020. Multi-task self-supervised learning for disfluency detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 9193–9200.

[20] Zhongqing Wang, Yue Zhang, and Ching Yun Chang. 2017. Integrating order information and event relation for script event prediction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 57–67.

[21] Xianchao Wu, Ander Martinez, and Momo Klyen. 2018. Dialog generation using multi-turn reasoning neural networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 2049–2059.