Effect Generation Based on Causal Reasoning

Feiteng Mu[1], Wenjie Li[1], Zhipeng Xie[2]
[1]The Department of Computing, The Hong Kong Polytechnic University, Hong Kong
[2]School of Computer Science, Fudan University, Shanghai, China
csfmu,cswjli@comp.polyu.edu.hk,xiezp@fudan.edu.cn

Abstract

Causal reasoning aims to predict the future scenarios that may be caused by the observed actions. However, existing causal reasoning methods deal with causalities on the word level. In this paper, we propose a novel event-level causal reasoning method and demonstrate its use in the task of effect generation. In particular, we structuralize the observed cause-effect event pairs into an event causality network, which describes causality dependencies. Given an input cause sentence, a causal subgraph is retrieved from the event causality network and is encoded with the graph attention mechanism, in order to support better reasoning of the potential effects. The most probable effect event is then selected from the causal subgraph and is used as guidance to generate an effect sentence. Experiments show that our method generates more reasonable effect sentences than various well-designed competitors.

1 Introduction

Causal reasoning is the process of observing an action and reasoning future scenarios that may be potentially caused by it (Radinsky et al., 2012). Earlier causal reasoning methods (Roemmele et al., 2011; Luo et al., 2016) collect causally related word pairs (e.g., *earthquake→tsunami*) to build the statistical models of causality, and then predict effects words for given cause words. Recently, (Xie and Mu, 2019) uses causal embedding to predict possible effect words of the input causes. (Li et al., 2020) proposed the lexically-constrained beam-search to generate possible effects given provided word guidance. However, all these methods tend to reason causalities at word-level.

Causalities between word pairs are not always self-contained (i.e., intelligible) when they are extracted without the context (Hashimoto et al., 2014)). For example, "*quarrel→break*" is not self-contained since this is not intelligible without the context: "*They always quarrel→They break up*". A word-level causal reasoning method may only predict the unintelligible effect of "*break*" conditioned "*quarrel*". Considering this deficiency, a better way is to enhance causal reasoning with causal events (Radinsky et al., 2012; Zhao et al., 2017; Martin et al., 2018; Ammanabrolu et al., 2020). However, an observed causal event is very likely to appear only once, which brings about huge sparsity to causalities and great difficulty to the event-level causal reasoning. To solve this problem, we structuralize observed causal events into an event causality network, where similar events are clustered together. Given an input cause sentence, a causal subgraph is retrieved and is encoded with the graph attention mechanism, in order to support better effect reasoning. As such, we are able to predict the most reasonable effect event based on the event causality network. The predicted effect event contains the skeleton information, with the detailed context neglected in the event extraction process. So we further rewrite the predicted effect event to an effect sentence in order to fill in the missing information.

The contributions of this paper are twofold: i) we devise a effect generation method which is based on causal event reasoning (EGCER) to generate effect sentences for given input cause sentences, ii) experiments demonstrate that our model achieves better performances compared among various well-designed baselines.

2 Event Causality Network Construction

In this paper, we use causal events to bridge the causalities between input sequences and generated sequences. Hence, we must first collect sufficient cause-effect sentence pairs so that from each sentence pair a cause-effect event pair can be eventified. Then we construct an event causality network based on the extracted causal event pairs. The construction process includes two steps: 1) Event Eventification, 2) Events Structuralization.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 527–533
November 7–11, 2021.

Event Eventification: Following (Do et al., 2011; Asghar, 2016; Luo et al., 2016; Hassanzadeh et al., 2019), we make use of a few high-precision causal connectives to extract cause-effect sentence pairs, for example *'because'*, *'as a result'*, etc. Then we extract causal event pairs from causal sentence pairs based on dependency analysis[1]. We adopt the commonly used 4-tuple event representation (s, v, o, m) (Pichotta and Mooney, 2016) where v denotes the verb, s denotes the head noun of the subject, o denotes the head noun of the direct object or the adjective, and m denotes the head noun of the prepositional or indirect object.

Events Structuralization: We structuralize the extracted causal event pairs into an event causality network, in which semantically similar events are clustered together. We use event abstractions to judge whether two events are semantically similar. The abstraction of an event is obtained by generalizing its components to their categories in linguistic resources. Specifically, the verb in each event is generalized to its class in VerbNet (Schuler, 2005). The other components are generalized by the WordNet (Miller, 1995) synset two levels up in the inherited hypernym hierarchy. In addition, we explicitly use the semantic-similarity based inferring rule. For example, assume we have observed that A has the same abstraction with B, and a causal relation holds from A to C, then it is most likely to conclude that there may be a causal relation from B to C. Such a manipulation significantly reduces the sparsity of causalities in the event causality network, and hence supports better reasoning about the effect events. The weight of an edge in our event causality network is derived by the following rules:

1) If the edge between the event pair (e_i, e_j) is extracted from the dataset, the weight w_{ij} of this edge is $w_{ij} = 1$;

2) If the edge of (e_i, e_j) is inferred based on the semantic-similarity between (e_i, e_k) and the causal relation between (e_k, e_j), we have $w_{ij} = sim(e_i, e_k)$, where $sim(e_i, e_k)$, calculated by the path-similarity measure in WordNet, is the semantic-similarity score between e_i and e_k.

3 Effect Generation

Task Discription: The goal of effect generation consists of predicting the an effect event for the input cause and rewriting the predicted effect event into an effect sentence. Formally, given a cause sentence $X = \{x_1 x_2 \cdots x_m\}$, and a causal subgraph $CG = \{e_1, e_2, \cdots, e_{N_{CG}}\}$, which consists of a set of events $\{e_j = (s_j, v_j, o_j, m_j)\}$ $(j = 1, \cdots, N_{CG})$ as nodes, our model first predicts an effect event e_Y from CG according to X, then rewrites e_Y to an effect sentence $Y = \{y_1 y_2 \cdots y_n\}$. The overview of the proposed EGCER is illustrated in Figure 1, which consists of two modules: 1) Effect Event Predictor, and 2) Effect Event Rewriter.

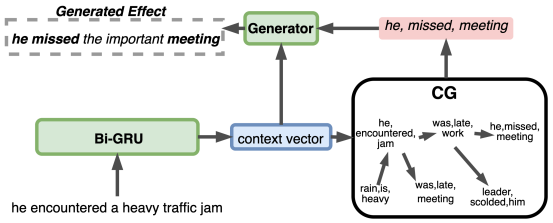


Figure 1: The overview of EGCER.

Effect Event Predictor: Given the cause sentence X, a bidirectional GRU model (Cho et al., 2014) is used to reads the sequence X from both directions and computes hidden states $\overrightarrow{\mathbf{h}_{x_i}}$ and $\overleftarrow{\mathbf{h}_{x_i}}$ for the token x_i. The final hidden vectors of X is $\mathbf{H}_X = \{\mathbf{h}_{x_1}, \cdots, \mathbf{h}_{x_m}\}$, where $\mathbf{h}_{x_i} = [\overrightarrow{\mathbf{h}_{x_i}}; \overleftarrow{\mathbf{h}_{x_i}}]$.

We then eventify the cause event from X, and match the event abstraction in the event causality network. Once the abstraction is matched, a L-hop causal-related subgraph CG is preserved. The neighborhood information in CG represents the causality tendencies, which are useful for reasoning the most reasonable effect event. We use a simple graph neural network (GNN) (Kipf and Welling, 2016; Veličković et al., 2017) to capture the neighborhood information. Specifically, the l-th layer's vectors of e_i and its neighbors are pooled to obtain the vector of e_i on the $(l+1)$-th layer with a activation function σ (ReLU by default):

$$\mathbf{z}_i^l = \mathbf{W}^l \mathbf{e}_i^l$$
$$\mathbf{e}_i^{l+1} = \sigma\Big(\sum_{j=1}^{N_{CG}} \frac{\exp(w_{ij}(\mathbf{z}_i^{l\mathrm{T}} \cdot \mathbf{z}_j^l)}{\sum_k \exp(w_{kj}(\mathbf{z}_k^{l\mathrm{T}} \cdot \mathbf{z}_j^l)} \mathbf{z}_j^l\Big), \quad (1)$$

where $\mathbf{W}^l$ is a parameter, $\cdot$ denotes the inner product of the two vectors, w_{ij} is the weight of the edge (e_i, e_j), $\mathbf{e}_i^l$ is the vector of e_i at l-th layer, $\mathbf{e}_i^0 = [\mathbf{e}_{s_i}; \mathbf{e}_{v_i}; \mathbf{e}_{o_i}; \mathbf{e}_{m_i}]$ is the concated word embedding of all components of e_i.

The final hidden vector $\mathbf{e}_i^L$ $(i = 1, \cdots, N_{CG})$ of events are used to select the guided effect event

[1] https://spacy.io/

e_Y by $e_Y = \max_i cs_i$, where $cs_i = \mathbf{e}_i^{LT} \cdot h_X$ is the causal score between each candidate event e_i and X, $\mathbf{h}_X = \frac{1}{m}\sum_{k=1}^{m} \mathbf{h}_{x_k}$ is the mean-pooling representation of X.

Effect Event Rewriter: The predicted e_Y contains the skeleton information, we want retain all tokens of e_Y when generating the effect sentence to avoiding the causal information carried by e_Y degrading to word-level. Inspired by (Mou et al., 2016; Martin et al., 2018), we rewrite $e_Y = (s, v, o, m)$ into the effect sentence which conforms to the format of $[\_s][\_v][\_o][\_m]$, where blanks indicate the place words should be added to in order to make a sentence richer in content. We use a decoder with attention mechanism (Bahdanau et al., 2014) to generate words in each blank until generating the "<eos>" token.

4 Experiments

4.1 Datasets

English Wikipedia(Enwiki): We extract cause-effect sentence pairs from the English Wikipedia corpus[2], resulting in about 80K pairs. We split all pairs into training/validation/test with the ratio of 8:1:1, and tune parameters on the validation data. The training data is used to construct the event causality network. We retrieve 2-hop causal subgraphs according to input cause sentences because it is the most commonly used setting. The percentage of the test samples whose gold effect events exist in the retrieved causal subgraphs is 70.8%.

COPA Benchmark: The *Choice of Plausible Alternatives* (COPA) (Roemmele et al., 2011) dataset consists of 1,000 multiple-choice questions (500 for validation and 500 for testing) requiring causal reasoning in order to answer correctly. Each question is composed of a premise and two alternatives, and the task is to select a more plausible alternative as a cause (or an effect) of the premise. We use the most plausible alternative and its premise to collect cause-effect sentence pairs. The COPA causes are used to retrieve causal subgraphs from our event causality network, leading to 186 COPA pairs with their corresponding causal subgraphs. The percentage of the samples whose gold effect events exist in causal subgraphs is 11.2%. Because there is no released training data for the COPA task, we train all models on Enwiki and evaluate them on COPA.

[2]https://dumps.wikimedia.org/enwiki/20201020/enwiki-20201020-pages-articles.xml.bz2

4.2 Baselines and Evaluation

Baselines: We compare our method with state-of-the-art text generation methods, including GPT2 (Radford et al., 2019), BART(Lewis et al., 2019), CopyNet(Zhu et al., 2017) and CausalBERT(Li et al., 2020). Details can be seen in Appendix A.

Metrics: For automatic evaluation, we use metrics including BLEU-4 (Papineni et al., 2002), Distinct-n (Li et al., 2015) to evaluate the generated effect sentences. Abstraction-Matching (AbsMat) evaluates the percentage of the generated effect sequences that have the same abstraction as the corresponding gold effect sequences.

For the manual evaluation, we examine whether the generated sequence is a plausible effect of the input, which is denoted as *plausibility* (Plau). Details can be seen in Appendix B.

Result: The result is shown in Table 1, where EGCER achieves the best results. BART performs better than GPT2 due to the adopted encoder-decoder architecture. Based on the event skeletons provided by the effect event predictor, CopyNet and EGCER are aware of the topic which should be generated, and hence perform better than BART and GPT2. CopyNet performs worse than EGCER because CopyNet cannot cover all tokens of the retrieved event, as a result, the causal information in the generated sequence is incomplete. We also find that CopyNet tends to copy an event token repeatedly. CausalBert performs worse than EGCER because it is based on the word-level causal analysis, which can also be found in Section 4.3. Given the effect event, EGCER sees a more complete scenario, hence generate a more reasonable effect sentence.

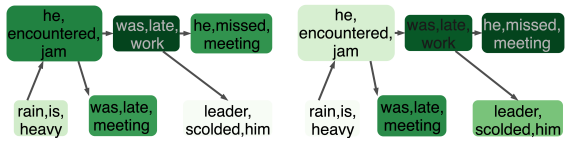
The result of the manual evaluation is also shown in Table 1. As for EGCER, we find that it may sometimes generate negation expressions or grammatical errors, as a result, the generated sequence is not a plausible effect even if the retrieved event is plausible. The proportion of the generated sequences in this case is about 21%. We speculate that the errors in data preprocessing and the insufficiently powerful generator are the possible reasons. In the future, we will further improve generators in order to generate more high-quality effect sentences. It can also be found that EGCER performs far worse on COPA than on Enwiki, this is because a great gap exists between these two datasets. However, EGCER is still superior to any other model, which demonstrates event-level causal reasoning

Model	EnWiki				COPA			
	BLEU-4	Distinct-1/2	AbsMat	Plau	BLEU-4	Distinct-1/2	AbsMat	Plau
GPT2	0.69	5.57/16.82	0.3	0.08	1.35	22.61/44.25	0.2	0.02
BART	1.28	8.23/24.83	1.7	0.11	1.22	22.37/43.71	0.5	0.04
CausalBERT	0.74	5.33/22.23	8.5	0.12	0.92	22.39/52.56	3.7	0.06
CopyNet	2.85	10.63/39.82	16.4	0.17	1.18	32.74/75.17	2.6	0.04
EGCER(ours)	4.90	13.99/43.58	26.4	0.27	1.74	48.08/83.97	5.3	0.07

Table 1: Automatic and manual evaluation results.

contributes to the effect sentence generation.

4.3 Visualization



(a) The causal scores calculated using the event vectors on the first layer of GNN.

(b) The causal scores calculated using the event vectors on the second layer of GNN.

Figure 2: The darker blue indicates the higher causal score.

Appendix C presents a case with generations of different models. CausalBERT generates "missing bus" given "missing" as guidance. However, from the input we can see that this person may be in a car, therefor the generated sequence is not an effect. That is CausalBERT, which is based on the word-level analysis, generates causal inconsistent sequence. In contrast, our method successfully predicts the expected effect event "*(he,missed,meeting)*", and generates the correct effect sentence.

We extract a part of CG according to the input cause, and visualize the causal scores cs using event vectors on the first and second layers of GNN respectively, as shown in Figure 2a and 2b. In Figure 2a, the "*(was, late, work)*" receives the highest score, followed by "*(he, encountered, jam)*" and "*(was, late, meeting)*" in one-hop reasoning. And, the "*(leader, scolded, him)*" receives the lowest score. Noted that "*(he, encountered, jam)*" is actually not an effect event. However, in Figure 2b, the "*(he, missed, meeting)*" receives the highest score, followed by "*(was, late, work)*", "*(was, late, meeting)*" and "*(leader, scolded, him)*" in two-hop reasoning. The "*(he, encountered, jam)*" and "*(rain, is, heavy)*" receive lower scores. This makes sense because they are not effect events at all. This shows that the multi-layer GNN can well capture multi-hop causal relationships and thus are able to select the plausible effect events.

4.4 Ablation Study

Models	BLEU-4	Distinct-1/2	AbsMat	Plau
Full model	4.90	13.99/43.58	26.4	0.27
w/o weights	4.37	14.10/42.86	23.3	0.24
w/o 2nd layer	3.89	13.15/41.56	20.6	0.21
w/o GNN	2.89	13.00/42.02	18.3	0.19

Table 2: Ablation study on the Enwiki testset.

To understand the importance of the key components of our approach, we perform an ablation study by training multiple ablated versions of our model, including the one without *weights* of edges in the retrieved causal subgraph, the one without the *2nd-layer* of GNN, and the one without *GNN*. The results are provided in Table 2. When the GNN module is gradually ablated, the performance of the model gradually degrades. This demonstrates that all modules of our multi-layer GNN effectively contribute to effect sentence generation.

5 Conclusion and Future Work

We present an event-level causal reasoning based effect generation method to generate the plausible effect sentences for the input cause sentences. Experiments show that our method performs better than competitors in capturing the causal semantics which should be generated. In the future, we would like to develop more effective approaches to enhance the effect event reasoning, and more powerful generators to generate the effect sentences with higher quality.

6 Acknowledgements

The work described in this paper was supported by and Research Grants Council of Hong Kong(PolyU/5210919, PolyU/15207920, PolyU/15207821), National Natural Science Foundation of China (61672445, 62076212, 62076072) and PolyU internal grants (ZVQ0). We are grateful to the anonymous reviewers for their valuable comments.

References

Prithviraj Ammanabrolu, Ethan Tien, Wesley Cheung, Zhaochen Luo, William Ma, Lara J Martin, and Mark O Riedl. 2020. Story realization: Expanding plot events into sentences. In *AAAI*, pages 7375–7382.

Nabiha Asghar. 2016. Automatic extraction of causal relations from natural language texts: a comprehensive survey. *arXiv preprint arXiv:1605.07895*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Quang Do, Yee Seng Chan, and Dan Roth. 2011. Minimally supervised event causality identification. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 294–303.

Chikara Hashimoto, Kentaro Torisawa, Julien Kloetzer, Motoki Sano, István Varga, Jong-Hoon Oh, and Yutaka Kidawara. 2014. Toward future scenario generation: Extracting event causality exploiting semantic relation, context, and association features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 987–997.

Oktie Hassanzadeh, Debarun Bhattacharjya, Mark Feblowitz, Kavitha Srinivas, Michael Perrone, Shirin Sohrabi, and Michael Katz. 2019. Answering binary causal questions through large-scale text mining: An evaluation using cause-effect pairs from human experts. In *IJCAI*, pages 5003–5009.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Zhongyang Li, Xiao Ding, Ting Liu, J Edward Hu, and Benjamin Van Durme. 2020. Guided generation of cause and effect. IJCAI.

Zhiyi Luo, Yuchen Sha, Kenny Q Zhu, Seung-won Hwang, and Zhongyuan Wang. 2016. Commonsense causal reasoning between short texts. In *KR*, pages 421–431.

Lara J Martin, Prithviraj Ammanabrolu, Xinyu Wang, William Hancock, Shruti Singh, Brent Harrison, and Mark O Riedl. 2018. Event representations for automated story generation with deep neural nets. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. 2016. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. *arXiv preprint arXiv:1607.00970*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Karl Pichotta and Raymond J Mooney. 2016. Learning statistical scripts with lstm recurrent neural networks. In *Thirtieth AAAI Conference on Artificial Intelligence*.

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

Kira Radinsky, Sagie Davidovich, and Shaul Markovitch. 2012. Learning causality for news events prediction. In *Proceedings of the 21st international conference on World Wide Web*, pages 909–918.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI spring symposium: logical formalizations of commonsense reasoning*, pages 90–95.

Karin Kipper Schuler. 2005. Verbnet: A broad-coverage, comprehensive verb lexicon.

Robyn Speer and Catherine Havasi. 2012. Representing general relational knowledge in conceptnet 5. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3679–3686.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Zhipeng Xie and Feiteng Mu. 2019. Distributed representation of words in cause and effect spaces. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7330–7337.

Sendong Zhao, Quan Wang, Sean Massung, Bing Qin, Ting Liu, Bin Wang, and ChengXiang Zhai. 2017. Constructing and embedding abstract event causality networks from text snippets. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 335–344. ACM.

Wenya Zhu, Kaixiang Mo, Yu Zhang, Zhangbin Zhu, Xuezheng Peng, and Qiang Yang. 2017. Flexible end-to-end dialogue system for knowledge grounded conversation. *arXiv preprint arXiv:1709.04264*.

A Experiment Setting

We concat cause-effect sentence pairs and finetune GPT2(117M) in a language model setting. BART is finetuned with the encoder-decoder setting. Both GPT2 and BART are implemented by *transformers*[3]. CopyNet employs the copy mechanism which either copies tokens from the retrieved event or generates words from the vocabulary. CausalBERT employs the lexically-constrained beam-search to generate possible effects for provided word guidance. ConceptNet(Speer and Havasi, 2012) is used to retrieve causal relevant constraints for CausalBERT.

Our effect event predictor consists of a 2-layer bidirectional GRU for encoding input sequences and a 2-layer GNN for updating event representations. Our event rewriter is a GRU decoder. The predictor and the rewriter do not share parameters, and their hidden sizes are set to 512. The word embedding size is 300. We use the Adam optimizer with the mini-batch size of 96. The learning rate is 0.001.

We use the gold effect event to supervise our event predictor. The objective is:

$$J_1 = -\log p(e_Y|X, CG). \quad (2)$$

For our event rewriter, the objective is to maximize the estimated probability of the gold effect sequence:

$$J_2 = P(Y|e_Y, X) = \sum_t -\log p(y_t|y_{<t}). \quad (3)$$

The final loss function is the combination of the above two

$$J = J_1 + J_2 \quad (4)$$

B Details for Manual Evaluation

100 samples are randomly selected from the Wikipedia test set and COPA, respectively, and distribute them to the two graduate students from the NLP field. Each student is asked to give a score from {0, 0.5, 1} for the (input, generation) pair, given the following guidelines. Assign 0 to the pair if the generation can never be considered as a possible effect of the input, assign 0.5 to the pair if the generation is a possible effect of the input but has certain grammatical errors and assign 1 to the pair if the generation is a possible effect of the input and there is no grammatical error. We average scores over the two annotators. The *cohen's kappa* scores on Enwiki and COPA are 0.65 and 0.63, respectively.

C Generation Example

Input cause	he encountered a heavy traffic jam.
GPT2	the lighthouse was closed over three weeks.
BART	he was delayed for over an hour.
CopyNet	he missed missed the meeting.
CausalBert	causing him to miss bus.
EGCER	he missed the important meeting.

Table 3: A case with generations of different models.

Given the input cause, CauseBERT generates the unexpected sequences by using "missing" as constraint, which demonstrates that word-level causal analysis is not always self-contained. CopyNet repeatedly generates the "missed" token. EGCER rewrites the predicted effect event "(*he, missed, meeting*)" into the reasonable effect sentence.

[3]https://huggingface.co/