Logic Enhanced Commonsense Inference with Chain Transformer

Chenxi Yuan
Department of Computer Science and Technology, Tsinghua University
ycx18@mails.tsinghua.edu.cn

Chun Yuan*
Tsinghua ShenZhen International Graduate School, Peng Cheng Laboratory
yuanc@sz.tsinghua.edu.cn

Yang Bai
Department of Computer Science and Technology, Tsinghua University
bai-y18@mails.tsinghua.edu.cn

Ziran Li
Department of Computer Science and Technology, Tsinghua University
lizr18@mails.tsinghua.edu.cn

ABSTRACT

We study the commonsense inference task that aims to reason and generate the causes and effects of a given event. Existing neural methods focus more on understanding and representing the event itself, but pay little attention to the relations between different commonsense dimensions (e.g. causes or effects) of the event, making the generated results logically inconsistent and unreasonable. To alleviate this issue, we propose Chain Transformer, a logic enhanced commonsense inference model that combines both direct and indirect inferences to construct a logical chain so as to reason in a more logically consistent way. First, we apply a self-attention based encoder to represent and encode the given event. Then a chain of decoders is implemented to reason and generate for different dimensions following the logical chain, where an attention module is designed to link different decoders and to make each decoder attend to the previous reasoned inferences. Experiments on two real-world datasets show that Chain Transformer outperforms previous methods on both automatic and human evaluation, and demonstrate that Chain Transformer can generate more reasonable and logically consistent inference results.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; *Knowledge representation and reasoning*.

KEYWORDS

commonsense inference; commonsense knowledge; logical chain; attention network

ACM Reference Format:
Chenxi Yuan, Chun Yuan, Yang Bai, and Ziran Li. 2020. Logic Enhanced Commonsense Inference with Chain Transformer. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3340531.3411895

*Corresponding author: yuanc@sz.tsinghua.edu.cn

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3411895

1 INTRODUCTION

Commonsense inference, which aims to reason about the unobserved conditions from the observed event, is a significant but challenging task in Neural Language Processing. It is easy for human with a certain foundation of commonsense knowledge to reason about unobserved causes and effects related to the observed event but hard for today's AI systems. In this paper, we focus on the commonsense inference task that reasons the causes and effects of events. An example is shown in Figure 1. From the given event "X walks into the hospital", it is easy to reason about the causes of the event (X's intent, what X needed to do before the event) and the effects of the event (the reactions of X and others, what X or others want to do after the event, etc.). We aim to generate textual descriptions of inference results instead of selecting a reasonable choice from some candidates. The generative process, which is more in line with the process of human knowledge inference, is still a new territory and under-explored.

Previous works using neural models based on encoder-decoder architecture treat the task as an end-to-end learning problem and implement multi-task learning for better representation of the event [20, 22]. Bosselut et al. [1] utilizes transfer learning to transfer indirect clue from large pre-trained language models [18] while Du et al. [5] proposes a context-aware variational autoencoder to learn the event background knowledge from auxiliary datasets. These approaches focus more on exploring the direct relations between the event and inference results or introducing background commonsense knowledge for better event representation, paying little attention to the indirect relations between different commonsense dimensions (e.g. the relation between the cause and effect of the event). Despite generating grammatically correct, fluent, and diverse inference results, these methods still suffer from the problem of logical inconsistency among different commonsense dimensions.

To guarantee logical consistence in commonsense inference, the event itself and the indirect relations among different commonsense dimensions should both be taken into account. For instance, given the event "X walks into the hospital", if we consider X's intent is "to visit a patient" then we can infer that he needs "to buy some flowers" and Y(the patient)'s reaction is "grateful to X". Otherwise, if X's

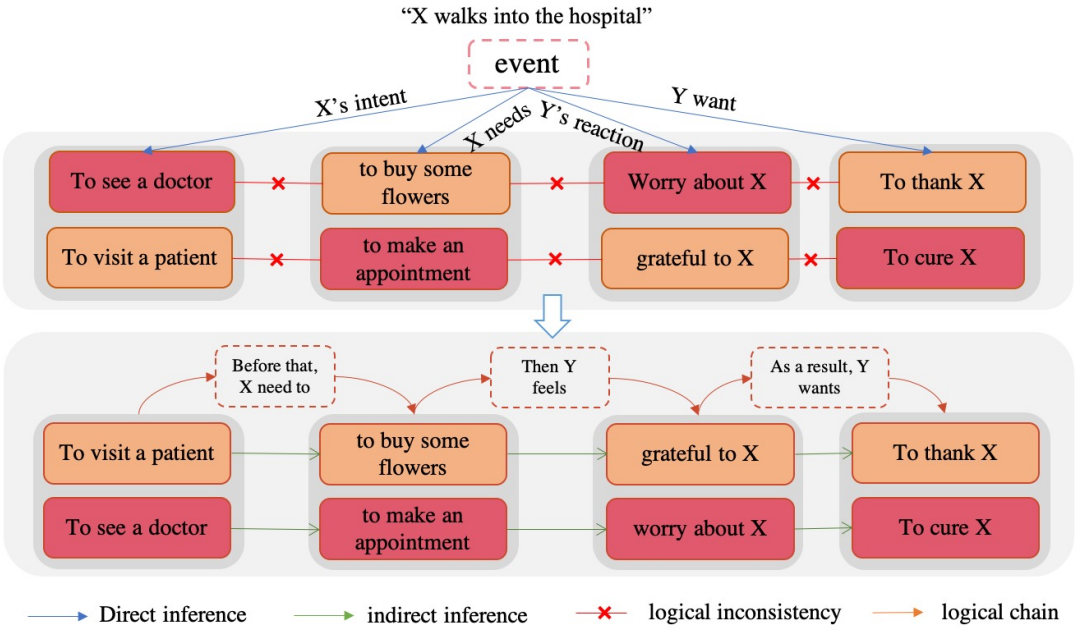

Figure 1: Comparing the reasoning process of previous methods(top) and our advanced method(bottom). Previous methods generate inference results directly, which usually leads to logical inconsistency. Our method follows the logical chain and leverage both direct and indirect clues, which can achieve logically consistent results.

intent is "to see a doctor", the inference results on other dimensions would be completely different. Consequently, we propose a logic enhanced reasoning method that incorporates both direct and indirect relations to construct a logical chain to model commonsense inference in a more consistent way. As illustrated in figure 1, we follow the timeline of the event development to organize the logical chain, which concatenates different commonsense dimensions in a logical order. During the reasoning process, the first step is to leverage the event as a direct clue to reason about the first commonsense dimension directly, which is then regarded as an indirect clue for the next commonsense dimension. Then guided by the logical chain, we can reason about all the dimensions based on both the direct clue and indirect clues.

In this paper, we design Chain Transformer to model the logic enhanced commonsense inference following the logic chain about the event, which incorporates both direct and indirect inferences. Firstly a multi-head self-attention based encoder is used to represent and encode the input event. Then a chain of multi-head attention based decoders is employed to decode and generate inference results of different dimensions, where a dimension corresponds to a decoder. For the direct inference, an event attention is applied to leverage the information captured by the event encoder. For the indirect inferences, all decoders are linked with a designed hierarchical attention module named chain attention, which allows each decoder to look back at the decoders in front of it, thereby utilizing previously generated results to guide the reasoning of current dimension. Besides, we devise a qualitative pretraining strategy on external commonsense corpora to learn background commonsense knowledge of the events so as to boost the ability of our model on indirect inference. With the benefit of the logical chain, our model can not only generate diverse and reasonable inference results but also guarantee the logical consistency among different commonsense dimensions.

We conduct experiments on two large benchmark datasets named Event2mind and ATOMIC, which aim to reason and generate the causes and effects of a given event. The experimental results demonstrate that our proposed model outperforms the state-of-the-art methods on both automatic evaluation metrics and human evaluation. We also implement ablation studies and multi-step reasoning experiments to evaluate the effectiveness of the chain structure. The results indicate that our model can reason and generate more reasonable and logically consistent inferences.

2 RELATED WORK

2.1 Event-centered Commonsense Inference

Event-centered commonsense inference is of great significance which can be applied to robots for better human-robot interaction [11, 14, 21] and constructing commonsense knowledge graph [1, 23], etc. A large number of researchers focus on event-centered commonsense inference such as script event prediction [9, 13, 26] and story ending generation [2, 15, 16]. There are several lines of work focusing on representing and reasoning about sequences of events. Chambers and Jurafsky [2] propose the "Narrative Cloze Test", ROCStories and the "Story Cloze Test" is proposed [15, 16] that reasons about which event happens next from an event chain.

Commonsense inference on causes and effects of events is also a significant domain, since the inference of causes and effects from the

observed event is a fundamental capability of humans [8]. Triangle-COPA [8] is based on a social psychology experiment where questions focus specifically on emotions, intentions, and other aspects of social psychology. Based on the stories from ROCStories, Rashkin et al. [19] propose Story Commonsense that reasons about the cause and effect of mental state changes of characters in a story. This task mainly focuses on the psychology of story characters. Then commonsense inference on intent and reactions is proposed by Rashkin et al. [20] named Event2mind. Each event in Event2mind involves one or two participants, and presents three tasks of predicting the primary participant's intentions and reactions, and predicting the reactions of others. The task is then extended to ATOMIC by Sap et al. [22], which is organized through 877k textual descriptions of inferential knowledge and contains nine different sub-tasks based on nine different "*if-then*" relations.

In this paper, we mainly focus on the commonsense inference that reasons about the causes and effects of the event, which are more challenging and require more commonsense knowledge of everyday events. Further, we focus on the generative tasks, which means reasoning and generating possible inference results instead of classifying or choosing a choice from some candidates.

2.2 Neural Models for Commonsense Inference

Existing approaches for generative commonsense inference apply neural models with encoder-decoder architecture to learn and understand events with encoder and to generate textual descriptions of the inferences with decoder. Rashkin et al. [20] explore different encoders for better representation of the input event such as CNN, which has the ability of representation learning and feature extraction and RNNs, which can learn non-linear features and time-series information of sequences efficiently. Since self-attention which is first proposed by Vaswani et al. [24] which can capture long dependency of texts better than RNN and CNN and has been proven effective on machine translation [24] and other generation tasks such as abstract text summarization [27] and paraphrase generation [25], Bosselut et al. [1] propose a transformer-based model (COMET) and implement transfer learning to transfer indirect knowledge from pretrained GPT models [18], aiming to generate rich and diverse commonsense descriptions in natural language for construction of commonsense knowledge graph.

A large amount of researches focuses on better representing and encoding the event, such as large-scale pre-training and multi-task learning. Rashkin et al. [20] and Sap et al. [22] use multi-task learning to train the neural models that different decoders share the same event encoder so that the encoder can learn from different reasoning relations for better understanding the events. Implementing large-scale pre-training on external commonsense knowledge corpus is another way for better commonsense inference. Recent popular pretraining language models include ELMO [17], BERT [4] and GPT [18] etc. Sap et al. [22] use pretrained ELMO for better representing the input event while Bosselut et al. [1] implement transfer learning to transfer implicit commonsense knowledge from pretrained GPT model [18]. Du et al. [5] propose a context-aware variational autoencoder to learn background commonsense knowledge of the events from auxiliary corpus. Since one event may be caused by several different causes and have different possible effects, diversity is another target of recent researches. Beam search [7] is applied to select some likely candidate results for each inference dimension [1, 20, 22].

However, these methods pay little attention to the indirect relations between different commonsense dimensions of the event. Further, implementing beam search for each dimension may cause the results lacking logical consistency. Differing from all these methods, we propose a logic enhanced reasoning method to leverage both direct and indirect inferences to improve the quality of inference results and keep logical consistency. For generating diverse inference results, we only implement beam search once on reasoning the first commonsense dimension and use greedy search for other dimensions. Then guided by the logical chain our model can generate diverse by logically consistent inference results for other dimensions.

3 METHOD

The input of our model is a sentence of event $x = (x_1, x_2, \ldots, x_N)$, where $\boldsymbol{x_i}$ is the word vector of i-th word in the sentence, and the output of our model is the inference results of different commonsense dimensions $y = \{y_1, y_2, \ldots, y_K\}$ where y_k is the inference results of the k-th commonsense dimension. The goal of our model is to maximum $P(y|x)$. Further, we formulate the logic chain as the joint probability of different commonsense dimensions and decompose $P(y|x)$ into:

$$P(y|x) = P(y_1, y_2, ..., y_K|x) = \prod_{k=1}^{K} P(y_k|y_{<k}, x), \tag{1}$$

where $P(y_k|y_{<k}, x)$ is estimated by our model with both direct and indirect inferences, $y_{<k}$ are seen as the indirect clues of y_k.

As shown in Figure 2, the architecture of our model consists of three following parts: (1) **Input Representation** where word embedding and position encoding are first applied to represent the input event, (2) **Event Encoder** where a multi-head self-attention based encoder is employed to encode the event, (3) **Chain of Decoders** where a chain of multi-head attention based decoders is employed to reason and generate inference results for different dimensions, where a chain attention is designed to make different decoders linked to model both direct and indirect inference.

3.1 Input Representation

The input event is a sequence of words which are first embedded to vectors $\boldsymbol{x} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}) \in \mathbb{R}^{n \times d_{model}}$, n is length of the sentence and d_{model} is the dimension of the word vector. Then the word embeddings are modified by a position encoding process which is used to leverage the order information of the sequence. The position encoding of a word in a sequence is defined as follows:

$$PE(pos, 2i) = \sin(pos/10000^{2i/d_{model}}), \tag{2}$$

$$PE(pos, 2i+1) = \cos(pos/10000^{2i/d_{model}}), \tag{3}$$

where pos is the position of the word in the sequence, i is the dimension. The position encodings have the same dimension as the word embeddings. Then the word embeddings and position encodings are sumed to represent the word. Therefore the input

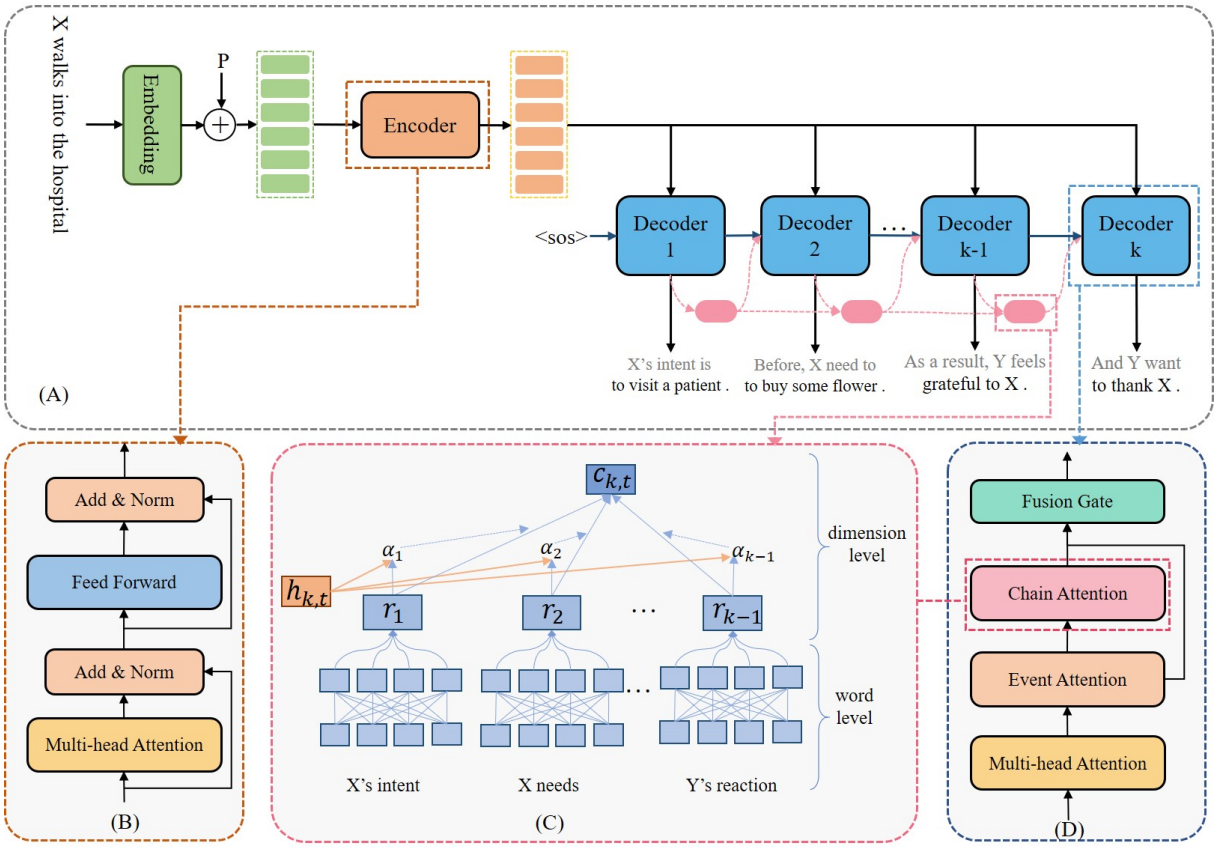

Figure 2: The architecture of out proposed model. (A) is the whole structure of the model. The model starts from a input representation then an event encoder and generates inference results for different commonsense dimensions of the event through a chain of decoders, (B)shows the structure of event encoder, (D) is the structure of one single decoder in which the chain attention (C) helps the decoder to leverage the information from the circumstantial clues. As shown in (C), the chain attention is a hierarchical structure, which contains a word-level attention and a dimension-level attention.

event is represented as $\boldsymbol{X} \in \mathbb{R}^{n \times d_{model}}$ which is then used as the input to multi-head self-attention based encoder.

3.2 Event Encoder

After the representation, the input event is then encoded by a multi-head self-attention based encoder, which contains a stack of blocks that are all identical in structure and do not share weights. Each block is broken down into two layers, multi-head self-attention layer, and feed-forward layer.

The input to encoder first flows through a multi-head self-attention layer, which helps the encoder capture the dependencies between different words in the event. Then the output of the attention layer is fed to a feed-forward neural network (FFN) and layer normalization:

$$\boldsymbol{x}^{\boldsymbol{e}} = \text{MultiHead}(\boldsymbol{x}, \boldsymbol{x}, \boldsymbol{x}) \tag{4}$$

$$\boldsymbol{e} = \text{LayerNorm}(\boldsymbol{x}^{\boldsymbol{e}} + FFN(\boldsymbol{x}^{\boldsymbol{e}})), \tag{5}$$

where Multihead is the multi-head self-attention layer that takes queries $\boldsymbol{Q}$, keys $\boldsymbol{K}$ and values $\boldsymbol{V}$ as inputs and is calculated as follows:

$$\text{MultiHead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \boldsymbol{z}_1 \oplus \boldsymbol{z}_2 \oplus, ..., \oplus \boldsymbol{z}_h, \tag{6}$$

$$\text{where } \boldsymbol{z}^h = \text{Attention}(\boldsymbol{Q}\boldsymbol{W}_q, \boldsymbol{K}\boldsymbol{W}_k, \boldsymbol{V}\boldsymbol{W}_v), , \tag{7}$$

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^{\boldsymbol{\intercal}}}{\sqrt{d_k}})\boldsymbol{V}, , \tag{8}$$

where $\boldsymbol{W}_q, \boldsymbol{W}_k, \boldsymbol{W}_v \in \mathbb{R}^{d_k \times d_k}$ are parameters, $d_k = d_{model}/h$, h is the number of attention heads, $\oplus$ represents the concatenation operation.

After the N-block encoding, the input event is encoded to $\boldsymbol{e} \in \mathbb{R}^{n \times d_{model}}$.

3.3 Chain of Decoders

After the event is encoded, a chain of multi-head attention based decoders is employed to decode and reason from the event to generate different target inferences. All decoders in the chain are identical in structure and do not share weights. To model both direct and indirect inference processes, all decoders are linked by a designed attention module, which we refer to as Chain Attention. With the benefit of Chain Attention, the chain structure allows each decoder to leverage the information captured by the previous decoders as

indirect clues for more reasonable and logically consistent inference.

3.3.1 Chain Attention. The goal of chain attention is to leverage the information captured by the previous decoders to update the hidden state in the current decoder:

$$\boldsymbol{c}_{k,t} = \text{Chain}(\hat{\boldsymbol{h}}_{k,t}, \boldsymbol{h}_{<k}), \tag{9}$$

where $\hat{\boldsymbol{h}}_{k,t}$ is the hidden state of the k-th decoder at the t-th decoding step, $\boldsymbol{h}_{<k}$ are the hidden states of the previous decoders.

Considering that not all previously obtained inferences have the same importance to the current inference dimension, we design the Chain Attention to help each decoder select more important ones from the indirect clues to update the hidden state of the decoder. This mechanism can also avoid error propagation between different decoders to the greatest extent. Chain attention is a hierarchical attention architecture which contains a word-level attention and a dimension-level attention. Here we take the k-th decoder as an example to show how chain attention works.

For word-level, first a multi-head self-attention layer is applied to capture dependency between different words in the same previously obtained inference, then a mean pooling layer and single-layer perceptron are used to represent each inference as a vector:

$$\hat{\boldsymbol{r}}_j = \text{Multihead}(\hat{\boldsymbol{h}}_j, \hat{\boldsymbol{h}}_j, \hat{\boldsymbol{h}}_j) \tag{10}$$

$$\boldsymbol{r}_j = \text{ReLU}(\boldsymbol{W}_h[\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{r}_{j,t}] + \boldsymbol{b}_h) \tag{11}$$

where $\boldsymbol{h}_j$ are the hidden state of the j-th decoder ($j < k$), $\boldsymbol{W}_h \in \mathbb{R}^{d_{model} \times d_{model}}$, $\boldsymbol{b}_h \in \mathbb{R}^{d_{model}}$ are parameters.

For dimension-level, an attention mechanism is applied to estimate the importance of each inference, the attention weight is calculated as follows:

$$\boldsymbol{u}_j = \tanh(\boldsymbol{r}_j \boldsymbol{W}_r + \boldsymbol{b}_r), \tag{12}$$

$$\alpha_j = \frac{\exp(\boldsymbol{u}_j^\top \boldsymbol{h}_{k,t})}{\sum_j^{k-1}\exp(\boldsymbol{u}_j^\top \boldsymbol{h}_{k,t})}, \tag{13}$$

where $\boldsymbol{W}_r \in \mathbb{R}^{d_{model} \times d_{model}}$, $\boldsymbol{b}_r \in \mathbb{R}^{d_{model}}$ are parameters. α_j represents the importance of $\boldsymbol{r}_j$ to the current reasoning target. Finally the current hidden state $\hat{\boldsymbol{h}}_{k,t}$ is updated as a weighted sum of the circumstantial clues:

$$\boldsymbol{c}_{k,t} = \text{Chain}(\hat{\boldsymbol{h}}_{k,t}, \boldsymbol{h}_{<k}) = \sum_j \alpha_j \boldsymbol{u}_j, \tag{14}$$

3.3.2 Single Decoder. Each single decoder is based on the Transformer [24] decoder including a stack of blocks while each block is composed of three different attention layers: masked multi-head self-attention layer, event attention layer and chain attention layer.

At each decoding step t in the k-th decoder, first a masked multi-head attention is applied to capture dependency of the generated words $\boldsymbol{y}_{<t}$ and obtain the hidden states $\boldsymbol{h}_{k,t}$, where the masked multi-head attention is almost the same as multi-head self-attention where the mask is utilized to ensure the prediction of generating the next word only depends on the sequence ahead of the word. Then an event attention layer and a chain attention layer are implemented to leverage the information from the event encoder and the previous decoders.

$$\hat{\boldsymbol{h}}_{k,t} = \text{MultiHead}(\boldsymbol{h}_{k,t}, \boldsymbol{e}, \boldsymbol{e}), \tag{15}$$

$$\boldsymbol{c}_{k,t} = \text{Chain}(\hat{\boldsymbol{h}}_{k,t}, \boldsymbol{h}_{<k}), \tag{16}$$

where $\boldsymbol{h}_{<k}$ represent the hidden states of the previous decoders, Chain is calculated by Section 3.3.1. Then a fusion gate is utilized to fuse the information captured by the two attention layers:

$$\boldsymbol{r}_h = \tanh([\boldsymbol{c}_{k,t} \oplus \hat{\boldsymbol{h}}_{k,t}]\boldsymbol{W}_r + \boldsymbol{b}_r) \tag{17}$$

$$\boldsymbol{g}_h = \sigma([\boldsymbol{c}_{k,t} \oplus \hat{\boldsymbol{h}}_{k,t}]\boldsymbol{W}_g + \boldsymbol{b}_g) \tag{18}$$

$$\boldsymbol{h}_{k,t} = \boldsymbol{r}_h \boldsymbol{g}_h + (1 - \boldsymbol{g}_h)\boldsymbol{c}_{k,t} \tag{19}$$

where $\boldsymbol{W}_r, \boldsymbol{W}_g \in \mathbb{R}^{2d_{model}, d_{model}}$, $\boldsymbol{b}_r, \boldsymbol{b}_g \in \mathbb{R}^{d_{model}}$ are parameters, $\boldsymbol{d}_t$ is the hidden state at the t-th decoding step.

After a linear layer with a softmax activation, the probability of generated inference result $y_{k,t}$ is obtained:

$$P(y_{k,t}|y_{k,<t}, y_{<k}, x) = \text{softmax}(\boldsymbol{h}_{k,t}\boldsymbol{W}_v), \tag{20}$$

where $\boldsymbol{W}_v \in \mathbb{R}^{d_{model} \times \mathcal{V}}$ is the word embedding matrix and $\mathcal{V}$ is the vocabulary size.

3.4 Training

3.4.1 Pretrain: Learning the Commonsense and Logic Knowledge. For better understanding and representing the event and to learn the indirect inference with logical chain, we pretrain our model on three external corpora ROCStories [15], VIST [10] and WritingPrompts [6] to learn background commonsense knowledge of the events. ROCStories and VIST are composed of five-sentence commonsense stories as shown in Figure 3. We follow the work of Du et al. [5] that cut the stories in WritingPrompts into five-sentence-paragraphs.

The pretraining is divided into two stages. First, we take the first four sentences as the input event x to directly reason about the last sentence as the inference target y as shown in Figure 3 (1-step). We use the event encoder to encode the first four sentences x to reason about and generate the last sentence with a single decoder.

Further, to learn the indirect reasoning process we decompose the reasoning process into a sentence-by-sentence generation process, which is shown in Figure 3 (2-step to Multi-step). We use the first k ($k \in [1, 4]$) sentences as the input event to generate the last $5 - k$ sentences one by one with the chain of decoders.

3.4.2 Fine-tune: Learning to Reasoning about Each Commonsense Dimensions. During the training stage, we adapt the pretrained model to Event2mind and ATOMIC to learn the commonsense knowledge for each dimension, where we use the event encoder to encode the event to reason about different commonsense dimensions one by one with the chain of decoders following the logical chain. During training stage we use the ground-truth inference results from the training set to fill the designed templates as the inputs to each decoder. Figure 4 shows a template of the input to decoder, which use some conjunctions, subjects and predicates to associate different commonsense dimensions together to get a coherent and complete paragraph.

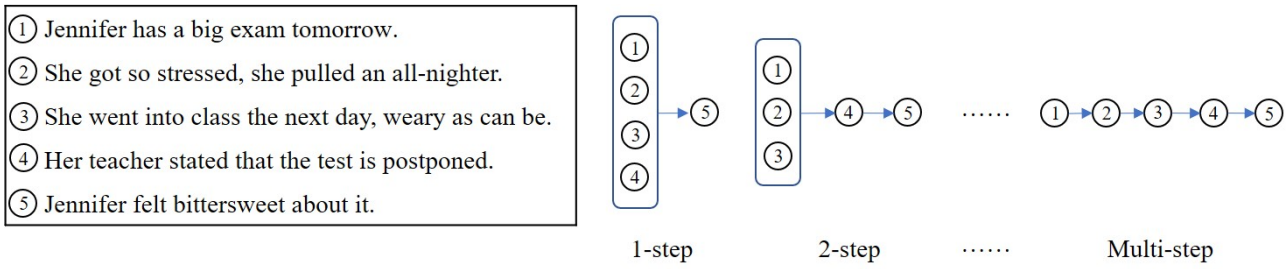

Figure 3: Illustration of the pre-training process of Chain Transformer, where the arrows represent the reasoning steps.

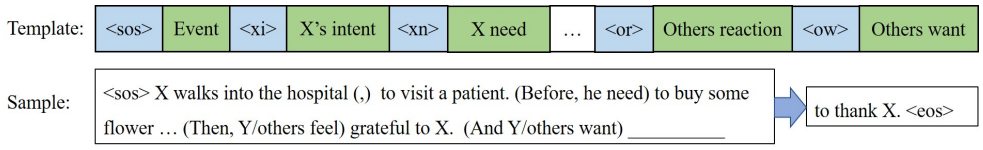

Figure 4: Template and a sample of the input to a Chain Transformer decoder.

Table 1: An example of ATOMIC which aims to reason about nine different dimensions from a given event. Each dimension contains various possible inference results.

Event	Causes of event		Effects on X		Effects on Y	
X walks into the hospital	xIntent	to visit a patient to see a doctor	xEffect	worrying about Y feeling better	oEffect	be happy relived
	xNeed	to buy some flowers to make an oppointment	xReact	sympathetic worried	oReact	grateful to X worry about X
	xAttr	kindhearted careful	xWant	to comfort Y to care about himself	oWant	to thank X to cure X

3.4.3 Prediction: Reasoning about Different Dimensions from Unseen Events. During the prediction stage, given the input event, we reason about different commonsense dimensions one by one following the logical chain. During the testing stage we use the inference results generated by the previous decoders to fill the template as the input to the posterior decoders.

For diverse inference results, we implement beam search when reasoning about the first dimension and obtain k different results. Then following the logical chain we can reason about diverse inferences for the following dimensions. Moreover, with the guidance of the logical chain, the obtained results of different dimensions are more logically coherent.

3.4.4 Loss Function. We train our model as a joint learning of different dimensions to minimize the negative log-likelihood of the gold inferences:

$$\mathcal{L} = -\sum_{(x,y)\in\mathcal{D}} \sum_{k} \sum_{t} \log P(y_{k,t} | y_{k,<t}, y_{<k}, x) \tag{21}$$

where $\mathcal{D}$ represents all the training samples including input event and target inferences.

4 EXPERIMENTS

4.1 Experiment Settings

Datasets: Experiments are conducted on two datasets: Event2Mind [20] which aims to generate three different inferences (X's intent, X's reaction and others' reaction) from a given event. ATOMIC which is proposed based on Event2mind, is extended to nine dimensions according to nine "*if-then*" relations. An example of ATOMIC is shown in Table 1.

Training Details: The parameters of our model are initialized from the pretrained model mentioned above. Then the model is fine-tuned to learn from each inference dimension. During fine-tuning stage, we first separate the nine commonsense dimensions of ATOMIC into three groups as shown in Table 1. Then we design two different logical chains for reasoning: causes of the event (xIntent » xNeed » xAttr) » effects on X (xEffect » xReact » xWant) and causes of the event » effects on Y/others (oEffect » oReact » oWant). [1] Following the logical chains, we train Chain Transformer for each inference dimension separately one by one by fixing the parameters of the previous decoders, then jointly. During joint training we average the loss of the dimensions.

[1] Considering that the logical relations between effects on X and effects on Y/others are not that obvious and longer logical chain may cause more serious error propagation, here we cut the nine commonsense dimensions into two different logical chains.

Table 2: Perplexity and BLEU score of different models on Event2mind, where −Pretrain represents the model without pre-training stage whose parameters are initialized randomly, −Chain represents the model without Chain Attention.

Model	PPL			BLEU		
	xIntent	xReact	oReact	xIntent	xReact	oReact
Seq2seq [22]	40.75	30.38	15.27	13.84	3.38	5.04
Transformer [24]	31.37	25.74	13.35	16.16	4.51	5.97
GPT [18]	30.15	24.32	12.04	17.68	4.76	6.53
CWVAE [5]	29.23	23.17	11.04	12.98	5.65	6.97
ChainT	**27.67**	**22.58**	**10.61**	**19.41**	**5.83**	**7.54**
−Pretrain	28.31	23.06	11.28	18.25	5.19	6.81
−Chain	29.05	23.65	11.73	17.48	4.96	6.27
−Both	30.54	24.67	12.30	16.75	4.53	6.05

Table 3: Perplexity and BLEU score of different models on ATOMIC.

Metric	Model	xIntent	xNeed	xAttr	xEffect	xReact	xWant	oEffect	oReact	oWant
PPL	Seq2seq [22]	22.07	23.98	26.20	60.15	27.80	26.36	32.41	18.34	15.07
	Transformer [24]	18.38	22.56	24.30	55.87	25.54	26.17	30.15	17.82	14.26
	GPT [18]	16.23	20.17	22.69	53.51	23.18	25.82	28.74	17.16	13.64
	CWVAE [5]	15.93	20.32	23.85	50.74	21.39	24.02	29.13	14.02	11.70
	ChainT	**14.37**	**16.75**	**20.18**	**48.21**	**19.94**	**22.78**	**23.71**	**13.27**	**10.20**
	−Pretrain	15.54	19.20	22.17	52.38	22.07	23.25	26.62	16.19	12.07
	−Chain	16.12	20.27	22.85	53.34	22.71	23.55	27.20	16.93	12.74
	−Both	16.84	21.02	23.72	55.40	24.21	25.95	28.37	17.51	13.42
BLEU	Seq2seq [22]	8.51	14.80	5.06	10.64	5.43	14.5	8.02	6.38	12.48
	Transformer [24]	15.68	19.25	8.95	10.05	8.21	19.19	7.25	5.21	16.68
	GPT [18]	16.08	22.54	16.38	10.36	9.27	23.78	7.68	5.97	18.21
	CWVAE [5]	12.12	15.67	5.63	14.64	8.13	15.01	13.83	8.58	14.93
	ChainT	**17.15**	**25.56**	**20.96**	**15.45**	**13.91**	**26.25**	**13.98**	**8.87**	**21.37**
	−Pretrain	16.23	22.38	18.26	11.90	10.35	23.54	12.24	7.85	19.94
	−Chain	16.32	22.14	16.93	11.28	9.34	21.16	11.07	7.05	18.24
	−Both	15.91	21.75	16.12	10.37	8.96	20.62	9.74	6.52	17.56

Hyperparameters: We set the vocabulary size $V = 20,000$. The dimension of word vector and hidden state $d_{model} = 300$ and the number of multi-head attention $h = 6$. We apply dropout at a rate of 0.3. Learning rate is 0.003 with an Adam optimizer [12]. We train our model on different dimensions separately for 10 epochs and jointly for 30 epochs.

Evaluation Metrics: Following the recent works [5, 20, 22], we choose perplexity and BLEU score [3] to evaluate our model. We also implement qualitative human evaluation to measure the quality of the results. We do not follow the work of [1] that evaluate the novelty of results. Because their goal is to construct knowledge graph automatically but our goal is to make a more accurate and logically consistent commonsense inference.

4.2 Baselines

We compare our model with some strong baseline models as follows: **Seq2seq** is a LSTM-based Sequence-to-sequence model proposed by Sap et al. [22] as the baseline of ATOMIC. **Transformer** [24] is a self-attention based encoder-decoder model, we also pretrain Transformer on ROCStories and then implement it on the two datasets. **GPT** [18] is a transformer based pretrained language model which is used for commonsense inference by Bosselut et al. [1] for commonsense knowledge graph construction. **CWVAE** [5] is a context-aware variational autoencoder which learns background information of events.

4.3 Overall Results

The results of the automatic evaluation on two datasets are reported in Table 2 and Table 3. The results show that our proposed model outperforms all the baseline models on the two datasets and achieves state-of-the-art performance, demonstrating that our proposed model is able to facilitate commonsense inference. Specifically, we find that:

(1) The improvement of our model on all the commonsense dimensions of the two datasets confirms that our proposed logic enhanced reasoning method can enhance the ability of neural models on commonsense inference.

Table 4: BLEU score of our model and different multi-task learning setups on the test set of ATOMIC.

Model	xIntent	xNeed	xAttr	xEffect	xReact	xWant	oEffect	oReact	oWant
9ENC9DEC	12.68	15.25	6.95	11.05	8.21	15.19	7.85	7.21	13.68
EVENT2(IN)VOLUNTARY	11.57	17.86	7.16	10.13	7.76	16.75	8.59	7.03	14.74
EVENT2PERSONX/Y	11.31	16.42	7.24	12.16	8.07	14.27	19.28	6.89	15.25
EVENT2PRE/POST	12.04	15.83	6.58	10.84	9.38	15.56	7.40	6.57	13.32
ChainT	**15.15**	**20.56**	**9.96**	**15.45**	**10.91**	**18.25**	**13.98**	**8.87**	**17.37**

Table 5: Human evaluation results of different models on ATOMIC.

Model	Fluency	Rationality	Consistency
Seq2seq	3.97	2.48	2.12
Transformer	4.13	2.75	2.25
GPT	4.29	2.93	3.07
ChainT	4.31	3.09	3.57
−Pretrain	4.25	2.96	3.42
−Chain	4.27	2.67	3.18
−Both	4.18	2.52	3.02
Reference	4.52	4.27	4.06

(2) Pretrained models (GPT, CWVAE and ChainT) perform better than those without pretraining (Seq2seq, Transformer and ChainT-pretrain), showing that learning background knowledge with pretraining strategy enhances the ability of models on understanding and representing the input events for more accurate commonsense inference.

(3) The comparison between the performances of ChainT and -Chain demonstrates that the Chain Attention really performs on improving the indirect inferences for more accurate and reasonable commonsense inference.

(4) The great improvement of our model on "xEffect", "xReact" and "xWant" in ATOMIC (Table 3) further illustrates the effectiveness of the indirect inference, since we choose "xIntent" as the start of reasoning process and the relations between the causes and effects on X are much closer.

4.4 Human Evaluation

To further evaluate the quality of the generated results of our model, we implement human evaluation on three aspects: **fluency** which measures whether the generated inferences are fluent and grammatically correct, **rationality** which measures whether the inferences are reasonable and faithful to the input event, **consistency** which measures whether the inferences of different dimensions are logically consistent (e.g. X's intent *"to see a doctor"* and others' reaction *"thankful"* are not logically consistent). We randomly select 300 samples from the test set of ATOMIC generated by three baselines and our proposed model. We invite five experts with enough background commonsense knowledge to score the results from different models from 0−5, higher is better.

The results are shown in Table 5, showing that our proposed model outperforms all the baselines on all the three metrics. Specifically, we find that:

(1) All neural models perform well on fluency, illustrating that recent neural encoder-decoder models are able to generate fluent and grammatically correct descriptions.

(2) Pretrained models have better ability on generating more reasonable inferences than those without pretraining, which demonstrates that learning background knowledge with pretraining truly improves the ability of neural models on commonsense inference.

(3) Our proposed model gets a remarkable improvement on consistency, indicating the efficiency of the proposed logic enhanced reasoning method on generating more logically consistent inferences.

The results of human evaluation demonstrate that our model can generate more fluent, reasonable and logically consistent inferences.

4.5 Comparison with Multi-task Learning

To further estimate our proposed logic enhanced reasoning method that incorporates indirect inference with indirect inference, we conduct experiments on ATOMIC to compare our method with multi-task learning methods, which is also implemented on some previous works[1, 20, 22]. During multi-task learning, the nine commonsense dimensions of ATOMIC are first separated into different groups according to the hierarchical structure of the commonsense dimensions. The dimensions in the same group share the same encoder and are trained jointly by averaging the loss of each dimension. We follow the work of [22] that train our model (without chain) with different multi-task learning setups based on different grouping rules. For comparison, we also train nine single models separately for nine different dimensions (9ENC9DEC).

The results are reported in Table 4, showing that our proposed method has a better performance compared to all multi-task learning methods, further demonstrating that our proposed logic enhanced reasoning method can better exploit the relations between different commonsense dimensions. We also find that multi-task learning can improve the performance of the model on some dimensions to some extent. In multi-task learning, the relations between different dimensions are only used for category, and there is no interaction during reasoning process. On the contrary, our proposed logic enhanced method can leverage the relations to construct logical chain to guide the reasoning for better commonsense inference.

4.6 Multi-step Reasoning Experiment

Since the logic enhanced reasoning method has been proven effective on improving commonsense inference with the benefit of logical chain and indirect inference, we hope to test the limitation of our proposed method. To this end, we conduct a multi-step reasoning experiment on ATOMIC to test whether a longer logical chain

Table 6: BLEU score and human evaluation results of multi-step reasoning experiments.

Steps	xIntent	xNeed	xAttr	xEffect	xReact	xWant	oEffect	oReact	oWant	Fluency	Rationality	Consistency
1-step	12.68	15.25	6.95	11.05	8.21	15.19	7.85	7.21	13.68	4.15	2.81	2.23
2-step	13.76	19.73	7.67	13.25	10.91	15.37	12.62	7.94	16.04	4.20	2.87	3.07
4-step	14.38	19.24	8.94	13.68	**11.25**	16.94	**14.27**	8.58	16.16	**4.33**	2.95	3.34
6-step	**15.15**	**20.56**	**9.96**	**15.45**	10.91	**18.25**	13.98	**8.87**	**17.37**	4.31	**3.09**	**3.57**
9-step	12.85	17.07	7.38	12.2	8.54	15.45	12.05	7.57	12.37	4.21	2.78	3.32

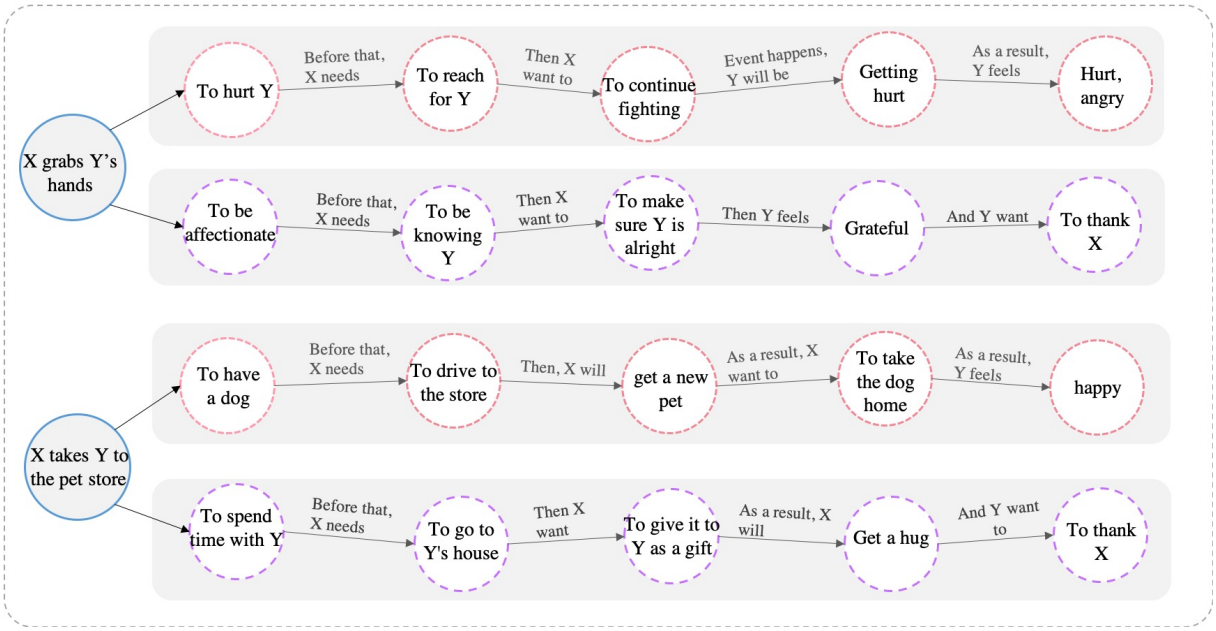

Figure 5: Two samples from the validation set of ATOMIC with the inferences on different dimensions generated by our model. The inferences in the same line are generated following the same logical chain.

(more indirect clues) can lead to a better inference. We design some different logical chain setups of Chain Transformer for comparison:

1-step means reasoning about nine dimensions independently from the event directly (event » all, the same as 9ENC9DEC).

2-step means reasoning about "xIntent" first then other dimensions of the event (event » xIntent » others).

4-step means reasoning about the causes of the event then others (xevent » causes (xIntent » xNeed » xAttr) » others).

6-step means reasoning about the causes of the event then effects of the event, which is illustrated in Section 4.1 (Training Details).

9-step represents that all dimensions are reasoned about one by one following the timeline of the event development (event » xIntent » xNeed » xAttr » xEffect » xReact » xWant » oEffect » oReact » oWant).

Table 6 shows the performances of our model with different logical chain setups. The results show that indirect inference with a certain number of steps is able to improve the performance of the neural model on commonsense inference on both automatic and human evaluation, which also demonstrates the effectiveness of the logic enhanced reasoning method. But the performance would decay with too many steps because more circumstantial clues would cause more serious error propagation during the step-by-step reasoning process.

4.7 Case Study

Figure 5 shows two samples from the test set of ATOMIC with the inferences reasoned by Chain Transformer. From the given event "X takes Y to the per store", Chain Transformer first reasons about that X's intent would be "to have a dog" or "to spend time with Y" with the first decoder. Then with a different logical starting point, Chain Transformer then generates different inference results on other dimensions following different logical chains. If the intent is "to have a dog", then our model reasons about that X needs "to get money, to drive to the store". Further, other dimensions are reasoned out following this chain. On the contrary, following the second logical chain, the results are totally different. The other sample also illustrates the same conclusion.

Different from the previous neural models that use beam search on each dimension to generate diverse inference results, we only use beam search on the first dimension (here we choose "xIntent" as the first inference dimension) and then reason about other dimensions based on the event as well as the reasoned inferences following the

logical chains. Therefore our model can not only generate diverse inferences but also guarantee the inference results on different dimensions are logically consistent. These two cases show that our model can reason and generate more accurate, reasonable and logically consistent inferences on different dimensions.

5 CONCLUSION

In this work, we explore the indirect relations between different commonsense dimensions and present a novel logic enhanced reasoning method that can leverage both direct and indirect inference to construct a logical chain to guide the reasoning. We design Chain Transformer for more logically consistent commonsense inference, where a chain of decoders are employed to reason about different commonsense dimensions one by one following the logical chain. Further, we pretrain our model on external commonsense corpora to learn background commonsense knowledge for better understanding and representing the events and to learn the ability of indirect inference. Experiments on two real-world datasets show that our model outperforms all previous models on both automatic and human evaluations, and demonstrate that our model can generate more accurate, reasonable and logically coherent results.

ACKNOWLEDGMENTS

This work was supported by NSFC project Grant No. U1833101, SZSTI under Grant No. JCYJ20190809172201639 and the Joint Research Center of Tencent and Tsinghua.

REFERENCES

[1] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. In *ACL*. Florence, Italy, 4762–4779. https://doi.org/10.18653/v1/P19-1470

[2] Nathanael Chambers and Daniel Jurafsky. 2008. Unsupervised Learning of Narrative Event Chains.. In *ACL*, Kathleen R. McKeown, Johanna D. Moore, Simone Teufel, James Allan, and Sadaoki Furui (Eds.). The Association for Computer Linguistics, 789–797. http://dblp.uni-trier.de/db/conf/acl/acl2008.html#ChambersJ08

[3] Boxing Chen and Colin Cherry. 2014. A Systematic Comparison of Smoothing Techniques for Sentence-Level BLEU.. In *WMT@ACL*. The Association for Computer Linguistics, 362–367. http://dblp.uni-trier.de/db/conf/wmt/wmt2014.html#ChenC14

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.. In *NAACL-HLT (1)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. http://dblp.uni-trier.de/db/conf/naacl/naacl2019-1.html#DevlinCLT19

[5] Li Du, Xiao Ding, Ting Liu, and Zhongyang Li. 2019. Modeling Event Background for If-Then Commonsense Reasoning Using Context-aware Variational Autoencoder. 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)*. Association for Computational Linguistics, Hong Kong, China, 2682–2691. https://doi.org/10.18653/v1/D19-1270

[6] Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 889–898. https://doi.org/10.18653/v1/P18-1082

[7] Markus Freitag and Yaser Al-Onaizan. 2017. Beam Search Strategies for Neural Machine Translation.. In *NMT@ACL*, Thang Luong, Alexandra Birch, Graham Neubig, and Andrew M. Finch (Eds.). Association for Computational Linguistics, 56–60. http://dblp.uni-trier.de/db/conf/aclnmt/aclnmt2017.html#FreitagA17

[8] Andrew S. Gordon. 2016. Commonsense Interpretation of Triangle Behavior. In *AAAI*, Dale Schuurmans and Michael P. Wellman (Eds.). AAAI Press, 3719–3725. http://dblp.uni-trier.de/db/conf/aaai/aaai2016.html#Gordon16

[9] Mark Granroth-Wilding and Stephen Clark. 2016. What Happens Next? Event Prediction Using a Compositional Neural Network Model.. In *AAAI*, Dale Schuurmans and Michael P. Wellman (Eds.). AAAI Press, 2727–2733. http://dblp.uni-trier.de/db/conf/aaai/aaai2016.html#Granroth-Wilding16

[10] Ting-Hao (Kenneth) Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross B. Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh, Lucy Vanderwende, Michel Galley, and Margaret Mitchell. 2016. Visual Storytelling. *CoRR* abs/1604.03968 (2016). http://dblp.uni-trier.de/db/journals/corr/corr1604.html#HuangFMMADGHKBZ16

[11] Yuqian Jiang, Nick Walker, Justin Hart, and Peter Stone. 2019. Open-World Reasoning for Service Robots. In *Proceedings of the 29th International Conference on Automated Planning and Scheduling (ICAPS 2019)* (Berkeley, CA, USA).

[12] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. http://arxiv.org/abs/1412.6980 cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[13] Zhongyang Li, Xiao Ding, and Ting Liu. 2018. Constructing Narrative Event Evolutionary Graph for Script Event Prediction.. In *IJCAI*, Jérôme Lang (Ed.). ijcai.org, 4201–4207. http://dblp.uni-trier.de/db/conf/ijcai/ijcai2018.html#LiDL18

[14] Dongcai Lu, Shiqi Zhang, Peter Stone, and Xiaoping Chen. 2017. Leveraging Commonsense Reasoning and Multimodal Perception for Robot Spoken Dialog Systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Vancouver, Canada).

[15] 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 NAACL-HLT 2016*. Association for Computational Linguistics, San Diego, California, 839–849. https://doi.org/10.18653/v1/N16-1098

[16] Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James Allen. 2017. LSDSem 2017 Shared Task: The Story Cloze Test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*. Association for Computational Linguistics, Valencia, Spain, 46–51. https://doi.org/10.18653/v1/W17-0906

[17] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 2227–2237. https://doi.org/10.18653/v1/N18-1202

[18] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. (2018).

[19] Hannah Rashkin, Antoine Bosselut, Maarten Sap, Kevin Knight, and Yejin Choi. 2018. Modeling Naive Psychology of Characters in Simple Commonsense Stories.. In *ACL (1)*, Iryna Gurevych and Yusuke Miyao (Eds.). Association for Computational Linguistics, 2289–2299. http://dblp.uni-trier.de/db/conf/acl/acl2018-1.html#KnightCSRB18

[20] Hannah Rashkin, Maarten Sap, Emily Allaway, Noah A. Smith, and Yejin Choi. 2018. Event2Mind: Commonsense Inference on Events, Intents, and Reactions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*. Association for Computational Linguistics, Melbourne, Australia.

[21] Stefano Rosa, Andrea Patanè, Xiaoxuan Lu, and Niki Trigoni. 2018. CommonSense: Collaborative learning of scene semantics by robots and humans.. In *IoPARTS@MobiSys*. ACM, 1–6. http://dblp.uni-trier.de/db/conf/mobisys/ioparts2018.html#RosaPLT18

[22] Maarten Sap, Ronan Lebras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. ATOMIC: An Atlas of Machine Commonsense for If-Then Reasoning. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Inteligence (AAAI-19)*. AAAI Press, Honolulu, HI, USA.

[23] Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17) and the Twenty-Ninth Innovative Applications of Artificial Intelligence Conference (IAAI-17)*. AAAI Press, San Francisco, CA, USA.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[25] Su Wang, Rahul Gupta, Nancy Chang, and Jason Baldridge. 2019. A Task in a Suit and a Tie: Paraphrase Generation with Semantic Augmentation. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Inteligence (AAAI-19)*. AAAI Press, Honolulu, HI, USA.

[26] Zhongqing Wang, Yue Zhang, and Ching-Yun Chang. 2017. Integrating Order Information and Event Relation for Script Event Prediction.. In *EMNLP*, Martha Palmer, Rebecca Hwa, and Sebastian Riedel (Eds.). Association for Computational Linguistics, 57–67. http://dblp.uni-trier.de/db/conf/emnlp/emnlp2017.html#WangZC17

[27] Haoyu Zhang, Yeyun Gong, Yu Yan, Nan Duan, Jianjun Xu, Ji Wang, Ming Gong, and Ming Zhou. 2019. Pretraining-Based Natural Language Generation for Text Summarization. *Computing Research Repository* arXiv: 1902.09243 (2019). http://dblp.uni-trier.de/db/journals/corr/corr1902.html#abs-1902-09243