Story Ending Generation with Incremental Encoding and Commonsense Knowledge

AAAI 2019

Jian Guan, Yansen Wang, Minlie Huang

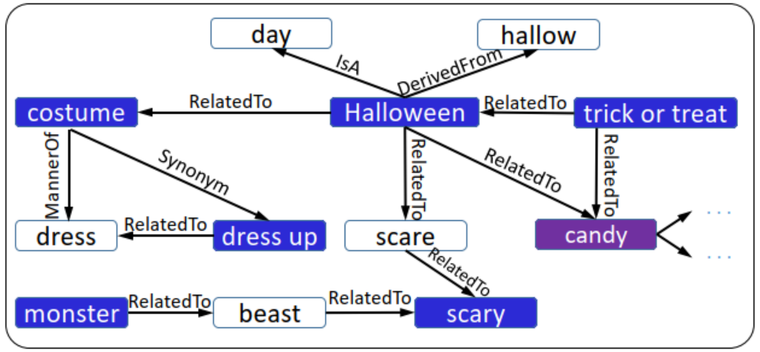
Outline

Introduction

- Story ending generation task
  - $\hat{Y} = argmax_Y(p_\theta(Y \mid X_1,\ X_2,\ \dots X_K))$
  - The formula is similar to multi-turn dialogue
- Commonsense knowledge
  - triple *R = (h, r, t)*
  - head concept h has the relation r with tail concept t

Today is **Halloween** .
Jack is so excited to go **trick or treating** tonight .
He is going to **dress up** like a **monster** .
The **costume** is real **scary** .

He hopes to get a lot of **candy** .



Motivation

- Deciding a reasonable ending not only depends on representing the **context** clues properly, but also on the ability of language understanding with **implicit knowledge** that is beyond the text surface.

Back to Outline

Motivation

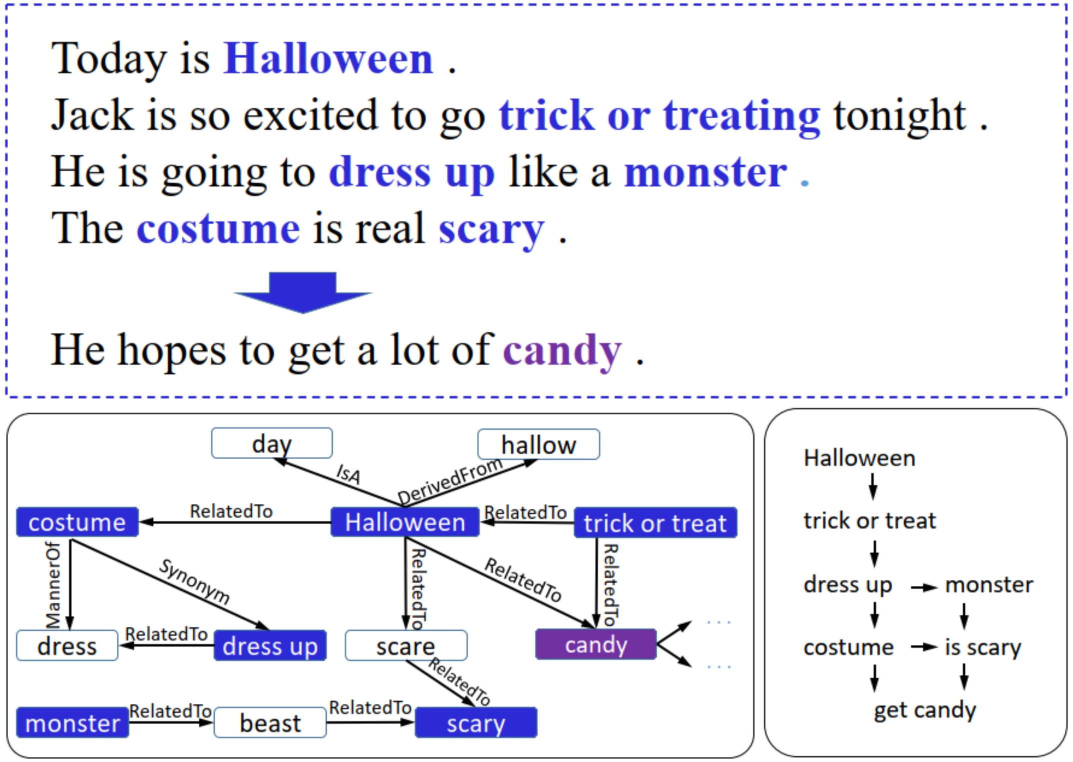


Figure 1: A story example. Words in blue/purple are events and entities. The bottom-left graph is retrieved from ConceptNet and the bottom-right graph represents how events and entities form the context clue.

No clue about “candy” In the context.

Halloween is related to “candy” in commonsense.

Generates ending about “candy” after incorporating commonsense knowledge into the model.

Method

- Incremental encoding (IE) scheme
- Multi-source attention (MAS) mechanism
- Supervision on the encoding network

Back to Outline

Method

- Incremental encoding (IE) scheme

$$\mathbf{h}_j^{(i)} = \mathbf{LSTM}(\mathbf{h}_{j-1}^{(i)}, \boldsymbol{e}(x_j^{(i)}), \mathbf{c}_{\mathbf{l}j}^{(i)}),\ i \geq 2. \tag{5}$$

where $\mathbf{h}_j^{(i)}$ denotes the hidden state at the j-th position of the i-th sentence, $\boldsymbol{e}(x_j^{(i)})$ denotes the word vector of the j-th word $x_j^{(i)}$. $\mathbf{c}_{\mathbf{l},j}^{(i)}$ is the context vector which is an attentive read of the *preceding* sentence X_{i-1}, conditioned on $\mathbf{h}_{j-1}^{(i)}$.

Method

- Multi-source attention (MSA) mechanism

$$\mathbf{c}_{\mathbf{l}j}^{(i)} = \mathbf{W}_{\mathbf{l}}([\mathbf{c}_{\mathbf{h}j}^{(i)}; \mathbf{c}_{\mathbf{x}j}^{(i)}]) + \mathbf{b}_{\mathbf{l}}, \tag{7}$$

where $\oplus$ indicates vector concatenation. Hereafter, $\mathbf{c}_{\mathbf{h}j}^{(i)}$ is called *state context vector*, and $\mathbf{c}_{\mathbf{x}j}^{(i)}$ is called *knowledge context vector*.

Method

- Multi-source attention (MSA) mechanism
  - State context vector
    - Simple attention mechanism for seq2seq
  - Knowledge context vector
    - 1) graph attention (Velickovic et al. 2018; Zhou et al. 2018)
    - 2) contextual attention (Mihaylov and Frank 2018).

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. Graph attention networks. In ICLR.
Zhou, H.; Yang, T.; Huang, M.; Zhao, H.; Xu, J.; and Zhu, X. 2018. Commonsense knowledge aware conversation generation with graph attention. In IJCAI.
Mihaylov, T., and Frank, A. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In ACL, 821–832.

Back to Outline

Method

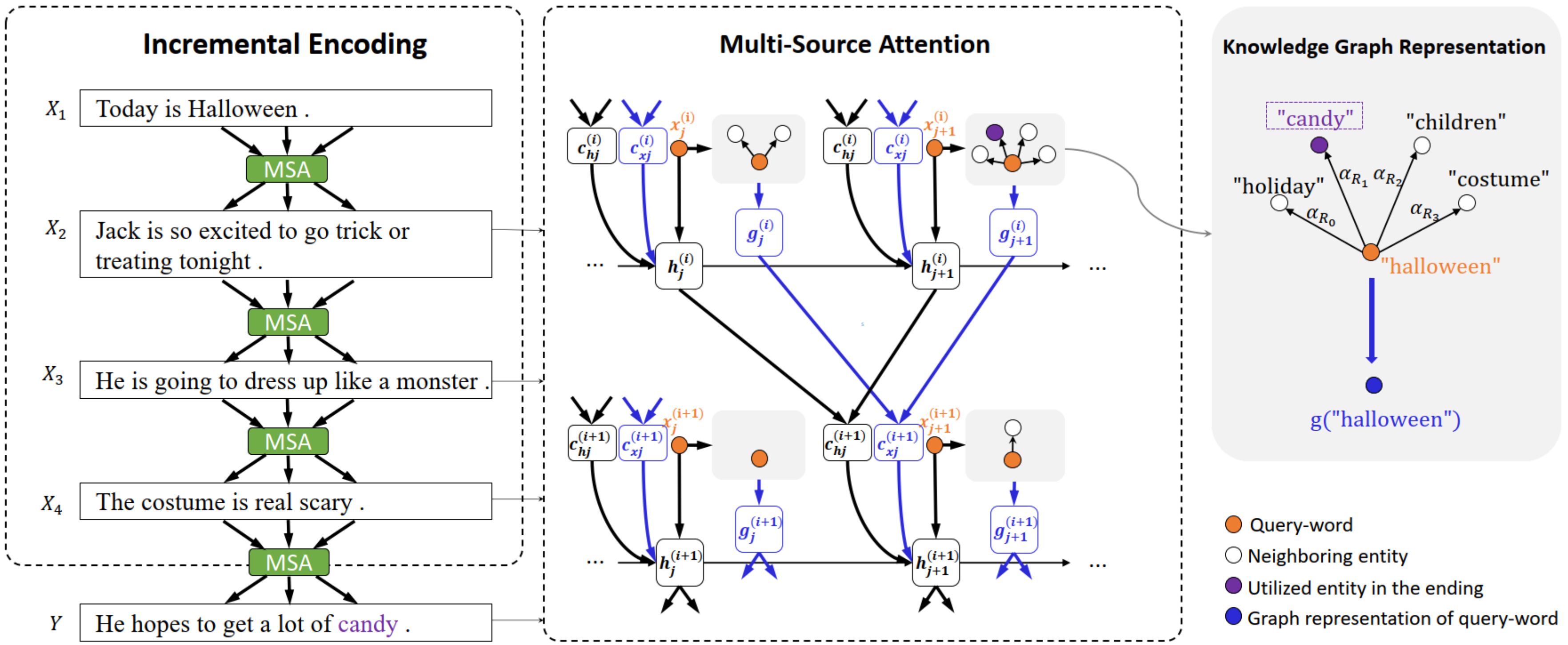


Figure 2: Model overview. The model is equipped with incremental encoding (IE) and multi-source attention (MSA). $x_j^{(i)}$: the j-th word in sentence i; $\mathbf{c}_{hj}^{(i)}$: state context vector; $\mathbf{c}_{xj}^{(i)}$: knowledge context vector; $\mathbf{g}_j^{(i)}$: graph vector of word $x_j^{(i)}$; $\mathbf{h}_j^{(i)}$: j-th hidden state of sentence i. The state (knowledge) context vectors are attentive read of hidden states (graph vectors) in the preceding sentence.

[Back to Outline](#)

Method

- Supervision on the encoding network

At each encoding step, we also generate a distribution over the vocabulary, very similar to the decoding process:

$$\mathcal{P}(y_t|y_{<t}, X) = \mathbf{softmax}(\mathbf{W}_0\mathbf{h}_j^{(i)} + \mathbf{b}_0), \qquad (21)$$

Back to Outline

Method

Then, we calculate the negative data likelihood as loss function:

$$\Phi = \Phi_{en} + \Phi_{de} \tag{22}$$

$$\Phi_{en} = \sum_{i=2}^{K} \sum_{j=1}^{l_i} -\log \mathcal{P}(x_j^{(i)} = \widetilde{x}_j^{(i)} | x_{<j}^{(i)}, X_{<i}), \tag{23}$$

$$\Phi_{de} = \sum_{t} -\log \mathcal{P}(y_t = \tilde{y}_t | y_{<t}, X), \tag{24}$$

Back to Outline

Experiment

- Dataset: ROCStories corpus
- Automatic metric: PPL and BLEU
- Human evaluation: grammar and logicality

Back to Outline

Experiment

- Models
  - Baselines
    - Seq2seq
    - Hierarchical LSTM (HLSTM)
    - HLSTM + copy
    - HLSTM + MSA
  - This paper
    - IE
    - IE + MSA

Back to Outline

Experiment

Model	PPL	BLEU-1	BLEU-2	Gram.	Logic.
Seq2Seq	18.97	0.1864	0.0090	1.74	0.70
HLSTM	17.26	0.2459	0.0242	1.57	0.84
HLSTM+Copy	19.93	0.2469	0.0248	1.66	0.90
HLSTM+MSA(GA)	15.75	0.2588	0.0253	1.70	1.06
HLSTM+MSA(CA)	12.53	0.2514	0.0271	1.72	1.02
IE (ours)	11.04	0.2514	0.0263	**1.84**	1.10
IE+MSA(GA) (ours)	9.72	0.2566	0.0284	1.68	**1.26**
IE+MSA(CA) (ours)	**8.79**	**0.2682**	**0.0327**	1.66	1.24

Table 1: Automatic and manual evaluation results.

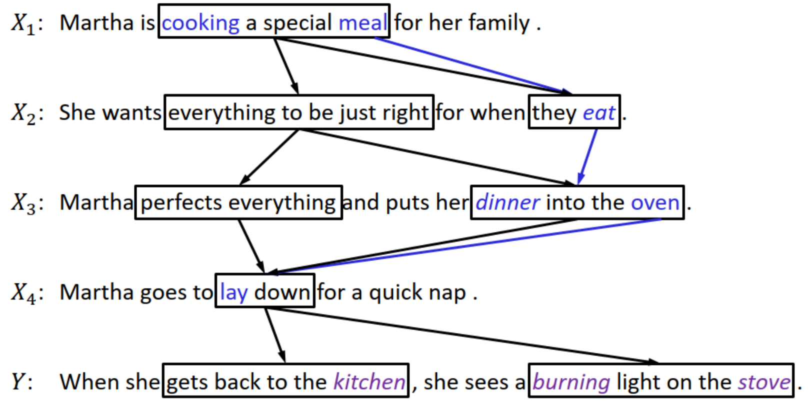
Gram.-Logic. Score	2-2	2-1	1-2	1-1
Seq2seq	20.0%	22.0%	6.5%	1.5%
HLSTM	21.0%	17.0%	10.0%	3.5%
HLSTM+Copy	28.0%	19.0%	7.0%	5.5%
HLSTM+MSA(GA)	33.5%	25.0%	5.0%	4.0%
HLSTM+MSA(CA)	30.0%	26.0%	2.0%	8.0%
IE (ours)	36.0%	**34.0%**	2.0%	4.0%
IE+MSA(GA) (ours)	**45.0%**	24.0%	5.0%	2.0%
IE+MSA(CA) (ours)	41.0%	27.0%	4.0%	2.0%

Table 2: Data distribution over Gram.-Logic. scores. a-b denotes that the grammar score is a and the logicality score is b. Each cell denotes the proportion of the endings with score a-b.

Experiment

Context:	Martha is **cooking** a special **meal** for her family. She **wants everything to be just right** for when they eat. Martha **perfects everything** and puts her **dinner** into the **oven**. Martha goes to **lay down** for a quick **nap**.
Golden Ending:	She **oversleeps** and runs into the **kitchen** to take out her **burnt dinner**.
Seq2Seq:	She was so happy to have a ***new cake***.
HLSTM:	Her family ***and her family*** are very happy with her **food**.
HLSTM+ Copy:	**Martha** is happy to be able to ***eat her family***.
HLSTM+ GA:	She is happy to be able to **cook her dinner**.
HLSTM+ CA:	She is very happy that she has made a new **cook** .
IE:	She is very happy with her **family**.
IE+GA:	When she gets back to the **kitchen**, she sees a **burning light** on the **stove**.
IE+CA:	She realizes the **food** and is happy she was ready to **cook** .

Table 3: Generated endings from different models. **Bold** words denote the **key** entity and event in the story. *Improper* words in ending is in *italic* and proper words are underlined.



Entity	commonsense knowledge
cook	(cook, AtLocation, *kitchen*)
	(cook, HasLastSubevent, *eat*)
meal	(meal, AtLocation, *dinner*)
	(meal, RelatedTo, *eat*)
eat	(eat, AtLocation, *dinner*)
oven	(oven, AtLocation, *stove*)
	(oven, RelatedTo, *kitchen*)
	(oven, UsedFor, *burn*)

Figure 3: An example illustrating how incremental encoding builds connections between context clues.

Back to Outline

Conclusion

- Results:
  - For imposing supervision on the encoding network, the paper claims that "experiments show that it is better in logic than merely imposing supervision on the decoding network". **But the paper does not present this experiment**.
  - Multi-source attention leads to generate story endings that have **more overlaps** with the reference endings.
  - Incremental encoding (IE) is effective.
  - Using commonsense knowledge (IE + MSA) leads to significant improvements in **logicality**.
  - HLSTM equipped with MSA is better than those without MSA, indicating that commonsense knowledge is helpful.