Story Ending Generation with Incremental Encoding and Commonsense Knowledge

Jian Guan*, Yansen Wang*, Minlie Huang†
Dept. of Computer Science & Technology, Tsinghua University, Beijing 100084, China
Institute for Artificial IntelligenceTsinghua University (THUAI), China
Beijing National Research Center for Information Science and Technology, China
guanj15@mails.tsinghua.edu.cn;ys-wang15@mails.tsinghua.edu.cn;
aihuang@tsinghua.edu.cn

Story Ending Generation Tasks

- Given a story context
- Conclude the story and complete the plot

Context: 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 .

Ending : He hopes to get a lot of candy .

Story Ending Generation Tasks

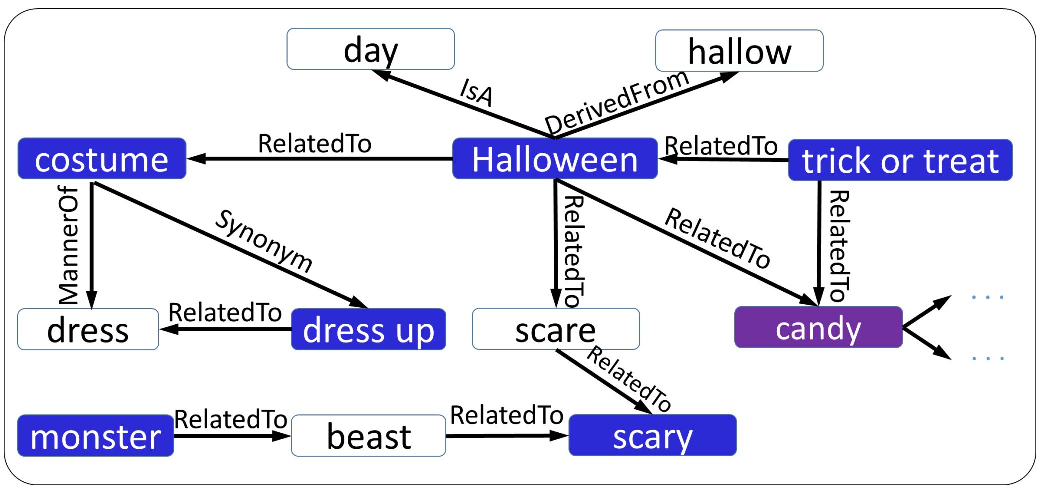
Generating a good ending requires:

- Representing the **context clues** which contain key information for planning a reasonable ending
- Using **implicit knowledge** (e.g., commonsense knowledge) to facilitate understanding of the story and better predict what will happen next.

Story Ending Generation Tasks

Context: 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** .

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



Implicit Knowledge

Context Clues

Task Overview

- Given a story context consisting of a sentence sequence:

$$X = \{X_1, X_2, X_2, \ldots, X_K\}, \text{ where } X_i = x_1^{(i)} x_2^{(i)} \ldots x_{l_i}^{(i)}$$

- The model should generate a one-sentence ending:

$$Y = y_1 y_2 \ldots y_l$$

- Formally:

$$Y^* = \underset{Y}{argmax}\, \mathcal{P}(Y|X).$$

Background

Sequence to Sequence:

- Encoder：

$$\mathbf{h}_t = \mathbf{LSTM}(\mathbf{h}_{t-1}, \boldsymbol{e}(x_t)),$$

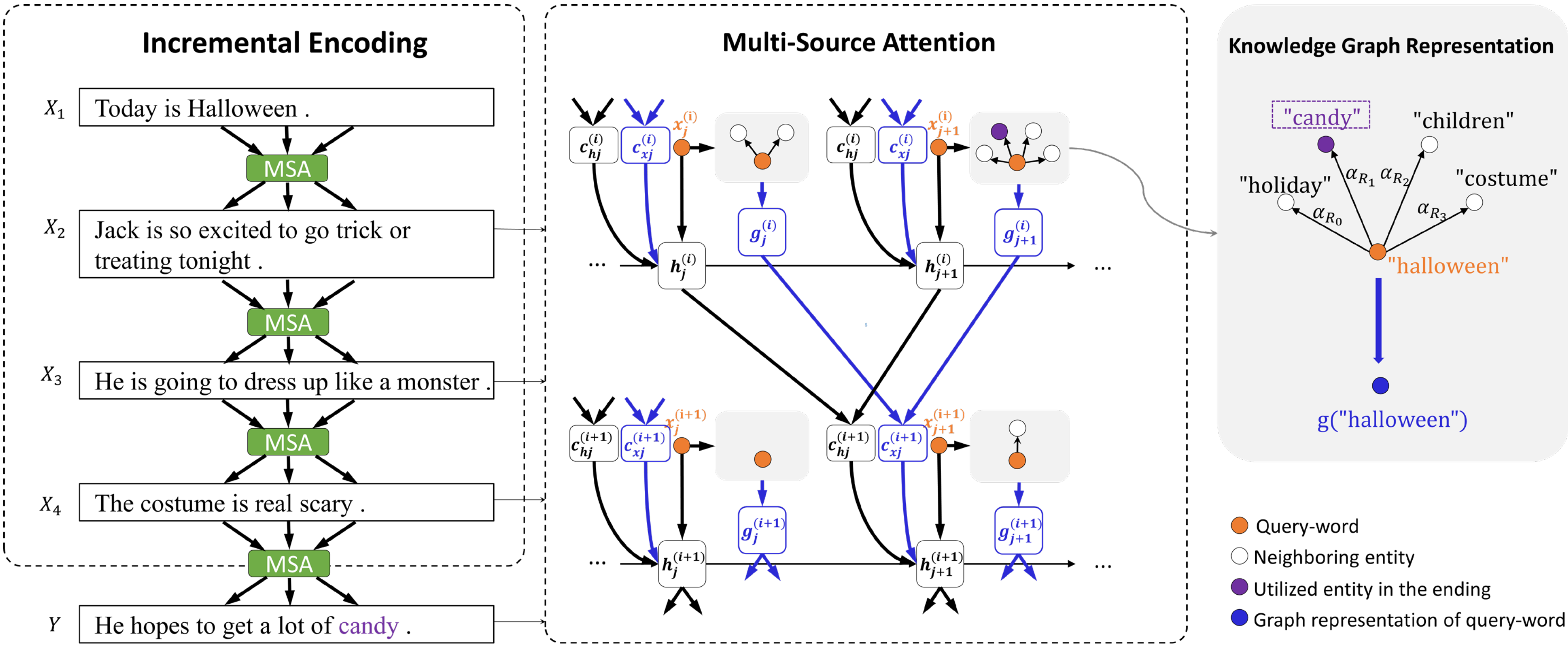
- Decoder:

$$\mathcal{P}(y_t|y_{<t}, X) = \mathbf{softmax}(\mathbf{W}_0 \boldsymbol{s}_t + \mathbf{b}_0),$$
$$\mathbf{s}_t = \mathbf{LSTM}(\mathbf{s}_{t-1}, \boldsymbol{e}(y_{t-1}), \mathbf{c}_{t-1}),$$

$\mathbf{c}_{t-1}$ in the decoder is an attentive read of the encoder states.

$$\mathbf{c}_{t-1} = \textstyle\sum_{i=1}^{m} \alpha_{(t-1)i} \mathbf{h}_i,$$

Model Overview



Model

Encode the story context

- Concatenating the K sentences to a long sentence and encoding it with an LSTM
- Using a hierarchical LSTM with hierarchical attention (Yang et al. 2016)
- Incremental Encoding

Model

Incremental Encoding

- Effective to represent the context clues which may **capture the key logic information.**
- When encoding the current sentence X_i, it obtains a context vector which is **an attentive read of the preceding sentence** $X_{i\text{-}1}$:

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

- During the decoding process, the decoder obtains **a context vector from the last sentence** X_K in the context to utilize the context clues:

$$\mathbf{s}_t = \mathbf{LSTM}(\mathbf{s}_{t-1}, \boldsymbol{e}(y_{t-1}), \mathbf{c}_{\mathbf{l}t}),$$

$$\mathcal{P}(y_t|y_{<t}, X) = \mathbf{softmax}(\mathbf{W}_0 \boldsymbol{s}_t + \mathbf{b}_0),$$

Model

Context vector

- Capture the relationship between words (or states) in the current sentence and those in the preceding sentence
- Contains implicit knowledge that is beyond the text
- Formally: $\mathbf{c}_{\mathbf{l}j}^{(i)} = \mathbf{W}_{\mathbf{l}}([\mathbf{c}_{\mathbf{h}j}^{(i)}; \mathbf{c}_{\mathbf{x}j}^{(i)}]) + \mathbf{b}_{\mathbf{l}},$
  - $\mathbf{c}_{\mathbf{h}j}^{(i)}$ is called **state context vector**
  - $\mathbf{c}_{\mathbf{x}j}^{(i)}$ is called **knowledge context vector**

Model

Context vector

- **state context vector**

$$\mathbf{c}_{\mathbf{h}j}^{(i)} = \sum_{k=1}^{l_{i-1}} \alpha_{h_k,j}^{(i)} \mathbf{h}_k^{(i-1)},$$

$$\alpha_{h_k,j}^{(i)} = \frac{e^{\beta_{h_k,j}^{(i)}}}{\sum_{m=1}^{l_{i-1}} e^{\beta_{h_m,j}^{(i)}}},$$

$$\beta_{h_k,j}^{(i)} = \mathbf{h}_{j-1}^{(i)\mathrm{T}} \mathbf{W_s} \mathbf{h}_k^{(i-1)},$$

- **knowledge context vector**

$$\mathbf{c}_{\mathbf{x}j}^{(i)} = \sum_{k=1}^{l_{i-1}} \alpha_{x_k,j}^{(i)} \mathbf{g}(x_k^{(i-1)}),$$

$$\alpha_{x_k,j}^{(i)} = \frac{e^{\beta_{x_k,j}^{(i)}}}{\sum_{m=1}^{l_{i-1}} e^{\beta_{x_m,j}^{(i)}}},$$

$$\beta_{x_k,j}^{(i)} = \mathbf{h}_{j-1}^{(i)\mathrm{T}} \mathbf{W_k} \mathbf{g}(x_k^{(i-1)}),$$

Model

Knowledge graph retrieval

- ConceptNet
  - A commonsense semantic network
  - Consists of triples $\boldsymbol{R} = (\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$ meaning that head concept $\boldsymbol{h}$ has the relation $\boldsymbol{r}$ with tail concept $\boldsymbol{t}$
    - e.g. (*costume*, */R/MannerOf*, *dress*)
  - Each word in a sentence is used as a query to **retrieve a one-hop graph** from ConceptNet.

Model

Knowledge graph representation

- The knowledge graph for a word extends (encodes) its meaning by **representing the graph** from neighboring concepts and relations.
  - Graph Attention (Velikovi et al. 2018; Zhou et al. 2018)
  - Contextual attention (Mihaylov and Frank 2018)

Model

Knowledge graph representation

- Graph Attention

$$\mathbf{g}(x) = \sum_{i=1}^{N_x} \alpha_{R_i} [\mathbf{h}_i; \mathbf{t}_i],$$

$$\alpha_{R_i} = \frac{e^{\beta_{R_i}}}{\sum_{j=1}^{N_x} e^{\beta_{R_j}}},$$

$$\beta_{R_i} = (\mathbf{W_r r}_i)^{\mathrm{T}} \, tanh(\mathbf{W_h h}_i + \mathbf{W_t t}_i),$$

- Contextual Attention

$$\mathbf{g}(x) = \sum_{i=1}^{N_x} \alpha_{R_i} \mathbf{M}_{R_i},$$

$$\mathbf{M}_{R_i} = BiGRU(\mathbf{h}_i, \mathbf{r}_i, \mathbf{t}_i),$$

$$\alpha_{R_i} = \frac{e^{\beta_{R_i}}}{\sum_{j=1}^{N_x} e^{\beta_{R_j}}},$$

$$\beta_{R_i} = \mathbf{h}_{(x)}^{\mathrm{T}} \mathbf{W_c M}_{R_i},$$

Model

Loss Function

- To better model the chronological order and causal relationship between adjacent sentences, we **impose supervision on both the encoding network and decoding network:**

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

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

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

- The parameters of the LSTMs are **shared** by the encoder and the decoder: **data augmentation.**

Experiments

Resources

- ROCStories corpus
  - Each story consists of **five sentences**, our task is to generate the ending given the first 4 sentence
  - 90,000 for training and 8,162 for evaluation
  - Average length of $X_1/X_2/X_3/X_4/Y$ is 8.9/9.9/10.1/10.0/10.5

- Concept Net
  - Only retrieve the relations whose head entity and tail entity are **noun or verb**, meanwhile **both occurring in SCT**.
  - Retain at most 10 triples if there are too many for a word.
  - Average number of triples for each query word is 3.4

Experiments

Evaluation

- Automatic Evaluation
  - Perplexity, BLEU-1 and BLEU-2
    - How well a model fits the data

- Manual Evaluation
  - Grammar (Gram.)
    - Score 2 : without any grammar errors
    - Score 1 : with a few errors but still understandable
    - Score 0 : with severe errors and incomprehensible
  - Logicality (Logic.)
    - Score 2 : totally reasonable endings
    - Score 1 : relevant but with some discrepancy
    - Score 0 : totally incompatible endings

Experiments

Evaluation result

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.

Experiments

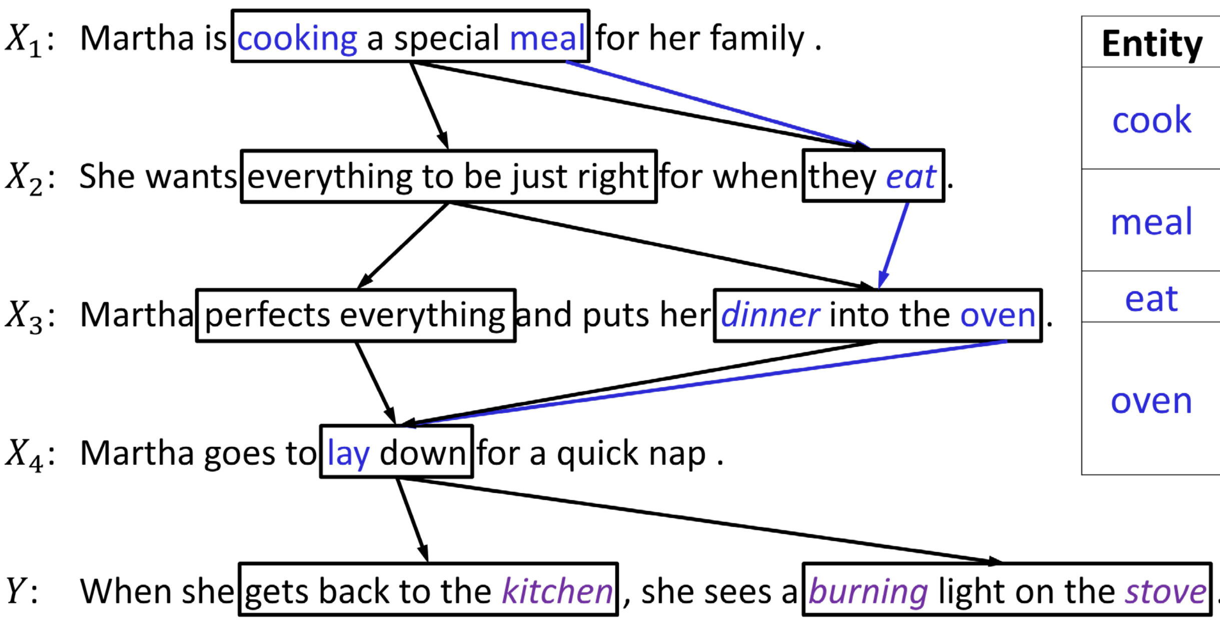
Case study

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.

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 <u>**oversleeps**</u> and runs into the <u>**kitchen**</u> to take out her <u>**burnt dinner**</u>.
Seq2Seq:	She was so happy to have a ***new cake***.
HLSTM:	Her family ***and her family*** are very happy with her <u>**food**</u>.
HLSTM+ Copy:	<u>**Martha**</u> is happy to be able to ***eat her family***.
HLSTM+ GA:	She is happy to be able to <u>**cook her dinner**</u>.
HLSTM+ CA:	She is very happy that she has made a new <u>**cook**</u> .
IE:	She is very happy with her <u>**family**</u>.
IE+GA:	When she gets back to the <u>**kitchen**</u>, she sees a <u>**burning light**</u> on the <u>**stove**</u>.
IE+CA:	She realizes the <u>**food**</u> and is happy she was ready to <u>**cook**</u> .

Experiments

Case study



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.

Summary

- Effective representation and utilization of **context clues** and **implicit knowledge** contributes to a reasonable story ending
- Addressing the problem to generate story ending from the perspective of logicality
- Still a long way to go:
  - Extended to the whole story generation?
  - Applied to other tasks e.g. multi-turn conversational system?

Thanks for your attention!

Any questions?