A Skeleton-Based Model for Promoting Coherence Among Sentences in Narrative Story Generation

Jingjing Xu, Xuancheng Ren, Yi Zhang, Qi Zeng, Xiaoyan Cai, Xu Sun

MOE Key Lab of Computational Linguistics, School of EECS, Peking University
School of Automation, Northwestern Polytechnical University
jingjingxu@pku.edu.cn

Outline

- Introduction
  - Task
  - Challenge
- Skeleton-Based Model
  - Overview
  - Skeleton-Based Generative Module
  - Skeleton Extraction Module
- Experiment
  - Dataset
  - Baselines
  - Results
- Analysis
  - Incremental Analysis
  - Error Analysis
- Conclusion

Introduction

Narrative Story Generation

Task Definition

Input: A short description of a scene or an event.
Output: A relevant narrative story following the input.

Input: Fans came together to celebrate the opening of a new studio for an artist.

Output: The artist provided champagne in flutes for everyone. Friends toasted and cheered the artist as she opened her new studio.

Input: Last week I attended a wedding for the first time.

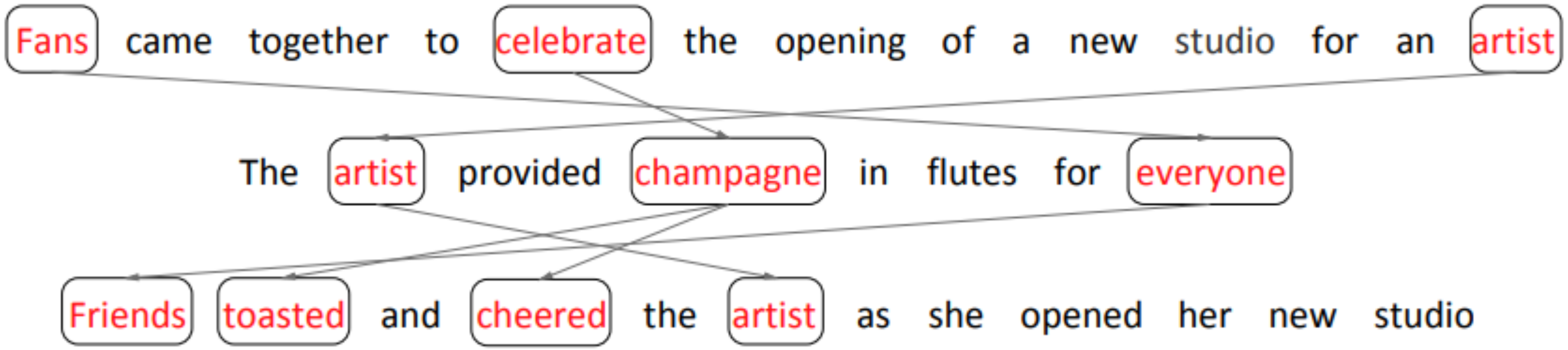
Output: There were a lot of families there. They were all taking pictures together. Everyone was very happy. The bride and groom got to ride in a limo that they rented.

Difficulty

- High requirements on **tight semantic connection** among sentences
- If the given input is about "artist and studio opening"
  - 👍 The artist provided champagne for everyone
  - 👍 Fans toasted and cheered the artist
  - 👍 The fans were very happy to see this event
  - 👎 The football game is so exciting
  - 👎 The weather is so cold and all trees are covered with heavy snow

Motivation

- The connection among sentences is mainly reflected through key phrases (or skeleton)
- We explicitly model the relation among skeletons in this work.



Skeleton-based Model

- Motivated by this fact:

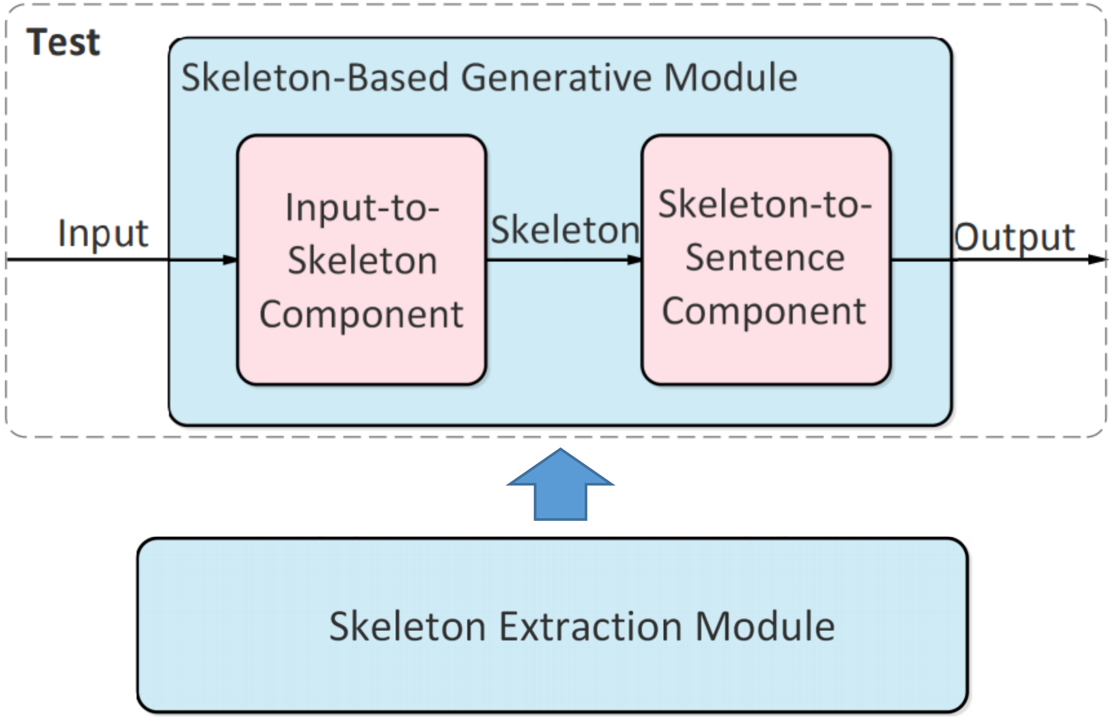
    1. We do not generate a sentence from left to right
    2. Instead, we first generate a skeleton
    3. Expand the skeleton to a complete and fluent sentence

Approach

Overview

❑ Our model contains two parts

- Skeleton-based generative module
  1. Input-to-skeleton component
  2. Skeleton-to-target component
- Skeleton extraction module



Skeleton-Based Generative Module

❑ Generation Process

1. Fans came together to celebrate the opening of a new studio for an artist.

↓

artist champagne

↓

2. The artist provided champagne in flutes for everyone.

↓

Friends toasted cheered artist.

↓

3. Friends toasted and cheered the artist as she opened her new studio.

The Structure of Skeleton-Based Generative Module

- Input-to-Skeleton Component
  - Learn the dependency between source inputs and target skeletons
  - Model: A Seq2Seq
  - Training Loss: Cross-entropy loss
- Skeleton-to-Target Component
  - Expand a skeleton to a complete sentence
  - Model: Seq2Seq
    - Encoder: LSTM
    - Decoder: LSTM
  - Training Loss: Cross-Entropy Loss

Skeleton-Extractive Module

- Skeletons as supervisory signal
  - In real-world datasets, the human-annotated skeleton is usually unavailable
- Rule-based approaches
  - Advantage: Simple
  - Drawback: Difficult to define the unified rules of extracting skeletons
- To address this problem, we build a skeleton extraction module to automatically explore sentence skeletons

Skeleton-Extraction Module

- Reformulate skeleton extraction as a sentence compression problem
  - **Advantage**: sentence compression requires systems to keep core information, which is consistent with our target.
  - **Drawback**: sentence compression results usually are grammatical, which limits the quality of skeletons.

The lady wearing the pink shirt decided to stop playing the video and chatted with other guests.

compress →

Lady decided to stop playing the video

The structure of Skeleton Extraction Module

- ❑ Sequence to sequence model
  - ➢ Encoder: LSTM
  - ➢ Decoder: LSTM
- ❑ Input: original text
- ❑ Output: the skeleton of the original text
- ❑ Loss function: cross-entropy loss

$$L_{\gamma} = -\sum_{i=1}^{T} P_E(s_i|\boldsymbol{x}, \gamma)$$

How to improve the quality of skeleton-extraction module

- Motivation
  - Use the feedback of generative module to improve it
- Challenge
  - Loss is no longer differentiable
- Approach
  - Reinforcement Learning Method
    - Treat the feedback of generative module as reward
    - Optimize the extraction model by

$$\nabla J(\gamma) = \mathbb{E}[R_c \cdot \nabla \log(P_E(\boldsymbol{s}|\boldsymbol{x}), \gamma)]$$

Reward

❑ Multiplication of the cross-entropy loss

- K: Upper bound of the reward,
- R_1: Cross-entropy losses in the input-to-skeleton component
- R_2: Cross-entropy losses in the skeleton-to-target component, respectively.

$$R_c = [K - (R_1 \times R_2)^{\frac{1}{2}}]$$

Experiment

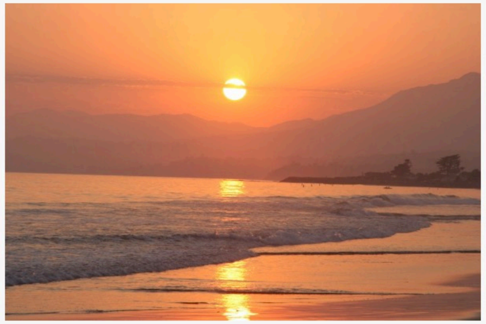
Dataset

❑ Source:

- ✓ Visual Storytelling

❑ Dataset:

- ✓ Input: the first sentence
- ✓ Output: the following sentences



Having a good time bonding and talking.	[M] got exhausted by the heat.	Sky illuminated with a brilliance of gold and orange hues.

Baselines

- ❑ Entity-Enhanced Seq2Seq Model (EESeq2Seq)
  - ✓ Combine entity context and text context together
- ❑ Dependency-Tree Enhanced Seq2Seq Model (DE-Seq2Seq)
  - ✓ Based on dependency parsing labels
- ❑ Generalized-Template Enhanced Seq2Seq Model (GE-Seq2Seq)
  - ✓ Use existing knowledge bases (WordNet, VerbNet) to get a generalized sentence representation

Results

❑ Automatic Evaluation

Models	BLEU
EE-Seq2Seq	0.0029
DE-Seq2Seq	0.0027
GE-Seq2Seq	0.0022
Proposed Model	**0.0042 (+44.8%)**

Human Evaluation

- ✓ 100 items
- ✓ Fluency and coherence

Models	Fluency	Coherence	G-Score
EE-Seq2Seq	6.28	5.14	5.68
DE-Seq2Seq	8.48	3.54	5.48
GE-Seq2Seq	9.48	3.58	5.82
Proposed Model	8.69	5.62	**6.99 (+20.1%)**

Generated examples

- Input 1: [Female] and her friend had a night out on the town.
  - EE-Seq2Seq: They were very happy to see us. [Male] and [female] were getting married today.
  - DE-Seq2Seq: We walked through the streets of the city and found a place to sit. The whole family came out to support their families.
  - GE-Seq2Seq: My friends and I went to a bar last night. [Female] was so happy to be there.
  - Proposed Model: They went to the bar. They had a great time.

Analysis

Error Analysis

❑ 4 types of errors

- ✓ Irrelevant scene
- ✓ Chaotic syntax
- ✓ Chaotic timeline
- ✓ Repeated scene

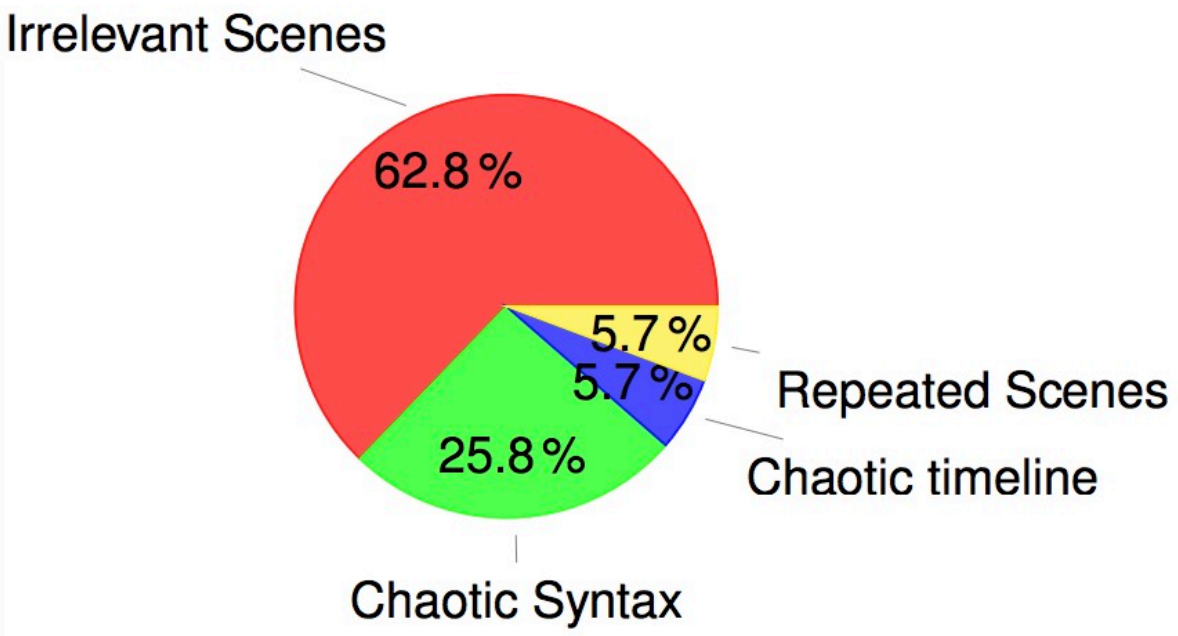


Figure 3: The distribution of error types.

Conclusion

A. We propose a new skeleton-based model for generating coherent narrative stories

B. Experimental results show that our model significantly improves the quality of generated stories

C. Error analysis shows that there are still many challenges in narrative story generation, which we would like to explore in the future

Thank You!

If you have any question, please send an e-mail to jingjingxu@pku.edu.cn