# Plan-And-Write: Towards Better Automatic Storytelling

**Lili Yao**[1,3*], **Nanyun Peng**[2*], **Ralph Weischedel**[2] **Kevin Knight**[2], **Dongyan Zhao**[1] **and Rui Yan**[1†]
liliyao@tencent.com, {npeng,weisched,knight}@isi.edu
{zhaodongyan,ruiyan}@pku.edu.cn
[1] Institute of Computer Science and Technology, Peking University
[2] Information Sciences Institute, University of Southern California, [3] Tencent AI Lab

AAAI 2019

# Story Generation

Two steps:

1. a sensible sequence of events
2. composing coherent natural language texts to describe them

Harder than text generation

# Previous Work

- Most focused on

Two steps:

1. a sensible sequence of events
2. composing coherent natural language texts to describe them

- these approaches rely heavily on human annotation or are restricted to limited domains.

# Introduction

- In poetry compositon, Wang et al.[2016] provides a sequence of words to guide poetry generation.

- In conversational systems, Mou et al.[2016] takes keywords as the main gist of the reply to guide response generation.

- They take a similar approach to represent a story plot with a sequence of words.

# Contributions

Propose a *plan-and-write* generation framework:

- input: title
- first plans a storyline
- then generate a story based on the storyline

Two planning strategies:

- *dynamic schema*
- *static schema*

| | |
|---|---|
| **Title (Given)** | The Bike Accident |
| **Storyline (Extracted)** | Carrie → bike → sneak → nervous → leg |
| **Story (Human Written)** | Carrie had just learned how to ride a bike. She didn't have a bike of her own. Carrie would sneak rides on her sister's bike. She got nervous on a hill and crashed into a wall. The bike frame bent and Carrie got a deep gash on her leg. |

Table 1: An example of title, storyline and story in our system. A storyline is represented by an ordered list of words.

# Problem Formulation

- Input:

  title: $t = \{t_1, t_2, ..., t_n\}$

  where $t_i$ is the *i*-th word in the title

- Output:

  a story: $s = \{s_1, s_2, ..., s_m\}$

  where $s_i$ denotes a sentence

- Storyline:

  a storyline: $l = \{l_1, l_2, ..., l_m\}$

  where $l_i$ denotes a word

# Storyline Preparation

- Extract one word from each sentence of a story to form a storyline
- Method:
  - RAKE algorithm(Rose et al.2010): combines several word frequency based and graph-based metrics to weight the importance of the words
  - extract the most important word from each sentence

# RAKE algorithm

- keyword extraction: unsupervised method
- Document-oriented method: regardless of the current state of a corpus
- Method:
  - text -> a set of candidate keywords
    split by phrase delimiters and stop words
  - compute scores based on word co-occurrences

$$\frac{deg(w)}{freq(w)}$$

  - select top T scoring keywords

# Method

- *Dynamic Schema*: adjust the plot while writing progresses.
- *Static Schema*: plans the entire plot before writing.

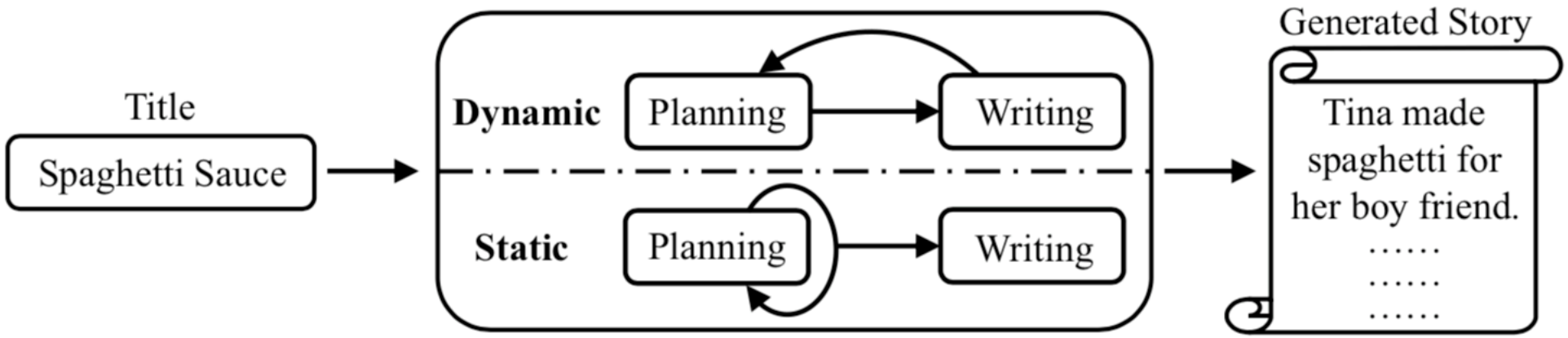
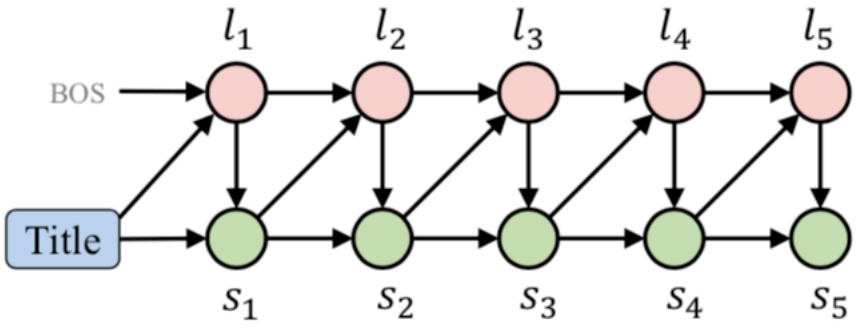



Figure 1: An overview of our system.

# Dynamic Schema

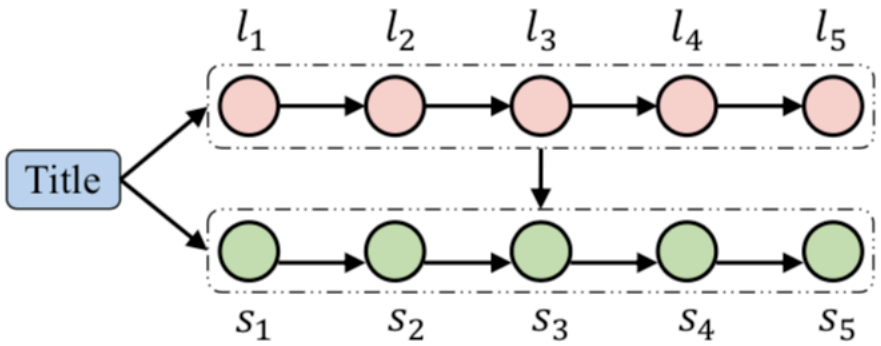
- It generates the next word in the storyline and the next sentence in the story at each step.




(a) Dynamic schema work-flow.

# Static Schema

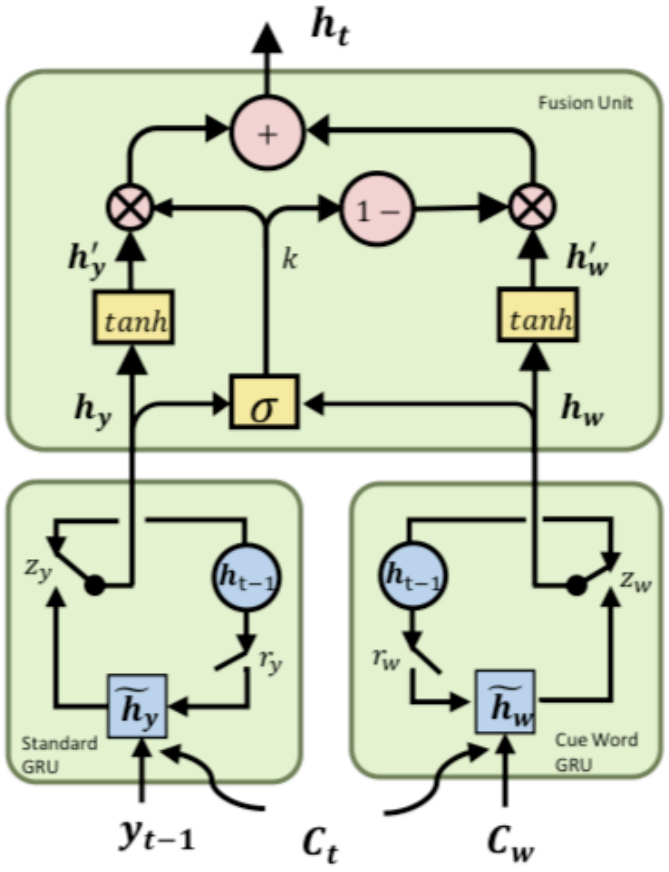
- It first generates a whole storyline which does not change during story writing.




(b) Static schema work-flow.

# Content-introducing Method

- *Towards Implicit Content-Introducing for Generative Short-text Conversation Systems*, Yao et al. EMNLP 2017
- incorporate additional information into the Seq2Seq model



# Dynamic: Storyline Planning

content-introducing generation problem

- context + the previous word in storyline

the first i-1 sentences in the story

context $ctx = [t, s_{1:i-1}]$

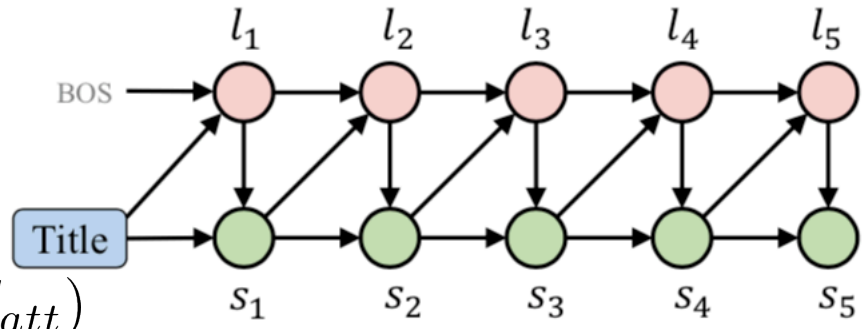
$\tilde{h}_{ctx} = Encode_{ctx}(ctx) = [\overrightarrow{h_{ctx}}; \overleftarrow{h_{ctx}}]$

$h_y = GRU(BOS, C_{att})$ $h_w = GRU(l_{i-1}, C_{att})$

$h'_y = tanh(W_1 h_y)$ $h'_w = tanh(W_2 h_y)$

$k = \sigma(W_k[h'_y; h'_w])$

$p(l_i|ctx, l_{i-1}) = g(k \circ h_y + (1-k) \circ h_w)$



(a) Dynamic schema work-flow.

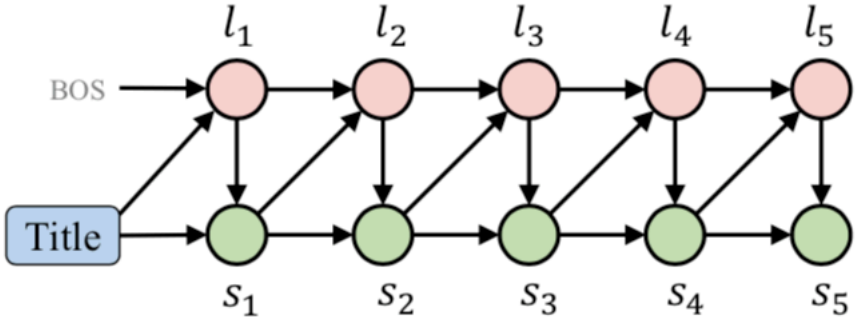
the beginning of decoding

the attention-based context

# Dynamic: Story Generation

- context + an additional storyline word

content-introducing generation problem



(a) Dynamic schema work-flow.

- minimize the negative log-probability

$$L(\theta)_{dyna} = \frac{1}{N}\sum_{j=1}^{N}[-log\prod_{i=1}^{m}p(s_i|ctx, l_i)]_j$$

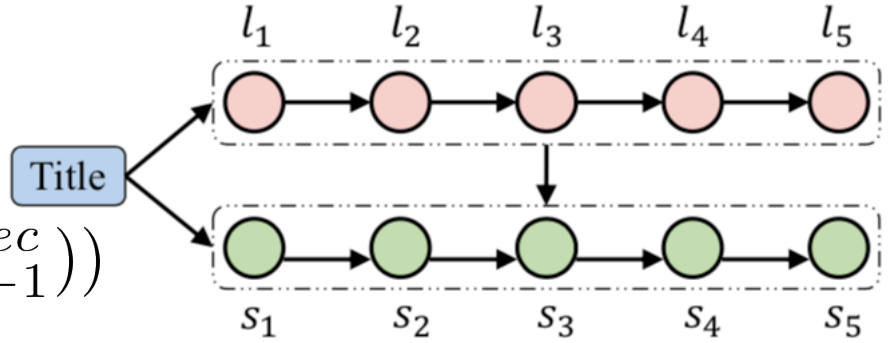
N: the number of stories in training data
m: the number of sentences in a story

# Static Schema

- It first generates a whole storyline which does not change during story writing.

**Storyline Planning:**
Seq2Seq conditional generation model

$$p(l_i|t, l_{1:i-1}; \theta) = g(LSTM_{att}(\tilde{h}, l_{i-1}, h_{i-1}^{dec}))$$




(b) Static schema work-flow.

**Story Generation:**
Seq2Seq conditional generation model

$$\tilde{h}_{tl} = Encode_{tl}([t, l]) = [\overrightarrow{h_{tl}}; \overleftarrow{h_{tl}}]$$

$$L(\theta)_{static} = \frac{1}{N} \sum_{j=1}^{N} [-log \prod_{i=1}^{m} p(s_i|\tilde{h}_{tl}, s_{1:i-1})]_j$$

# Dataset

ROCStories corpus (Mostafazadeh et al. 2016a)

- five-sentence stories
- capture a rich set of causal and temporal commonsense relations between daily events

| | |
|---|---|
| Number of Stories | 98, 161 |
| Vocabulary size | 33, 215 |
| Average number of words | 50 |

Table 2: Statistics of the ROCStories dataset.

8:1:1 for training, validation and testing

# Baseline

- Inc-S2S: Seq2Seq with attention

  generate the first sentence from a given title,

  generate the i-th sentece from the title

  and the previously generated i-1 sentences.

resemble dynamic schema without planning

- Cond-LM: Seq2Seq with attention

  generate the whole story word by word from a given title.

resemble static schema without planning

# Evaluation

- Objective metrics
  - inter-story repetition
  - intra-story repetition
- Subjective metrics(human):
  - fidelity
  - coherence
  - interestingness
  - overall user preference

# Objective Metrics

- quantify diversity across the generated stories.
- two meaurements: inter- and intra-story repetition.

each sentence position i

the repetition rate between stories at sentence i

the inter-story

$$r_e^i = 1 - \frac{T(\sum_{j=1}^{N} s^{ji})}{T_{all}(\sum_{j=1}^{N} s^{ji})}$$

the intra-story

$$r_a^i = \frac{1}{N}\sum_{j=1}^{N}\left[\frac{\sum_{k=1}^{i-1} T(s^i \bigcap s^k)}{(i-1) * T(s^i)}\right]^j$$

the average repetition of sentence $s^i$ comparing with former sentences in a story

T() and $T_{all}$(): the number of distinct and total trigram

$s^{ji}$: i-th sentence in j-th story

$s^i \cap s^k$: the distinct trigram intersection set between sentence $s^i$ and $s^k$

# Aggregate scores

$$r_e^{agg} = 1 - \frac{T(\sum_{j=1}^{N}\sum_{i=1}^{m} s^{ji})}{T_{all}(\sum_{j=1}^{N}\sum_{i=1}^{m} s^{ji})}$$

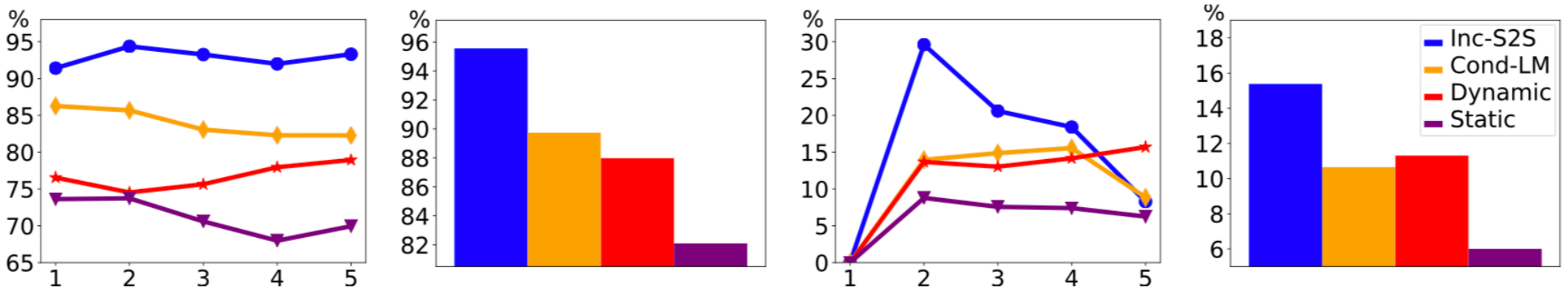
the overall repetition of all stories

$$r_a^{agg} = \frac{1}{m}\sum_{i=1}^{m} r_a^i$$

$$\sum_{j=1}^{N}\sum_{i=1}^{m} s^{ji}$$

the set of N stories with m sentences

# Results-Objective




(a) Inter-story repetition curve by sentence.

(b) Inter-story aggregate repetition scores.

(c) Intra-story repetition curve by sentence.

(d) Intra-story aggregate repetition scores.

Figure 3: Inter- and intra-story repetition rates by sentences (curves) and for the whole stories (bars), the lower the better. As reference points, the aggregate repetition rates on the human-written training data are 34% and 0.3% for the inter- and intra-story measurements respectively.

# Subjective Metrics

- fidelity
  whether the story is on-topic with the given title
- coherence
  whether the story is logically consistent and coherent
- interestingness
  whether the story is interesting
- overall user preference
  how do users like the story

# Results-Subjective

- Randomly sample 300 titles from the test data
- present a title and two generated sotries at a time
- ask them to decide which is better

| Choice % | Dynamic *vs* Inc-S2S | | | Static *vs* Cond-LM | | | Dynamic *vs* Static | | |
|---|---|---|---|---|---|---|---|---|---|
| | Dyna. | Inc. | Kappa | Static | Cond. | Kappa | Dyna. | Static | Kappa |
| Fidelity | **35.8** | 12.9 | 0.42 | **38.5** | 16.3 | 0.42 | 21.47 | **38.00** | 0.30 |
| Coherence | **37.2** | 28.6 | 0.30 | **39.4** | 32.3 | 0.35 | 28.27 | **49.47** | 0.36 |
| Interestingness | **43.5** | 26.7 | 0.31 | **39.5** | 35.7 | 0.42 | 34.40 | **42.60** | 0.35 |
| Overall Popularity | **42.9** | 27.0 | 0.34 | **40.9** | 34.2 | 0.38 | 30.07 | **50.07** | 0.38 |

# Dynamic vs. Static

- Storyline analysis

| Method | l-B1 | l-B2 | l-s |
|---|---|---|---|
| Dynamic | 6.46 | 0.79 | 0.88 |
| Static | 9.53 | 1.59 | 0.89 |

Table 6: The storyline BLEU score (only BLEU-1 and BLEU-2) and the correlation of storyline-story l-s.

With better storylines, it is easier to generate more relevant and coherent stories.

# Case study

| | | Title: Computer |
|---|---|---|
| **Baselines** | Inc-S2S | Tom's computer broke down. He needed to buy a new computer. He decided to buy a new computer. Tom bought a new computer. Tom was able to buy a new computer. |
| | Cond-LM | The man bought a new computer. He went to the store. He bought a new computer. He bought the computer. He installed the computer. |
| **Dynamic** | Storyline | needed → money → computer → bought → happy |
| | Story | John needed a computer for his birthday. He worked hard to earn money. John was able to buy his computer. He went to the store and bought a computer. John was happy with his new computer. |
| **Static** | Storyline | computer → slow → work → day → buy |
| | Story | I have an old computer. It was very slow. I tried to work on it but it wouldn't work. One day, I decided to buy a new one. I bought a new computer . |

Baseline without storylines tend to generate repetitive sentences that do not exhibit much of a story progress.

# Case study

**Title: The Virus**

| | | |
|---|---|---|
| **Baselines** | Inc-S2S | His computer was fixed and he fixed it. John got a new computer on his computer. John was able to fix it himself. John was able to fix his computer and was able to fix his computer. John was able to fix his computer and had a virus and was able to fix his computer. |
| | Cond-LM | Tim was working on a project. He was working on a project. Tim was working on a project. The project was really good. Tim was able to finish the project. |
| **Dynamic** | Storyline | computer → use → anywhere → house → found |
| | Story | I was working on my computer today. I was trying to use the computer. I couldn't find it anywhere. I looked all over the house for it. Finally, I found it. |
| **Static** | Storyline | work → fix → called → found → day |
| | Story | I had a virus on my computer. I tried to fix it but it wouldn't work. I called the repair company. They came and found the virus. The next day, my computer was fixed. |

The storyline generated by dynamic schema is not very coherent and this affects story quality.

# Error Analysis

- Select from stories generated by static schema

| Title / Problem | Story |
|---|---|
| **Taxi / off-topic** | I got a new car. It was one day. I decided to drive to the airport. I was driving for a long time. I had a great time . |
| **Cut / repetitive** | Anna was cutting her nails. She cut her finger and cut her finger. Then she cut her finger. It was bleeding! Anna had to bandage her finger. |
| **Eight glasses/ inconsistent** | Joe needed glasses. He went to the store to buy some. He did n't have any money. He found a pair that he liked. He bought them. |

The system can generate grammatical sentences that are coherent within a local context.

Generate a sequence of coherent and logically consistent sentences is still an challenge.

# Conclusion

- propose a plan-and-write framework.
- generate stories from given titles with explicit storyline planning.
- compare two strategies: dynamic and static schema.
- both outperform the baselines without storylines.
- static schema performs better than the dynamic.