Controllable Text Generation: Types, Knowledge, and Logic

Dr. Minlie Huang (黄民烈)
Associate Professor
CS Department, Tsinghua University
aihuang@tsinghua.edu.cn
http://coai.cs.tsinghua.edu.cn/hml
Research Picture of the CoAI Group

Conversational AI

Natural Language Understanding
Deep Learning
Reinforcement Learning

Task-Oriented Dialogue
Social Chatbot
Question Answering
Reading Comprehension
Sentiment Analysis
Recent Papers on RL

- Learning Structured Representation with RL (AAAI 2018)
- Data Denoising in Relation Extraction (AAAI 2018)
  - 入选PaperWeekly 2017年度最值得读的10篇NLP论文
- Weakly Supervised Label Correction (IJCAI-ECAI 2018)
- Hierarchical Relation Extraction (submitted to AAAI 2019)
- Learning to Collaborate: Joint Ranking Optimization (WWW 2018)
  - Multi-agent reinforcement learning; deterministic policy; actor-critic
- Search Result Aggregation (in preparation)
Recent Papers on Dialogue

- Perceiving and Expressing **Emotions** *(AAAI 2018)*
- Proactive Behavior by **Asking Good Questions** *(ACL 2018)*
- Controlling **Sentence Function** *(ACL 2018)*
- **Topic** Change in Multi-turn Dialog Systems *(SIGIR 2018)*
- **Explicit Personality** Assignment *(IJCAI-ECAI 2018)*
- Better Understanding and Generation Using **Commonsense Knowledge** *(IJCAI-ECAI 2018 Distinguished Paper)*
- **Discourse parsing** in multi-party dialogues *(AAAI 2019)*
- **Memory** augmented dialog management (submitted to TOIS)
- **Multimodality** neural belief tracker (submitted to WWW 2019)
Good Examples from LG Models

User: 最幸福的人不是拿到了最好的东西，而是学会去珍惜手上已经拥有的人。
   The happiest person is not the one who gets the best thing, instead who cherishes what she/he already has.
Machine: 是的，所以我也要珍惜自己的幸福
   Agree, I will cherish my happiness too.

User: 时间啊，你这把杀猪刀。。。;
   Time is a killer...
Machine: 我是被杀的
   I am the one who was killed.

User: 怕孤独，又舍不下自己的小自由。
   Afraid to be lonely, but wont’ give up my little freedom.
Machine: 我想去流浪，但没有人陪我去。
   I want to wander, but no one accompany me.
Bad Examples are More!

User: 其实，他没有那么爱你。
    Ohh, he does not love you that much.
Machine: 幸福的一天开始了。
    A happy day starts.

User: 回北京了！很冷很高兴！
    Back to Beijing now. So cold so happy!
Machine: 什么时候来北京啊？
    When will you come to Beijing?

User: 我来自上海。
    I come from Shanghai.
Machine: 真巧，我来自海南，老乡啊
    So cool, I am from Hainan, the same province!
Controllability of Text Generation

- Three **fundamental problems** in current neural language generation models
  - Semantics
  - **Consistency** (long text generation)
  - **Logic** (reasonable and making sense)
Conditional Text Generation

- Free Text to Text
  - Dialogue Generation
  - Story Generation
  - Abstractive Summarization
- Structured Data to Text
- Category to Text
  - Emotion/Sentiment Generation
- Keywords to Text
  - Poetry Generation
  - Essay/Narrative/Story Generation
- Image/Video to Text
  - Captioning
  - Visual story-telling

- Generation from Scratch: random variable
In this talk

- **Types**: Question Generation in Conversational Systems (ACL 2018)
- **Knowledge**: Commonsense-aware Dialogue Generation (IJCAI-ECAI 2018 Distinguished Paper)
- **Logic**: Storing Ending Generation (AAAI 2019)
Typed Decoder for Language Generation
Question Generation in Conversational Systems

我昨天晚上去聚餐了
I went to dinner yesterday night.

Yansen Wang, Chenyi Liu, Minlie Huang, Liqiang Nie. Learning to ask questions in open-domain conversation systems. ACL 2018.
Asking **good** questions requires **scene understanding**

Scene: Dining at a restaurant

我昨天晚上去了聚餐了

I went to dinner yesterday night.

- **Friends?**
- **Place?**
- **Food?**
- **Persons?**
- **Bill?**

WHO

WHERE

HOW-ABOUT

HOW-MANY

WHO

Yansen Wang, Chenyi Liu, Minlie Huang, Liqiang Nie.
Learning to ask questions in open-domain conversation systems. **ACL 2018.**
Question Generation in Conversational Systems

- Responding + **asking** (Li et al., 2016)
- **Key proactive** behaviors (Yu et al., 2016)
- Asking good questions are indication of **machine understanding**
- Key differences to **traditional** question generation (eg., reading comprehension):
  - **Different goals**: Information seeking vs. Enhancing interactivity and persistence of human-machine interactions
  - **Various patterns**: YES-NO, WH-, HOW-ABOUT, etc.
  - **Topic transition**: from topics in post to topics in response
Question Generation in Conversational Systems

- A good question is a natural composition of
  - **Interrogatives** for using various questioning patterns
  - **Topic words** for addressing interesting yet novel topics
  - **Ordinary words** for playing grammar or syntactic roles

Example 1:
User: I am too fat ...
Machine: *How about* climbing this weekend?

Example 2:
User: Last night, I stayed in KTV with friends.
Machine: *Are you happy with your* singing?
Encoder-decoder Framework

Encoder:

post: The cake tastes good <EOS>

Decoder:

response: Is it a cheese cake

\[ X = x_1 x_2 \cdots x_m \]

\[ Y = y_1 y_2 \cdots y_n \]

\[ Y^* = \arg\max_Y \mathcal{P}(Y | X). \]

\[ \mathcal{P}(y_t | y_{<t}, X) = \text{MLP}(s_t, e(y_{t-1}), c_t), \]

\[ s_t = \text{GRU}(s_{t-1}, e(y_{t-1}), c_t), \]

\[ c_t = \sum_{i=1}^{T} \alpha_{t,i} h_i \]

\[ h_t = \text{GRU}(h_{t-1}, e(x_t)), \]
Soft Typed Decoder (STD)

Encoder:
post: The cake tastes good <EOS>

Decoder:
response: Is it a cheese cake

Soft Typed Decoder (STD)

Decoder:
response: Is it a cheese cake

Decoding state
Soft Typed Decoder (STD)

• Applying **multiple type-specific generation distributions** over the same vocabulary

• Each word has a **latent** distribution among the set \( \text{type}(w) \in \{ \text{interrogative, topic word, ordinary word} \} \)

• STD is a very simple **mixture** model

\[
\mathcal{P}(y_t | y_{<t}, X) = \sum_{i=1}^{k} \mathcal{P}(y_t | ty_t = c_i, y_{<t}, X) \cdot \mathcal{P}(ty_t = c_i | y_{<t}, X),
\]

- **type-specific generation distribution**
- **word type distribution**
Soft Typed Decoder (STD)

- Estimate the **type distribution** of each word:
  \[ P(ty_t | y_{<t}, X) = \text{softmax}(W_0 s_t + b_0), \]

- Estimate the **type-specific generation distribution** of each word:
  \[ P(y_t | ty_t = c_i, y_{<t}, X) = \text{softmax}(W_{c_i} s_t + b_{c_i}), \]

- The final generation distribution is a **mixture** of the three type-specific generation distribution.

\[ P(y_t | y_{<t}, X) = \sum_{i=1}^{k} P(y_t | ty_t = c_i, y_{<t}, X) \cdot P(ty_t = c_i | y_{<t}, X), \]
Hard Typed Decoder (HTD)

• In soft typed decoder, word types are modeled in a latent, implicit way

• Can we control the word type more explicitly in generation?
  • Stronger control
Hard Typed Decoder (HTD)

Encoder:
post: The cake tastes good <EOS>

Decoder:
response: Is it a cheese cake

Hard Typed Decoder (HTD)
Gumbel-softmax

Decoder state
Hard Typed Decoder (HTD)

- Estimate the generation probability distribution
  \[ P(y_t | y_{<t}, X) = \text{softmax}(W_0s_t + b_0). \]
- Estimate the type probability distribution
  \[ P(ty_t | y_{<t}, X) = \text{softmax}(W_1s_t + b_1). \]
- Modulate words’ probability by its corresponding type probability:
  \[ P'(y_t | y_{<t}, X) = P(y_t | y_{<t}, X) \cdot m(y_t), \]

\( m(y_t) \) is related to the type probability of word \( y_t \).
### Hard Typed Decoder (HTD)

<table>
<thead>
<tr>
<th>Generation distr.</th>
<th>Type distr.</th>
<th>Modulated distr.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>what</strong> 0.3</td>
<td>$T_{interrogative}$ 0.7</td>
<td><strong>what</strong> 0.8</td>
</tr>
<tr>
<td><strong>food</strong> 0.2</td>
<td>$T_{topic}$ 0.1</td>
<td>$\rightarrow$ <strong>food</strong> 0.05</td>
</tr>
<tr>
<td><strong>is</strong> 0.4</td>
<td>$T_{ordinary}$ 0.2</td>
<td><strong>is</strong> 0.09</td>
</tr>
</tbody>
</table>

- **Argmax?** (firstly select largest type prob. then sample word from generation dist.)
- Indifferentiable
- Serious grammar errors if word type is wrongly selected
Hard Typed Decoder (HTD)

- **Gumble-Softmax:**
  - A differentiable surrogate to the argmax function.

\[
m(y_t) = \text{GS}(\mathcal{P}(ty_t = c(y_t)|y_{<t}, X)),
\]

\[
\text{GS}(\pi_i) = \frac{e^{(\log(\pi_i) + g_i)/\tau}}{\sum_{j=1}^{k} e^{(\log(\pi_j) + g_j)/\tau}},
\]
Hard Typed Decoder (HTD)

• In HTD, the types of words are given in advance.
  • *How to determine the word types?*
Hard Typed Decoder (HTD)

- **Interrogatives:**
  - A list of about 20 interrogatives are given by hand.

- **Topic words:**
  - Training: all nouns and verbs in response are topic words.
  - Test: 20 words are predicted by PMI.

- **Ordinary words:**
  - All other words, for grammar or syntactic roles

\[
PMI(w_x, w_y) = \log \frac{p(w_x, w_y)}{p_1(w_x) \cdot p_2(w_y)},
\]

\[
Rel(k_i, X) = \sum_{w_x \in X} e^{PMI(w_x, k_i)},
\]
Loss Function

- Cross entropy
- Supervisions are on both final probability and the type distribution:

\[
\Phi_1 = \sum_t - \log \mathcal{P}(y_t = \tilde{y}_t | y_{<t}, X),
\]
\[
\Phi_2 = \sum_t - \log \mathcal{P}(ty_t = \tilde{y}_t | y_{<t}, X),
\]
\[
\Phi = \Phi_1 + \lambda \Phi_2,
\]

- \( \lambda \) is a term to balance the two kinds of losses.
Dataset

• PMI estimation: calculated from 9 million post-response pairs from Weibo.

• Dialogue Question Generation Dataset (DQG), about 491,000 pairs:
  • Distilled questioning responses using about 20 hand-draft templates
  • Removed universal questions
  • Available at http://coai.cs.tsinghua.edu.cn/hml/dataset/
Baselines

• **Seq2Seq**: A simple encoder-decoder model (Luong et al., 2015)
• **Mechanism-Aware (MA)**: Multiple responding mechanisms represented by real-valued vectors (Zhou et al., 2017)
• **Topic-Aware (TA)**: Topic Aware Model by incorporating topic words (Xing et al., 2017)
• **Elastic Responding Machine (ERM)**: Enhanced MA using reinforcement learning (Zhou et al., 2018)
Automatic Evaluation

<table>
<thead>
<tr>
<th>Model</th>
<th>Perplexity</th>
<th>Distinct-1</th>
<th>Distinct-2</th>
<th>TRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seq2Seq</td>
<td>63.71</td>
<td>0.0573</td>
<td>0.0836</td>
<td>6.6%</td>
</tr>
<tr>
<td>MA</td>
<td>54.26</td>
<td>0.0576</td>
<td>0.0644</td>
<td>4.5%</td>
</tr>
<tr>
<td>TA</td>
<td>58.89</td>
<td>0.1292</td>
<td>0.1781</td>
<td>8.7%</td>
</tr>
<tr>
<td>ERM</td>
<td>67.62</td>
<td>0.0355</td>
<td>0.0710</td>
<td>4.5%</td>
</tr>
<tr>
<td>STD</td>
<td>56.77</td>
<td>0.1325</td>
<td>0.2509</td>
<td>12.1%</td>
</tr>
<tr>
<td>HTD</td>
<td>56.10</td>
<td>0.1875</td>
<td>0.3576</td>
<td>43.6%</td>
</tr>
</tbody>
</table>

Table 1: Results of automatic evaluation.

**Evaluation metrics**

- Perplexity & Distinct
- **TRR (Topical Response Ratio):**
  - 20 topic words are predicted with PMI for each post.
  - TRR is the proportion of the responses containing at least one topic word.
Manual Evaluation

• Pair-wise comparison: win, loss, tie
• Three evaluation criteria:
  • **Appropriateness**: whether a question is reasonable in logic and content, and has key info.
  • **Richness**: containing topic words or not
  • **Willingness** to respond to a generated question
## Manual Evaluation (Pairwise)

<table>
<thead>
<tr>
<th>Models</th>
<th>Appropriateness</th>
<th>Richness</th>
<th>Willingness</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Win (%)</td>
<td>Lose (%)</td>
<td>Tie (%)</td>
</tr>
<tr>
<td>STD vs. Seq2Seq</td>
<td>42.0</td>
<td>38.6</td>
<td>19.4</td>
</tr>
<tr>
<td>STD vs. MA</td>
<td>39.6*</td>
<td>31.2</td>
<td>29.2</td>
</tr>
<tr>
<td>STD vs. TA</td>
<td>42.2</td>
<td>40.0</td>
<td>17.8</td>
</tr>
<tr>
<td>STD vs. ERM</td>
<td>43.4*</td>
<td>34.4</td>
<td>22.2</td>
</tr>
<tr>
<td>HTD vs. Seq2Seq</td>
<td>50.6**</td>
<td>30.6</td>
<td>18.8</td>
</tr>
<tr>
<td>HTD vs. MA</td>
<td>54.8**</td>
<td>24.4</td>
<td>20.8</td>
</tr>
<tr>
<td>HTD vs. TA</td>
<td>52.0**</td>
<td>38.2</td>
<td>9.8</td>
</tr>
<tr>
<td>HTD vs. ERM</td>
<td>64.8**</td>
<td>23.2</td>
<td>12.0</td>
</tr>
<tr>
<td>HTD vs. STD</td>
<td>52.0**</td>
<td>33.0</td>
<td>15.0</td>
</tr>
</tbody>
</table>

Table 2: Annotation results. Win for “A vs. B” means A is better than B. Significance tests with Z-test were conducted. Values marked with * means $p$-value $< 0.05$, and ** for $p$-value $< 0.01$.

Score: the probability of win/lose/tie of our model vs. baseline
## Examples

| Post: | 我们昨天去唱歌了  
We went to karaoke yesterday. |
|-------|-------------------------------|
| Seq2Seq: | 什么时候？  
When? |
| MA: | 你怎么知道我是在问  
How did you know I am questioning you? |
| TA: | 什么？  
What? |
| ERM: | 什么情况？  
What happened? |
| STD: | 去哪里唱歌？  
Where did you sing karaoke? |
| HTD: | 你们几个人唱歌？  
How many people were singing with you? |

Table 4: Generated question examples. Interrogative words are bolded and topic words underlined.
More Examples

• Different questioning patterns and topic transition:

| Post-3:      | 我在体育中心上班。 | I work at the sports center. |
| Human:      | 你每天从家过去要多久？ | How long does it take you to go there from home? |
| STD:        | 你是什么部门的啊？ | Which department are you from? |
| HTD:        | 那你是体育学院的学生吗？ | Are you a student from the College of Physical Education? |

| Post-4:      | 今天的寿司真好吃。 | The sushi today is so delicious! |
| Human:      | 我记得你不喜欢吃日式料理的不是吗？ | I remember you didn’t like eating Japanese food, did you? |
| STD:        | 你什么时候来无锡，我请你。 | When will you come to Wuxi? It is my treat. |
| HTD:        | 是吗？改天尝尝！ | Really? I will try sometime! |
**Visualization**

- Type prediction at each decoding position

<table>
<thead>
<tr>
<th>Decoding steps</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Post:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Response:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interrogative</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>你 (you)</td>
<td>0.09</td>
<td>0.02</td>
<td>0.01</td>
<td>0.85</td>
<td>1.00</td>
<td>0.01</td>
</tr>
<tr>
<td>喜欢 (like)</td>
<td>0.26</td>
<td>0.35</td>
<td>0.71</td>
<td>0.14</td>
<td>0.00</td>
<td>0.02</td>
</tr>
<tr>
<td>兔子 (rabbit)</td>
<td>0.65</td>
<td>0.63</td>
<td>0.28</td>
<td>0.01</td>
<td>0.00</td>
<td>0.97</td>
</tr>
<tr>
<td>禾 (particle)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>畐 (？)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>EOS</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Knowledge in Language Generation
Commonsense Knowledge

- Commonsense knowledge consists of facts about the everyday world, that all humans are expected to know. (Wikipedia)
  - Lemons are sour
  - Tree has leaves
  - Dog has four legs

- Commonsense Reasoning ~ Winograd Schema Challenge:
  - The trophy would not fit in the brown suitcase because it was too big. What was too big?
  - The trophy would not fit in the brown suitcase because it was too small. What was too small?
Commonsense Knowledge

- lung disease
- respiratory disease
- air pollution
- chest tightness
- avoiding triggers

- IsA
- Prevented_by
- Caused_by
Commonsense Knowledge

- asthma
  - IsA: lung disease
  - IsA: respiratory disease
  - Caused_by: air pollution
  - Caused_by: chest tightness
  - Prevented_by: avoiding triggers
Post: I have an **asthma** since three years old.

**Triples in knowledge graph:**
(lung disease, IsA, **asthma**)
(**asthma**, Prevented_by, avoiding triggers)

---

From ConceptNet
Post: I have an asthma since three years old.

Response: I am sorry to hear that. Maybe avoiding triggers can prevent asthma attacks.
Post: I have an **asthma** since three years old.

**Triples in knowledge graph:**
- (lung disease, IsA, **asthma**)  
- (**asthma**, Prevented_by, avoiding triggers)

**Response:** I am sorry to hear that. Maybe **avoiding triggers** can prevent **asthma** attacks.
Input: I have asthma since three years old.

Output: It is good for you to avoid triggers.
Commonsense Knowledge in Chatbots

**Static graph attention**: encoding semantics in graph, Feeding knowledge-enhanced info. into the encoder
Commonsense Knowledge in Chatbots

Dynamic graph attention: first attend a graph, then to a triple within that graph, finally generate with the words in a graph.

\[ s_{t+1} = \text{GRU}(s_t, [c_t; c_t^g; c_t^k; e(y_t)]), \]
\[ e(y_t) = [w(y_t); k_j], \]
**Commonsense Knowledge in Chatbots**

**Dynamic graph attention:** first attend a graph, then to a triple within that graph, finally generate with the words in a graph.

\[ g_i = \sum_{n=1}^{N_{gi}} \alpha_n^s [h_n, t_n], \]
\[ \alpha_n^s = \frac{\exp(\beta_n^s)}{\sum_{j=1}^{N_{gi}} \exp(\beta_j^s)}, \]
\[ \beta_n^s = (W_r r_n)^T \tanh(W_h h_n + W_t t_n), \]
**Commonsense Knowledge in Chatbots**

**Dynamic graph attention**: first attend a graph, then to a triple within that graph, finally generate with the words in a graph.
Commonsense Knowledge in Chatbots

**Dynamic graph attention**: first attend a graph, then to a triple within that graph, finally generate with the words in a graph
Dynamic graph attention: first attend a graph, then to a triple within that graph, finally generate with the words in a graph
**Commonsense Knowledge in Chatbots**

**Dynamic graph attention**: first attend a graph, then to a triple within that graph, finally generate with the words in a graph.
**Commonsense Knowledge in Chatbots**

**Dynamic graph attention**: first attend a graph, then to a triple within that graph, finally generate with the words in a graph.
Commonsense Knowledge in Chatbots

**Dynamic graph attention**: first attend a graph, then to a triple within that graph, finally generate with the words in a graph.
Commonsense Knowledge in Chatbots

**Dynamic graph attention**: first attend a graph, then to a triple within that graph, finally generate with the words in a graph.
Commonsense Knowledge in Chatbots

**Dynamic graph attention:** first attend a graph, then to a triple within that graph, finally generate with the words in a graph
Commonsense Knowledge in Chatbots

- Dataset: filtered from 10M reddit single-round dialogs

<table>
<thead>
<tr>
<th>Conversational Pairs</th>
<th>Commonsense KB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>Entity</td>
</tr>
<tr>
<td>3,384,185</td>
<td>21,471</td>
</tr>
<tr>
<td>Validation</td>
<td>Relation</td>
</tr>
<tr>
<td>10,000</td>
<td>44</td>
</tr>
<tr>
<td>Test</td>
<td>Triple</td>
</tr>
<tr>
<td>20,000</td>
<td>120,850</td>
</tr>
</tbody>
</table>

Table 1: Statistics of the dataset and the knowledge base.
Hao Zhou, Tom Yang, Minlie Huang, Haizhou Zhao, Jingfang Xu, Xiaoyan Zhu.

# Commonsense Knowledge in Chatbots

## Automatic evaluation

<table>
<thead>
<tr>
<th>Model</th>
<th>Overall</th>
<th>High Freq.</th>
<th>Medium Freq.</th>
<th>Low Freq.</th>
<th>OOV</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ppx.</td>
<td>ent.</td>
<td>ppx.</td>
<td>ent.</td>
<td>ppx.</td>
</tr>
<tr>
<td>Seq2Seq</td>
<td>47.02</td>
<td>0.717</td>
<td>42.41</td>
<td>0.713</td>
<td>47.25</td>
</tr>
<tr>
<td>MemNet</td>
<td>46.85</td>
<td>0.761</td>
<td>41.93</td>
<td>0.764</td>
<td>47.32</td>
</tr>
<tr>
<td>CopyNet</td>
<td>40.27</td>
<td>0.96</td>
<td>36.26</td>
<td>0.91</td>
<td>40.99</td>
</tr>
<tr>
<td>CCM</td>
<td><strong>39.18</strong></td>
<td><strong>1.180</strong></td>
<td><strong>35.36</strong></td>
<td><strong>1.156</strong></td>
<td><strong>39.64</strong></td>
</tr>
</tbody>
</table>

## Manual evaluation

*(Sign-test, p-value<0.005)*

<table>
<thead>
<tr>
<th>Model</th>
<th>Overall</th>
<th>High Freq.</th>
<th>Medium Freq.</th>
<th>Low Freq.</th>
<th>OOV</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>app.</td>
<td>inf.</td>
<td>app.</td>
<td>inf.</td>
<td>app.</td>
</tr>
<tr>
<td>CCM vs. Seq2Seq</td>
<td>0.616</td>
<td>0.662</td>
<td>0.605</td>
<td>0.656</td>
<td>0.549</td>
</tr>
<tr>
<td>CCM vs. MemNet</td>
<td>0.602</td>
<td>0.647</td>
<td>0.593</td>
<td>0.656</td>
<td>0.566</td>
</tr>
<tr>
<td>CCM vs. CopyNet</td>
<td>0.600</td>
<td>0.640</td>
<td>0.606</td>
<td>0.669</td>
<td>0.586</td>
</tr>
</tbody>
</table>

**Sign-test, p-value<0.005**
Generation Examples 1

**Post:** He proposed **March 5th.** We will be **married** October 10th. So 7 months

**Knowledge:** (engaged, RelatedTo, married), (wedding, RelatedTo, married), (miss, DistinctFrom, married), (March, RelatedTo, months)

- **Seq2Seq:** I'm not sure how to answer that. I'm not sure how to make a date.
- **MemNet:** I'm assuming he's married to a gay marriage?
- **CopyNet:** I'm sorry, I'm not sure what you mean by that.
- **CCM:** I'm pretty sure he's going to be **engaged** in the next 6 months.
Generation Examples 2

Post: Totally thought it was going to be doug's grave.
Knowledge: (thought, RelatedTo, thinking), (doug, HasProperty, dead), (dead, RelatedTo, grave), (grave, RelatedTo, death)

- **Seq2Seq**: I was thinking the same thing.
- **MemNet**: I thought it was a dead end.
- **CopyNet**: I think it's a bit of a stretch.
- **CCM**: I thought it was doug's death. I was thinking the same thing.
Logic:
Story Ending Generation
Story Ending Generation

- Given a story context, conclude the story and complete the plot (reasonable in logic and causality)

**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

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.
Logic: Story Ending Generation

Finding context clues: plan the order of events and entities.

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**.

Jian Guan, Yansen Wang, Minlie Huang. **Story Ending Generation with Incremental Encoding and Commonsense Knowledge**. AAAI 2019
Logic: Story Ending Generation

Commonsense knowledge

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**.
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:

\[ Y^* = \arg\max_Y P(Y|X). \]
Logic: Story Ending Generation

Incremental Encoding

- $X_1$: Today is Halloween.
- $X_2$: Jack is so excited to go trick or treating tonight.
- $X_3$: He is going to dress up like a monster.
- $X_4$: The costume is real scary.

$Y$: He hopes to get a lot of candy.

Multi-Source Attention

Diagram showing the interactions and attention mechanisms involved in the story generation process.
Logic: Story Ending Generation

Attention to the knowledge base: static graph attention
Model--- Encoder

Possible solutions for encoding:

• 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 (our proposal)
Model --- Encoder

Incremental Encoding

- Effective to represent the context clues which may capture the key logic information.
- The current sentence $X_i$
- An attentive read of the preceding sentence $X_{i-1}: c_{lj}^{(i)}$

$$h_j^{(i)} = \text{LSTM}(h_{j-1}^{(i)}, e(x_j^{(i)}), c_{lj}^{(i)}), \ i \geq 2.$$  

- Story ending generation:

$$s_t = \text{LSTM}(s_{t-1}, e(y_{t-1}), c_{lt}),$$

$$\mathcal{P}(y_t | y_{<t}, X) = \text{softmax}(W_0 s_t + b_0),$$
Model --- Encoder

Context vector

• Capture the relationship between words (or states) in the current sentence and those in the preceding sentence

• Encode implicit knowledge that is beyond the text

• Formally:  \[ c_{l_j}^{(i)} = W_1([c_{h_j}^{(i)}; c_{x_j}^{(i)}]) + b_1, \]

  • \( c_{h_j}^{(i)} \) is called state context vector pointing to \( X_{i-1} \)
  
  • \( c_{x_j}^{(i)} \) is called knowledge context vector pointing to \( X_{i-1} \)
Model --- Encoder

- State context vector

\[ c_{h,j}^{(i)} = \sum_{k=1}^{l_{i-1}} \alpha_{h_k,j}^{(i)} h_k^{(i-1)}, \]

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

\[ \beta_{h_k,j}^{(i)} = h_{j-1}^{(i)T} W_s h_k^{(i-1)}, \]

- Knowledge context vector

\[ c_{x,j}^{(i)} = \sum_{k=1}^{l_{i-1}} \alpha_{x_k,j}^{(i)} g(x_k^{(i-1)}), \]

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

\[ \beta_{x_k,j}^{(i)} = h_{j-1}^{(i)T} W_k g(x_k^{(i-1)}), \]
Model --- Knowledge

Knowledge graph retrieval

• **ConceptNet**: a commonsense semantic network
• Consists of triples $R = (h, r, t)$ meaning that head concept $h$ has the relation $r$ with tail concept $t$
  • 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

- 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 Attention**

  \[
  g(x) = \sum_{i=1}^{N_x} \alpha_{R_i} [h_i; t_i],
  \]

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

  \[
  \beta_{R_i} = (W_r r_i)^T \tanh(W_h h_i + W_t t_i),
  \]

- **Contextual Attention**

  \[
  g(x) = \sum_{i=1}^{N_x} \alpha_{R_i} M_{R_i},
  \]

  \[
  M_{R_i} = BiGRU(h_i, r_i, t_i),
  \]

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

  \[
  \beta_{R_i} = h_{(x)}^T W_c M_{R_i},
  \]
Model --- Knowledge

• Impose supervision on both the encoding network and decoding network

\[ \Phi = \Phi_{en} + \Phi_{de} \]

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

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

• 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

• ConceptNet
  • 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
Metrics

• **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 2 : totally incompatible endings
## Results

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

Table 1: Automatic and manual evaluation results.
## Examples

| Context: | Martha is **cooking** a special **meal** for her family. She **wants everything to be just right** for when they eat. Martha **perfection** everything and puts her **dinner** into the **oven**. Martha goes to **lay down** for a quick **nap**. She **oversleep** and runs into the **kitchen** to take out her **burnt dinner**. |
| Golden Ending: | |
| **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**. |
| **GA:** | She is very happy that she has made a new **cook**. |
| **HLSTM+CA:** | |
| **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**. |
Generation Examples

**Story 1:**
**Context:**
Taj has *never drank* an *espresso drink*. He *ordered* one while out with his friends. The shot of *espresso* *tasted terrible* to him. Taj found that he *couldn't stop talking or moving*.

**Generated Ending:**
He decided to *never drink* again.

**Story 2:**
**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.

**Generated Ending:**
When she *gets back to the kitchen*, she sees a *burning light* on the *stove*. 
“Logic Chains”: Contextual Clue

Building context clues incrementally

$X_1$: Martha is cooking a special meal for her family.

$X_2$: She wants everything to be just right for when they eat.

$X_3$: Martha perfects everything and puts her dinner into the oven.

$X_4$: Martha goes to lay down for a quick nap.

$Y$: When she gets back to the kitchen, she sees a burning light on the stove.
Controllable Language Generation

- Three **fundamental problems** in current neural language generation models
  - **Semantics**
  - **Consistency** (long text generation)
  - **Logic** (reasonable and making sense)

- Long text generation: **planning**
Thanks for Your Attention

- http://coai.cs.tsinghua.edu.cn/ds/ 对话系统技术平台

- Acknowledgements
  - Prof Xiaoyan Zhu, Tsinghua colleagues, collaborators
  - Our students

- Contact:
  - Minlie Huang, Tsinghua University
  - aihuang@tsinghua.edu.cn
  - http://coai.cs.tsinghua.edu.cn/hml