Scalable Neural Language Generation for Open Domain Dialogue Systems

Speaker: Tsung-Hsien Wen
Supervisor: Professor Steve Young
Spoken Dialogue System

- Speech Recognition
- Language Understanding
- Speech Synthesis
- Language Generation
- Dialogue Manager

Connections:
- Knowledge Base
- Web

Diagram elements:
- Dialogue System
- Spoken Dialogue System
NLG: Problem Definition

- Given a meaning representation, map it into a natural language
  - inform(type=Seven_days, food=Chinese)
  - Seven_days serves good Chinese food.

- What we care about?
  - adequacy, fluency, readability, variation (Stent et al 2005)
Traditional pipeline approach

DM
- Content Selection
- Dialogue state
- Dialogue Act

NLG in SDS
- Sentence Planning
- Tree-like template
- Surface Realisation
- Utterance
Motivation

- Traditionally, NLG is not scalable because:
  - Embrace a rule-based regime
  - Highly specialised for in-domain applications

- Talking to NLG is not enjoyable because of:
  - Frequent repetition of certain output forms
  - Awkward responses that are not colloquial
Why RNN for NLG?

- Elegant structure for modeling **sequences**.
- Flexible architecture for adding **auxiliary information**.
- Collecting data is convenient and quick (**crowdsourcing**).
- More human-like and **colloquial**.
- **No expert** knowledge is required.
- **Extensible**, adaptation techniques exist.
- **Distributed representation**
- **Less cost, quicker** development cycle
- **End-to-End** trainable
Challenges

• How to render the exact information we want (with the existence of language variation)?

• Adopted methods:
  • Overgeneration – Reranking paradigm (Oh and Rudnicky 2000)
    • Sample words from a Recurrent Generation Model output.
    • Select top candidates based on some scoring criteria.
Part 1
Heuristically Gated RNN Generator
Outline

- Recurrent Generation Model
- Convolutional Semantic Reranker
- Backward RNN Reranker
- Experiments
  - Setup
  - Automatic Evaluation
  - Human Evaluation
Outline

- **Recurrent Generation Model**
- Convolutional Semantic Reranker
- Backward RNN Reranker
- Experiments
  - Setup
  - Automatic Evaluation
  - Human Evaluation
Recurrent Generation Model (1/2)

Inform(name=Seven_Days, food=Chinese)

\[
\begin{bmatrix}
0, 0, 1, 0, 0, \ldots, 1, 0, 0, \ldots, 1, 0, 0, 0, 0, 0, \ldots
\end{bmatrix}
\]

\textit{dialog act 1-hot representation}

delexicalisation

(Mikolov et al. 2010)
• Heuristically check (exact match) whether a given slot token has been generated.

• Apply a decay factor $\delta < 1$ on generated feature values.

• Use features to configure the network NOT to re-generate slots that have already generated.

• Binary slots and don’t care values cannot be handled.

<table>
<thead>
<tr>
<th>Feature value</th>
<th>$&lt;$s$&gt;$</th>
<th>SLOT_NAME</th>
<th>serves</th>
<th>SLOT_FOOD</th>
<th>.</th>
<th>$&lt;$s$&gt;$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>1</td>
<td>1</td>
<td>$\delta$</td>
<td>$\delta^2$</td>
<td>$\delta^3$</td>
<td>$\delta^4$</td>
</tr>
<tr>
<td>FOOD</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$\delta$</td>
<td>$\delta^2$</td>
</tr>
</tbody>
</table>
Outline

- Recurrent Generation Model
- **Convolutional Semantic Reranker**
- Backward RNN Reranker
- Experiments
  - Setup
  - Automatic Evaluation
  - Human Evaluation
Convolutional Semantic Reranker (1/2)

- Designed to handle:
  - Binary slots: ALLOW_KID=yes/no
  - “don’t care” values: AREA=dont_care
  - Use CNN for semantic validation
Convolutional Semantic Reranker (2/2)

Target dialogue act: `inform(name=Seven_days, food=Chinese)`
Generated candidate: `</s> SLOT_NAME serves SLOT_FOOD . </s>`

(Kalchbrenner et al., 2014)
Outline

• Recurrent Generation Model
• Convolutional Semantic Reranker
• **Backward RNN Reranker**
• Experiments
  • Setup
  • Automatic Evaluation
  • Human Evaluation
Backward RNN Reranker

- Motivation:
  - Considering backward context can reduce grammatical errors.
  - Ex. “Seven Days is an exceptional restaurant.”
  - Integrating information from both directions is tricky.
  - The generation procedure is sequential in one direction only.
  - Alternative => train an RNN in reverse direction and use it for rescoring.
Outline

- Recurrent Generation Model
- Convolutional Semantic Reranker
- Backward RNN Reranker
- **Experiments**
  - **Setup**
  - Automatic Evaluation
  - Human Evaluation
Setup

- Data collection:
  - SFX Restaurant domain: 8 system act types, 12 slots (1 is binary).
  - Workers recruited from Amazon MT
  - Asked to generate system responses given a dialogue act.
  - Result in ~5.1K utterances, 228 distinct acts
- Training: BPTT, L2 regularisation, SGD w/ early stopping.
  
  train/valid/test: 3/1/1, data up-sampling
Outline

- Recurrent Generation Model
- Convolutional Semantic Reranker
- Backward RNN Reranker
- Experiments
  - Setup
  - **Automatic Evaluation**
  - Human Evaluation
Automatic Evaluation (1/2)

- Test set: 1039 utterances, 1848 required slots.
- Metrics: BLEU-4 (against multiple references), ERR(slot errors)
- Results averaged over 10 random initialised networks
- Compared with class-based LM (classlm), handcrafted generator (hdc), and kNN based model.
## Automatic Evaluation (2/2)

<table>
<thead>
<tr>
<th>Selection Beam</th>
<th>BLEU</th>
<th>hdc</th>
<th>knn</th>
<th>classlm</th>
<th>rnn</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/20</td>
<td></td>
<td>0.440</td>
<td>0.591</td>
<td>0.757</td>
<td>0.777</td>
</tr>
<tr>
<td>5/20</td>
<td></td>
<td>-</td>
<td>-</td>
<td>0.678</td>
<td>0.712</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Selection Beam</th>
<th>ERR</th>
<th>hdc</th>
<th>knn</th>
<th>classlm</th>
<th>rnn</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/20</td>
<td></td>
<td>0</td>
<td>17.2</td>
<td>47.8</td>
<td>0</td>
</tr>
<tr>
<td>5/20</td>
<td></td>
<td>-</td>
<td>-</td>
<td>104.6</td>
<td>3.1</td>
</tr>
</tbody>
</table>
Outline

• Recurrent Generation Model
• Convolutional Semantic Reranker
• Backward RNN Reranker
• Experiments
  • Setup
  • Automatic Evaluation
  • Human Evaluation
Human Evaluation (1/3)

• Setup
  • Judges (~60) recruited from Amazon MT.
  • Asked to evaluate two system responses pairwise.
  • Comparing handcrafted (hdc), RNN top-1 (rnn\(^1\)), RNN sample from top-5 (rnn\(^5\)), and class-based LM sampled from top-5 (classlm\(^5\))

• Metrics:
  • Informativeness, Naturalness (rating out of 5)
  • Preference
## Human Evaluation (2/3)

<table>
<thead>
<tr>
<th>Metrics</th>
<th>hdc</th>
<th>rnn1</th>
<th>hdc</th>
<th>rnn5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info.</td>
<td>3.75</td>
<td>3.81</td>
<td>3.85</td>
<td>3.93*</td>
</tr>
<tr>
<td>Nat.</td>
<td>3.58</td>
<td>3.74**</td>
<td>3.57</td>
<td>3.94**</td>
</tr>
<tr>
<td>Pref.</td>
<td>44.8%</td>
<td>55.2%*</td>
<td>37.2%</td>
<td>62.8%**</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metrics</th>
<th>rnn1</th>
<th>rnn5</th>
<th>classlm5</th>
<th>rnn5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info.</td>
<td>3.75</td>
<td>3.72</td>
<td>4.02</td>
<td>4.15%*</td>
</tr>
<tr>
<td>Nat.</td>
<td>3.67</td>
<td>3.58</td>
<td>3.91</td>
<td>4.02</td>
</tr>
<tr>
<td>Pref.</td>
<td>47.5%</td>
<td>52.5%</td>
<td>47.1%</td>
<td>52.9%</td>
</tr>
</tbody>
</table>

*=p<.05, **<.005
## Human Evaluation (2/3)

<table>
<thead>
<tr>
<th>Metrics</th>
<th>hdc</th>
<th>rnn1</th>
<th>hdc</th>
<th>rnn5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info.</td>
<td>3.75</td>
<td>3.81</td>
<td>3.85</td>
<td>3.93*</td>
</tr>
<tr>
<td>Nat.</td>
<td>3.58</td>
<td>3.74**</td>
<td>3.57</td>
<td>3.94**</td>
</tr>
<tr>
<td>Pref.</td>
<td>44.8%</td>
<td>55.2%*</td>
<td>37.2%</td>
<td>62.8%**</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metrics</th>
<th>rnn1</th>
<th>rnn5</th>
<th>classlm5</th>
<th>rnn5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info.</td>
<td>3.75</td>
<td>3.72</td>
<td>4.02</td>
<td>4.15*</td>
</tr>
<tr>
<td>Nat.</td>
<td>3.67</td>
<td>3.58</td>
<td>3.91</td>
<td>4.02</td>
</tr>
<tr>
<td>Pref.</td>
<td>47.5%</td>
<td>52.5%</td>
<td>47.1%</td>
<td>52.9%</td>
</tr>
</tbody>
</table>

*=p<.05, **<.005
<table>
<thead>
<tr>
<th>Metrics</th>
<th>hdc</th>
<th>rnn1</th>
<th>hdc</th>
<th>rnn5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info.</td>
<td>3.75</td>
<td>3.81</td>
<td>3.85</td>
<td>3.93*</td>
</tr>
<tr>
<td>Nat.</td>
<td>3.58</td>
<td>3.74**</td>
<td>3.57</td>
<td>3.94**</td>
</tr>
<tr>
<td>Pref.</td>
<td>44.8%</td>
<td>55.2%*</td>
<td>37.2%</td>
<td>62.8%**</td>
</tr>
<tr>
<td>Metrics</td>
<td>rnn1</td>
<td>rnn5</td>
<td>classlm5</td>
<td>rnn5</td>
</tr>
<tr>
<td>Info.</td>
<td>3.75</td>
<td>3.72</td>
<td>4.02</td>
<td>4.15%*</td>
</tr>
<tr>
<td>Nat.</td>
<td>3.67</td>
<td>3.58</td>
<td>3.91</td>
<td>4.02</td>
</tr>
<tr>
<td>Pref.</td>
<td>47.5%</td>
<td>52.5%</td>
<td>47.1%</td>
<td>52.9%</td>
</tr>
</tbody>
</table>

*=p<.05, **<.005

Human Evaluation (2/3)
### Human Evaluation (2/3)

<table>
<thead>
<tr>
<th>Metrics</th>
<th>hdc</th>
<th>rnn1</th>
<th>hdc</th>
<th>rnn5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info.</td>
<td>3.75</td>
<td>3.81</td>
<td>3.85</td>
<td>3.93*</td>
</tr>
<tr>
<td>Nat.</td>
<td>3.58</td>
<td>3.74**</td>
<td>3.57</td>
<td>3.94**</td>
</tr>
<tr>
<td>Pref.</td>
<td>44.8%</td>
<td>55.2%*</td>
<td>37.2%</td>
<td>62.8%**</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metrics</th>
<th>rnn1</th>
<th>rnn5</th>
<th>classlm5</th>
<th>rnn5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info.</td>
<td>3.75</td>
<td>3.72</td>
<td>4.02</td>
<td>4.15*</td>
</tr>
<tr>
<td>Nat.</td>
<td>3.67</td>
<td>3.58</td>
<td>3.91</td>
<td>4.02</td>
</tr>
<tr>
<td>Pref.</td>
<td>47.5%</td>
<td>52.5%</td>
<td>47.1%</td>
<td>52.9%</td>
</tr>
</tbody>
</table>

*=p<.05, **<.005
Human Evaluation (2/3)

<table>
<thead>
<tr>
<th>Metrics</th>
<th>hdc</th>
<th>rnn1</th>
<th>hdc</th>
<th>rnn5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info.</td>
<td>3.75</td>
<td>3.81</td>
<td>3.85</td>
<td>3.93*</td>
</tr>
<tr>
<td>Nat.</td>
<td>3.58</td>
<td>3.74**</td>
<td>3.57</td>
<td>3.94**</td>
</tr>
<tr>
<td>Pref.</td>
<td>44.8%</td>
<td>55.2%*</td>
<td>37.2%</td>
<td>62.8%**</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metrics</th>
<th>rnn1</th>
<th>rnn5</th>
<th>classlm5</th>
<th>rnn5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info.</td>
<td>3.75</td>
<td>3.72</td>
<td>4.02</td>
<td>4.15%*</td>
</tr>
<tr>
<td>Nat.</td>
<td>3.67</td>
<td>3.58</td>
<td>3.91</td>
<td>4.02</td>
</tr>
<tr>
<td>Pref.</td>
<td>47.5%</td>
<td>52.5%</td>
<td>47.1%</td>
<td>52.9%</td>
</tr>
</tbody>
</table>

* = p < .05, ** = p < .005
### Human Evaluation (3/3)

<table>
<thead>
<tr>
<th>#</th>
<th>Dialogue act and example realisations of our system, by sampling from top-5 candidates</th>
</tr>
</thead>
</table>
| 1  | inform(name=”stroganoff restaurant”, pricerange=cheap, near=”fishermans wharf”)  
    stroganoff restaurant is a cheap restaurant near fishermans wharf.  
    stroganoff restaurant is in the cheap price range near fishermans wharf. |
| 2  | reject(kidsallowed=yes, food=”basque”)  
    unfortunately there are 0 restaurants that allow kids and serve basque. |
| 3  | informonly(name=”bund shanghai restaurant”, food=”shanghainese”)  
    i apologize, no other restaurant except bund shanghai restaurant that serves shanghainese.  
    sorry but there is no place other than the restaurant bund shanghai restaurant for shanghainese. |
| 4  | confirm(goodformeal=dontcare)  
    i am sorry. just to confirm. you are looking for a restaurant good for any meal?  
    can i confirm that you do not care about what meal they offer? |
| 5  | request(near)  
    would you like to dine near a particular location? |
| 6  | reqmore()  
    is there anything else i can do for you? |
| 7  | select(kidsallowed=yes, kidsallowed=no)  
    are you looking for a restaurant that allows kids, or does not allow kids? |
| 8  | goodbye()  
    thank you for calling. good bye. |
A Brief Summary

- RGM learns generation decisions from corpus.
- No rules, grammars, semantic alignments, or heavy feature engineering are required.

Can we do better?
- No heuristic rules for gates.
- Direct control of generating arbitrary slot-value pairs.
- Better performance.
Part 2
Semantically Controlled LSTM Generator
Outline

• **SC-LSTM**
• Deep Model
• Experiments
  • Automatic Evaluation
  • Human Evaluation
SC-LSTM (1/4)

- **Original LSTM cell**

  \[
  i_t = \sigma(W_{wi}w_t + W_{hi}h_{t-1}) \quad (1)
  \]

  \[
  f_t = \sigma(W_{wf}w_t + W_{hf}h_{t-1}) \quad (2)
  \]

  \[
  o_t = \sigma(W_{wo}w_t + W_{ho}h_{t-1}) \quad (3)
  \]

  \[
  \hat{c}_t = \tanh(W_{wc}w_t + W_{hc}h_{t-1}) \quad (4)
  \]

  \[
  c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t \quad (5)
  \]

  \[
  h_t = o_t \odot \tanh(c_t) \quad (6)
  \]

(Hochreiter and Schmidhuber, 1997)
SC-LSTM (2/4)

- **Original LSTM cell**
  \[
  i_t = \sigma(W_{wi}w_t + W_{hi}h_{t-1}) \\
  f_t = \sigma(W_{wf}w_t + W_{hf}h_{t-1}) \\
  o_t = \sigma(W_{wo}w_t + W_{ho}h_{t-1}) \\
  \hat{c}_t = \tanh(W_{wc}w_t + W_{hc}h_{t-1}) \\
  c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t \\
  h_t = o_t \odot \tanh(c_t)
  \]

- **DA cell**
  \[
  r_t = \sigma(W_{wr}w_t + \alpha W_{hr}h_{t-1}) \\
  d_t = r_t \odot d_{t-1}
  \]

(Hochreiter and Schmidhuber, 1997)
SC-LSTM (3/4)

- **Original LSTM cell**
  \[
  i_t = \sigma(W_{wi} w_t + W_{hi} h_{t-1}) \quad (1) \\
  f_t = \sigma(W_{wf} w_t + W_{hf} h_{t-1}) \quad (2) \\
  o_t = \sigma(W_{wo} w_t + W_{ho} h_{t-1}) \quad (3) \\
  \hat{c}_t = \tanh(W_{wc} w_t + W_{hc} h_{t-1}) \quad (4) \\
  c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t \quad (5) \\
  h_t = o_t \odot \tanh(c_t) \quad (6)
  \]

- **DA cell**
  \[
  r_t = \sigma(W_{wr} w_t + \alpha W_{hr} h_{t-1}) \quad (7) \\
  d_t = r_t \odot d_{t-1} \quad (8)
  \]

- **Modify eq. (6) to**
  \[
  c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t + \tanh(W_{dc} d_t) \quad (9)
  \]

(Hochreiter and Schmidhuber, 1997)
SC-LSTM (4/4)

- **Cost function**
  \[
  F(\theta) = \sum_t p_t^T \log(y_t) \\
  + \|d_T\| \\
  + \sum_{t=0}^{T-1} \eta \xi \|d_{t+1} - d_t\|
  \]

- **1st term**: cross entropy error
- **2nd term**: make sure rendering all the information needed
- **3rd term**: prevent undesirable gating behaviors

(Hochreiter and Schmidhuber, 1997)
Outline

• SC-LSTM

• **Deep Model**

• Experiments
  • Automatic Evaluation
  • Human Evaluation
Deep Model (1/2)
Deep Model (2/2)

• Techniques applied
  • Skip connection (Graves et al 2013)
  • RNN dropout (Srivastava et al 2014)

• Gating Equation is modified

\[ r_t = \sigma(W_{wr} w_t + \alpha W_{hr} h_{t-1}) \]  \hspace{1cm} (7)

• To

\[ r_t = \sigma(W_{wr} w_t + \sum_l \alpha_l W_{hr}^l h_{t-1}^l) \]  \hspace{1cm} (12)
Outline

• SC-LSTM
• Deep Model
• Experiments
  • Automatic Evaluation
  • Human Evaluation
Automatic Evaluation (1/3)

- Dataset: SFX Restaurant & SFX Hotel Domains
- 5K utterances, 3:1:1 splitting
- 248/164 distinct acts, 2.25/1.95 # of slot per DA

- Ontologies:

<table>
<thead>
<tr>
<th>act type</th>
<th>SF Restaurant</th>
<th>SF Hotel</th>
</tr>
</thead>
<tbody>
<tr>
<td>inform, inform_only, reject, confirm, select, request, reqmore, goodbye</td>
<td>name, type, *pricerange, price, phone, address, postcode, *area, *near</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>specific</th>
<th>SF Restaurant</th>
<th>SF Hotel</th>
</tr>
</thead>
<tbody>
<tr>
<td>*food</td>
<td>*hasinternet</td>
<td>*acceptscards</td>
</tr>
<tr>
<td>*goodformeal</td>
<td>*acceptscards</td>
<td>*dogs-allowed</td>
</tr>
<tr>
<td>*kids-allowed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*bold*=binary slots, *_=slots can take “don’t care” value
Automatic Evaluation (2/3)

Selection scheme : 5/20

BLEU

Model

hdc
knn
classlm
rnn
sc-lstm
+deep

Restaurant
Hotel

Selection scheme : 5/20
Automatic Evaluation (3/3)

Selection scheme: 5/20

- hdc
- knn
- classlm
- rnn
- sc-lstm
- +deep

ERR (%)

Model

Restaurant  Hotel

Selection scheme: 5/20
Outline

• SC-LSTM
• Deep Model
• Experiments
  • Automatic Evaluation
  • Human Evaluation
Human Evaluation (1/3)

- Setting
  - Done on SFX Restaurant domain
  - Comparing \textit{classlm, rnn w/ sc-lstm} and \textit{+deep}

- Metrics
  - Informativeness, Naturalness, Preference
# Human Evaluation (2/3)

<table>
<thead>
<tr>
<th>Method</th>
<th>Informativeness</th>
<th>Naturalness</th>
</tr>
</thead>
<tbody>
<tr>
<td>+deep</td>
<td>2.58</td>
<td>2.51</td>
</tr>
<tr>
<td>sc-lstm</td>
<td><strong>2.59</strong></td>
<td>2.50</td>
</tr>
<tr>
<td>rnn w/</td>
<td>2.53</td>
<td>2.42*</td>
</tr>
<tr>
<td>classlm</td>
<td>2.46**</td>
<td>2.45</td>
</tr>
</tbody>
</table>

* *p < 0.05 ** *p < 0.005
## Human Evaluation (3/3)

<table>
<thead>
<tr>
<th>Pref. %</th>
<th>classlm</th>
<th>rnn w/</th>
<th>sc-lstm</th>
<th>+deep</th>
</tr>
</thead>
<tbody>
<tr>
<td>classlm</td>
<td>-</td>
<td>46.0</td>
<td>40.9**</td>
<td>37.7**</td>
</tr>
<tr>
<td>rnn w/</td>
<td>54.0</td>
<td>-</td>
<td>43.0</td>
<td>35.7*</td>
</tr>
<tr>
<td>sc-lstm</td>
<td>59.1*</td>
<td>57</td>
<td>-</td>
<td>47.6</td>
</tr>
<tr>
<td>+deep</td>
<td>62.3**</td>
<td>64.3**</td>
<td>52.4</td>
<td>-</td>
</tr>
</tbody>
</table>

* $p < 0.05$  ** $p < 0.005$
Example
Conclusion
Conclusion – Why RNN for NLG?

✓ Elegant structure for modeling sequences.

• Flexible architecture for adding auxiliary information.

✓ Collecting data is convenient and quick (crowdsourcing).

✓ More human-like and colloquial.

✓ No expert knowledge is required.

• Extensible, adaptation techniques exist.

• Distributed representation

✓ Less cost, quicker development cycle

✓ End-to-End trainable
Papers


Thank you! Questions?

This project is supported by Toshiba Research Europe Ltd, Cambridge Research Laboratory.