Scalable Neural Language Generation for Spoken Dialogue Systems

XRCE Seminar, 23/02/2016

Tsung-Hsien (Shawn) Wen and Steve Young
Outline

- Intro
- Semantically Conditioned LSTM
- Domain adaptation for NLG
- Conclusion
Outline

- **Intro**
- Semantically Conditioned LSTM
- Domain adaptation for NLG
- Conclusion
Given a meaning representation, map it into natural language utterances.

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

Inform(
  name=Z_House,
  price=cheap
)

Z House is a cheap restaurant.

Dialogue Act

Tree-like template

Utterance
Problems

- Scalability
  - Grammars are handcrafted.
  - Require expert knowledge.
Problems

- Boring
  - Frequent repetition of outputs.
  - Non-colloquial, awkward utterances.

Seven Days is a nice restaurant in the expensive price range, in the north part of the town, if you don’t care about what food they serve.
Outline

- Intro
- **Semantically Conditioned LSTM**
- Experiments
- Adaptation – A preliminary work
- Conclusion
Recurrent Generation Model

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}

\[</s>\]

\[SLOT\_NAME \quad \text{serves} \quad SLOT\_FOOD\]

\textit{delexicalisation}

RNNLM (Mikolov et al, 2010)
SC-LSTM

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

- **Modify \( C_t \)**
  
  \[
  c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t + \tanh(W_{dc} d_t)
  \]

(Hochreiter and Schmidhuber, 1997)
Visualization

- A-inform
- pricerange=don't care
- kids_allowed=yes
- count=VALUE
- food=VALUE
- type=VALUE

- A-inform
- accepts_cards=yes
- hasインターネット=yes
- near=VALUE
- name=VALUE

(feature value)
SC-LSTM

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

- 1\textsuperscript{st} term: Log-likelihood
- 2\textsuperscript{nd} term: make sure rendering all the information needed
- 3\textsuperscript{rd} term: close only one gate each time step.

(Hochreiter and Schmidhuber, 1997)
Intuition behind the 3\(^{rd}\) term

\[ \eta = 0.01, \xi = 100 \]
Deep Architecture
Deep Architecture

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

- Intro
- Semantically Conditioned LSTM
  - Experiments
- Domain adaptation for NLG
- Conclusion
Setup

- Data collection:
  - SFX restaurant/hotel domains
## 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></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>shared</th>
<th>name, type, *pricerange, price, phone, address, postcode, *area, *near</th>
</tr>
</thead>
</table>

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

**bold**=binary slots, ***=slots can take “don’t care” value
Setup

- Data collection:
  - SFX restaurant/hotel domains
  - Workers recruited from Amazon MT.
  - Asked to generate system responses given a DA.
  - Result in ~5.1K utterances, 228/164 distinct acts.

- Training: BPTT, L2 reg, SGD w/ early stopping.
  
  train/valid/test: 3/1/1, data up-sampling

Available at: https://www.repository.cam.ac.uk/handle/1810/251304
Corpus-based Evaluation

- Test set: ~1K utterances each domain
- Metrics: BLEU-4 (against multiple references), ERR(slot error rates)
- Averaged over 5 random initialised networks.
- Over-gen 20, evaluate on top-5
- Models compared:
  - handcrafted generator (hdc)
  - kNN example-based generator (kNN)
  - class-based LM generator (classlm, O&R 2000)
  - heuristic gated rnn-based generator (rnn, Wen et al 2015)
Corpus-based Evaluation

Selection scheme : 5/20

Model

hdc    knn    classlm    rnn    sc-lstm    +deep

Restaurant   Hotel
Corpus-based Evaluation

Selection scheme: 5/20

<table>
<thead>
<tr>
<th>Model</th>
<th>ERR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>hdc</td>
<td></td>
</tr>
<tr>
<td>knn</td>
<td></td>
</tr>
<tr>
<td>classlm</td>
<td>8.00</td>
</tr>
<tr>
<td>rnn</td>
<td></td>
</tr>
<tr>
<td>sc-lstm</td>
<td>1.00</td>
</tr>
<tr>
<td>+deep</td>
<td></td>
</tr>
</tbody>
</table>

Restaurant | Hotel
Generated Examples

inform_only(name='red victorian bed breakfast',accepts_credit_cards='yes',near='haight',has_internet='yes')

red victorian bed breakfast is the only hotel near haight and accepts credit cards and has internet.
red victorian bed breakfast is the only hotel near haight and has internet and accepts credit cards.
red victorian bed breakfast is the only hotel near haight that accept credit cards and offers internet.
the red victorian bed breakfast has internet and near haight, it does accept credit cards.
the red victorian bed breakfast is the only hotel near haight that accepts credit cards, and offers internet.
Generated Examples

inform_no_match(area=tenderloin)

there are no restaurants in the tenderloin area .
there are 0 restaurants in the tenderloin area .
unfortunately there are 0 restaurants in the tenderloin area .
i could not find any restaurants in tenderloin .
there are no restaurants in tenderloin .
Generated Examples

?request(food)
what type of food are you looking for?
what type of food would you like?
what kind of food are you looking for?
what type of food are you interested in?
what kind of food would you like?
Human Evaluation

- **Setup**
  - Judges (~60) recruited from Amazon MT.
  - Asked to evaluate two system responses pairwise.
  - Comparing *classlm, rnn, sc-lstm, and +deep*

- **Metrics:**
  - Informativeness, Naturalness (rating out of 3)
  - Preference
## Human Evaluation

<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>2.59</td>
<td>2.50</td>
</tr>
<tr>
<td>rnn</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

<table>
<thead>
<tr>
<th>Pref. %</th>
<th>classlm</th>
<th>rnn</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</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>

* $p < 0.05$  ** $p < 0.005$
Outline

- Intro
- Semantically Conditioned LSTM
- **Domain adaptation for NLG**
  - Data counterfeiting – model initialisation
  - Discriminative training – better fine-tuning
- Conclusion
Domain Adaptation

- Adaptation for NN?
  - Continue to train the model on adaptation dataset
- Parameters are shared on LM part of the network
  - But not for the DA weights
  - New slot-value pairs can only be learned from scratch

```
Laptop Domain
[0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0]
```

```
TV Domain
```

...
Data counterfeiting

- Produce pseudo target domain data by replacing source domain slot-values pairs with target domains slot-value pairs.

- Procedure:

An example realisation in laptop (source) domain:

Zeus 19 is a heavy laptop with a 500GB memory

delexicalisation

(NAME-value) is a (WEIGHT-value) (TYPE-value) with a (MEMEORY-value) (MEMEORY-slot)

counterfeiting

(NAME-value) is a (FAMILY-value) (TYPE-value) with a (SCREEN-value) (SCREEN-slot)

A possible realisation in TV (target) domain:

Apollo 73 is a U76 television with a 29-inch screen
Choice of target domain slots?
- The realisation should be similar to the source one.
- Simple case: based on their functional class.
  - Informable, requestable, and binary slots.
- Example:

<table>
<thead>
<tr>
<th></th>
<th>Laptop</th>
<th>Television</th>
</tr>
</thead>
<tbody>
<tr>
<td>Informable</td>
<td>family, price_range, battery_rating,...</td>
<td>family, price_range, screen_size_range,...</td>
</tr>
<tr>
<td>Requestable</td>
<td>price, memory,...</td>
<td>price, resolution,...</td>
</tr>
<tr>
<td>Binary</td>
<td>is_for_business</td>
<td>has_usb_port</td>
</tr>
</tbody>
</table>
Laptop/TV dataset

- A more difficult dataset than restaurant/hotel
- Permute all possible DAs, ~13K/7K
- Only 1 example utterance for each DA

<table>
<thead>
<tr>
<th></th>
<th>Laptop</th>
<th>Television</th>
</tr>
</thead>
<tbody>
<tr>
<td>Informable slots</td>
<td>family, *pricerange, batteryrating, driverange, weightrange, <strong>isforbusinesscomputing</strong></td>
<td>family, *pricerange, screensizerange, ecorating, hdmiport, <strong>hasusbport</strong></td>
</tr>
<tr>
<td>Requestable slots</td>
<td>*name, *type, *price, warranty, battery, design, dimension, utility, weight, platform, memory, drive, processor</td>
<td>*name, *type, *price, resolution, powerconsumption, accessories, color, screenize, audio</td>
</tr>
</tbody>
</table>

**bold**=binary slots, * = overlap with SF Restaurant and Hotel domains, all **informable slots** can take ”dontcare” value
Data counterfeiting - Results

- Laptop 2 TV
- Restaurant+Hotel 2 Laptop+TV
Discriminative Training

- Explore model capacity and correct it.

![Diagram with a request(area) leading to a Model with questions]

- DT cost function:

\[
F(\theta) = -\mathbb{E}[L(\theta)] \\
= - \sum_{\Omega \in \text{Gen}(d_i)} p_\theta(\Omega|d_i) L(\Omega, \Omega_i)
\]

\(\Omega\): candidate sentence  
\(\Omega_i\): reference sentence  
\(d_i\): dialogue act  
\(L(\cdot)\): scoring function
Discriminative Training - Results

(a) Effect of DT on BLEU

(b) Effect of DT on slot error rate
Human Evaluation

<table>
<thead>
<tr>
<th>Method</th>
<th>TV to Laptop</th>
<th></th>
<th>laptop to TV</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>scrALL</td>
<td>2.64</td>
<td>2.37</td>
<td>2.54</td>
<td>2.36</td>
</tr>
<tr>
<td>DT-10%</td>
<td>2.52**</td>
<td>2.25**</td>
<td>2.51</td>
<td>2.19**</td>
</tr>
<tr>
<td>ML-10%</td>
<td>2.51**</td>
<td>2.22**</td>
<td>2.45**</td>
<td>2.22**</td>
</tr>
<tr>
<td>scr-10%</td>
<td>2.24**</td>
<td>2.03**</td>
<td>2.00**</td>
<td>1.92**</td>
</tr>
</tbody>
</table>

* p < 0.05, ** p < 0.005

- scrALL: train from scratch with 100% ID data.
- scr-10%: train from scratch with 10% ID data.
- ML-10%: data counterfeiting + ML training on 10% ID data.
- DT-10%: data counterfeiting + DT training on 10% ID data.
Outline

- Intro
- Semantically Conditioned LSTM
- Domain adaptation for NLG
- Conclusion
Conclusion

- NLG can be learned N2N from data.
  - Learn LM & slot gating control signal jointly
  - Corpus-based/Human evaluation.
  - More colloquial, more scalable.

- Domain Extension
  - Data counterfeiting facilitates domain adaptation.
  - Discriminative training can further improve.
Papers


Selected References


Thank you! Questions?

Tsung-Hsien Wen is supported by a studentship funded by Toshiba Research Europe Ltd, Cambridge Research Laboratory