Natural language generation

Milica Gašić

Dialogue Systems and Machine Learning Group,
Heinrich Heine University Düsseldorf
Natural language generation

Speech synthesis in dialogue
Natural language generation

- Speech recognition
- Semantic decoding
- Dialogue management
- Natural language generation
- Speech synthesis
- Ontology

Speech → text → dialogue acts → Dialogue management → Ontology
Role of natural language generation

- Converts dialogue act (semantics) into natural language
- Gives a persona to dialogue system
- Directly influences how the user perceives a dialogue system
Evaluating natural language generation

What makes a natural language generator good? [Stent et al., 2005]

- **adequacy**: correct meaning
- **fluency**: linguistic fluency
- **readability**: fluency in dialogue context
- **variation**: multiple realisations for the same concept
BLEU Score [Papineni et al., 2002]

- Evaluating similarity between paired sentences (n-gram match).
- There is a gap between human perception and automated measures.

<table>
<thead>
<tr>
<th>Correlation coefficient</th>
<th>Adequacy</th>
<th>Fluency</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLEU</td>
<td>0.388</td>
<td>-0.492</td>
</tr>
</tbody>
</table>

- Human evaluation is always the best way to evaluate language generation.
Template-based natural language generator

Define a set of templates which maps dialogue acts into utterances.

<table>
<thead>
<tr>
<th>Dialogue act</th>
<th>Delexicalised utterance</th>
</tr>
</thead>
<tbody>
<tr>
<td>confirm(area=$V)</td>
<td>Would you like a restaurant in the $V?</td>
</tr>
<tr>
<td>confirm(food=$V)</td>
<td>Would you like a $V restaurant?</td>
</tr>
<tr>
<td>confirm(food=$V,area=$W)</td>
<td>Would you like a $V restaurant in the $W?</td>
</tr>
</tbody>
</table>
Template-based natural language generator

Pros: simple, usually error free, controllable
Cons: time consuming, rigid, not scalable
Trainable generator [Walker et al., 2002]

- Divide the problem into a pipeline
  - **Sentence plan generator** Produces multiple sentence plans for a given dialogue act (or set of dialogue acts).
  - **Sentence plan reranker** Ranks possible candidates.
  - **Surface realiser** Turns the top candidate into an utterance.
Trainable generator overview

Inform(
  name=Z_House,
  price=cheap
)

Z House is a cheap restaurant.
Sentence plan generator

- Utilise machine learning to do reranking (RankBoost)
- Extract features from sentence plan trees: indicator function $f_i$ relating to traversal features, ancestor features, leaf features, etc. size 3291.

$$F(x) = \sum_i \alpha_i f_i(x)$$

$$Loss = \sum_{x,y \in D} \exp(-(F(x) - F(y)))$$

where $x$ and $y$ are sentence plans, $x$ is preferred to $y$, $\alpha_i$ are learnable parameters, and $D$ are sentence plans referring to a given dialogue act.
sentence plan generator - example

implicit-confirm(orig-city:NEWARK)
implicit-confirm(dest-city:DALLAS)
implicit-confirm(month:9)
implicit-confirm(day-number:1)
request(depart-time)

```
soft-merge-general
  /
\_________
  1
  /
\_________
  2
  /
\_________
  3
imp-confirm(day) imp-confirm(month) request(time) soft-merge-general
imp-confirm(dest-city) imp-confirm(orig-city)

period
  /
\_________
  1
  /
\_________
  2
period
  /
\_________
  3
relative-clause
imp-confirm(month) imp-confirm(day) soft-merge-general request(time)
imp-confirm(orig-city) imp-confirm(dest-city)
```
Other similar approaches

- Learning sentence plan generation rules.
- Statistical surface realisers.
Properties

Pros  Can generate sentences with complex linguistic structure.

Cons  Many rules, heavily engineered.
Class-based language modelling for NLG [Oh and Rudnicky, 2000]

- Language modelling

\[ p(W) = \prod_{t} p(w_t|w_0, \ldots, w_{t-1}) \]

- Class-based language modelling

\[ p(W|u) = \prod_{t} p(w_t|w_0, \ldots, w_{t-1}, u) \]

- Decoding

\[ W^* = \arg \max_{W} p(W|u) \]
Class-based language modelling for NLG

- Classes:
  - inform_area
  - inform_address
  - inform_phone
  - request_area
  ...

- Generation process:
  - Generate utterances by sampling words from a particular class language model in which the dialogue act belongs to.
  - Re-rank utterances according to scores.
Properties

**Pros**  no complicated rules, easy to implement, easy to understand

**Cons**  error-prone
Can we do better?

- RNN as language generator: natural model for modeling sequences
- Long-term dependencies?
- Flexible to condition on auxiliary inputs

Long-term dependencies in NLG:
Example: The restaurant (in the north) is a nice Chinese place.
RNN & Vanishing gradient [Pascanu et al., 2013]

\[ h_j = \sigma(W_r h_{j-1} + W_i w_j + b_h) \]
\[ y_j = \text{softmax}(W_o h_j + b_o) \]

\[ \frac{\partial E_3}{\partial W_r} = \sum_{k=0}^{3} \frac{\partial E_3}{\partial y_3} \frac{\partial y_3}{\partial h_3} \frac{\partial h_3}{\partial h_k} \frac{\partial h_k}{\partial W_r} \]
\[ = \sum_{k=0}^{3} \frac{\partial E_3}{\partial y_3} \frac{\partial y_3}{\partial h_3} \left( \prod_{j=k+1}^{3} \frac{\partial h_j}{\partial h_{j-1}} \right) \frac{\partial h_k}{\partial W_r} \]
\[ \frac{\partial h_j}{\partial h_{j-1}} = W_r^T \text{diag}(\sigma'(W_r h_{j-1} + W_i w_j + b_h)), \]

where \( \sigma' \) is elementwise derivative. The norm of the last term is smaller than 1 causing vanishing gradient.
LSTM [Hochreiter and Schmidhuber, 1997]

- Sigmoid gates

\[ i_t = \sigma(W_{wi}w_t + W_{hi}h_{t-1}) \]
\[ f_t = \sigma(W_{wf}w_t + W_{hf}h_{t-1}) \]
\[ o_t = \sigma(W_{wo}w_t + W_{ho}h_{t-1}) \]

- Proposed cell value

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

- Update cell and hidden layer

\[ C_t = i_t \odot \hat{C}_t + f_t \odot C_{t-1} \]
\[ h_t = o_t \odot \tanh(C_t) \]
LSTM

How it prevents vanishing gradient problem?

▶ Consider memory cell, where recurrence actually happens

\[ C_t = i_t \odot \hat{C}_t + f_t \odot C_{t-1} \]

▶ We can back-propagate the gradient by chain rule

\[
\frac{\partial E_t}{\partial C_{t-1}} = \frac{\partial E_t}{\partial C_t} \frac{\partial C_t}{\partial C_{t-1}} = \frac{\partial E_t}{\partial C_t} f_t
\]
RNN for NLG [Wen et al., 2015a]
Semantically conditioned LSTM [Wen et al., 2015b]
Learned alignments
### Human evaluation

<table>
<thead>
<tr>
<th>Method</th>
<th>Informativeness</th>
<th>Naturalness</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC-LSTM</td>
<td>2.59</td>
<td>2.50</td>
</tr>
<tr>
<td>Class LM</td>
<td>2.46</td>
<td>2.45</td>
</tr>
</tbody>
</table>
Examples

inform_no_match(area=\texttt{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.
Properties

**Pros**  more accurate, does not require intermediate alignments

**Cons**  does not utilise pre-trained word-vector embeddings
Generative pre-training (GPT)

- Autoregressive language model that utilises transformer architecture
- Pretrained on large amounts of crawled text
- Predicts the next token $u$ given context $U = [u_{-k}, \cdots, u_{-1}]$

$$h_0 = UW_e + W_p$$

$$h_l = \text{transformer}(h_{l-1}), l \in [1, n]$$

$$p(u) = \text{softmax}(h_n W_e^T)$$

- GPT-2 and GPT-3 have effectively the same model structure with substantially more parameters

<table>
<thead>
<tr>
<th></th>
<th>GPT</th>
<th>GPT-2</th>
<th>GPT-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>117M</td>
<td>1.5B</td>
<td>175B</td>
</tr>
<tr>
<td>Data</td>
<td>12GB</td>
<td>40GB</td>
<td>570GB</td>
</tr>
</tbody>
</table>
Generative pre-training (GPT)
Semantically conditioned GPT

- Pre-trains the GPT-2 model with a large corpus of dialogue act and utterance pairs.
- Fine-tuned with only a few domain-specific labels to adapt to new domains.
- Operates on lexicalised inputs.
Models trained on FewshotWOZ with only 50 dialogue acts in the training set and 500K in test set.

<table>
<thead>
<tr>
<th>Method</th>
<th>Informativeness</th>
<th>Naturalness</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC-GPT</td>
<td>2.64</td>
<td>2.47</td>
</tr>
<tr>
<td>SC-LSTM</td>
<td>2.29</td>
<td>2.13</td>
</tr>
<tr>
<td>Human</td>
<td>2.92</td>
<td>2.72</td>
</tr>
</tbody>
</table>
Summary of NLG

- Evaluating NLG is hard. The best way is human evaluation.
- Tree-based NLG is a linguistically motivated approach. By introducing machine learning in the pipeline enables the model to learn from data.
- Language Modeling casts NLG as a sequential prediction problem.
- LSTM overcomes vanishing gradient by sophisticated gating mechanism. The same idea was applied to NLG resulting in semantically conditioned-LSTM, a generator that can learn realisation and semantic alignments jointly.
- Pre-trained transformer models perform particularly well on few shot learning tasks.
Role of a speech synthesiser in a dialogue system

- In a dialogue system the context is available from the dialogue manager.
- Text-to-speech system can make use of the context to produce more natural and expressive speech [Yu et al., 2010].


References II


Credits

We thank Tsung-Hsien Wen for sharing his slides on Statistical Natural Language Generation.