Deep Learning for NLG

Tsung-Hsien (Shawn) Wen
thw28@cam.ac.uk
Part I: Overview

- Basic concepts and techniques in DL for NLG
- Recent progress of DL in NLG-related topics
Mapping MR(meaning representation) -> NL
- inform(name=Seven_Days, food=Chinese)
- Seven Days is a nice Chinese restaurant.

Evaluation
- Automatic metrics such as BLEU [Papineni et al, 2002]

<table>
<thead>
<tr>
<th>Correlation</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>

[Stent et al, 2005]

- Human Evaluation
Template-based NLG

- Define a set of rules to map MR to NL
  - Pros: simple, error-free, easy to control
  - Cons: time consuming, scalability

```plaintext
confirm()       “Please tell me more about the product you are looking for.”
confirm(area=$V) “Do you want somewhere in the $V?”
confirm(food=$V) “Do you want a $V restaurant?”
confirm(food=$V,area=$W) “Do you want a $V restaurant in the $W.”
...
```
Trainable Generator [Walker et al 2002]

- Divide the problem into pipeline

- Focus on applying ML to sentence plan reranker.
Following-up works

- Statistical sentence plan generator [Stent et al 2009]
- Statistical surface realisers [Dethlefs et al 2013, Cuayáhuitl et al 2014, ...]
- Learn from unaligned data [Dusek and Jurcicek 2015]

- Pros: can model complex linguistic structures
- Cons: heavily engineered, require domain knowledge
Sequential NLG models

- Class-based LM [Oh and Rudnicky, 2000]
  - Class-based Language Modeling
    \[ p(X|d) = \sum_t p(x_t|x_0, x_1, ... x_{t-1}, d) \]
  - Decoding
    \[ X^* = \arg\max_X p(X|d) \]
- Pros: easy to implement/understand, simple rules
- Cons: computationally inefficient
Sequential NLG models

- Phrase-based NLG using DBN [Mairesse et al, 2010]

Inform(type= restaurant, name=Charlie Chan, food=chinese, near=Cineworld, area=centre)
Sequential NLG models

- Phrase-based NLG using DBN [Mairesse et al, 2010]
  - Pros: efficient, good performance
  - Cons: require semantic alignments

<table>
<thead>
<tr>
<th>$r_t$</th>
<th>$s_t$</th>
<th>$h_t$</th>
<th>$l_t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;s&gt;</td>
<td>START</td>
<td>START</td>
<td>START</td>
</tr>
<tr>
<td>The Rice Boat</td>
<td>inform(name(X))</td>
<td>X</td>
<td>inform(name)</td>
</tr>
<tr>
<td>is a</td>
<td>inform</td>
<td>inform</td>
<td>EMPTY</td>
</tr>
<tr>
<td>restaurant</td>
<td>inform(type(restaurant))</td>
<td>restaurant</td>
<td>inform(type)</td>
</tr>
<tr>
<td>in the</td>
<td>inform(area)</td>
<td>area</td>
<td>inform</td>
</tr>
<tr>
<td>riverside</td>
<td>inform(area(riverside))</td>
<td>riverside</td>
<td>inform(area)</td>
</tr>
<tr>
<td>area</td>
<td>inform(area)</td>
<td>area</td>
<td>inform</td>
</tr>
<tr>
<td>that</td>
<td>inform</td>
<td>inform</td>
<td>EMPTY</td>
</tr>
<tr>
<td>serves</td>
<td>inform(food)</td>
<td>food</td>
<td>inform</td>
</tr>
<tr>
<td>French</td>
<td>inform(food(French))</td>
<td>French</td>
<td>inform(food)</td>
</tr>
<tr>
<td>food</td>
<td>inform(food)</td>
<td>food</td>
<td>inform</td>
</tr>
<tr>
<td>&lt;/s&gt;</td>
<td>END</td>
<td>END</td>
<td>END</td>
</tr>
</tbody>
</table>
Q & A
Neural Networks
NN basics

- **Artificial Neuron**
  \[ h_i = \sigma \left( \sum_j w_{ij} x_j + b_i \right) \]

- **Loss function**
  \[ \mathcal{L}(\theta) = -y^T \log p \]

- **Back-propagation**
  \[ \frac{\partial \mathcal{L}}{\partial \omega_{ij}} = \sum_k \frac{\partial \mathcal{L}}{\partial p_k} \frac{\partial p_k}{\partial h_i} \frac{\partial h_i}{\partial \omega_{ij}} \]
Gradient descent

\[ \omega'_{ij} = \omega_{ij} - \alpha \frac{\partial L}{\partial \omega_{ij}} \]
3 reasons why DL for NLP/NLG

• Generalisation
• Context Modeling
• Control
N-gram Language Modeling

- How likely is a sentence?
  - N-gram LM
    \[
    p(x_1, x_2, ..., x_T) = \prod_{t=1}^{T} p(x_t | x_1, ... x_{t-1}) \approx \prod_{t=1}^{T} p(x_t | x_{t-n}, ... x_{t-1})
    \]
  - Markovian assumption
  - Collect statistics from a large corpus:
    \[
    p(x_t | x_{t-n}, ... x_{t-1}) = \frac{\text{count}(x_{t-n}, ..., x_{t-1}, x_t)}{\text{count}(x_{t-n}, ..., x_{t-1})}
    \]
The data sparsity problem
- Vocab size $V$
- Possible n-grams $|V|^n$

Ways to mitigate:
- Smoothing, backoff

But still, lack of generalisation

<table>
<thead>
<tr>
<th>N-gram</th>
<th>logP</th>
</tr>
</thead>
<tbody>
<tr>
<td>camel</td>
<td>-2.0014</td>
</tr>
<tr>
<td>camel is</td>
<td>-2.5426</td>
</tr>
<tr>
<td>camel is like</td>
<td>-3.4456</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>alpaca</td>
<td>n/a</td>
</tr>
<tr>
<td>alpaca is</td>
<td>n/a</td>
</tr>
<tr>
<td>alpaca is a</td>
<td>n/a</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>llama</td>
<td>n/a</td>
</tr>
<tr>
<td>an llama</td>
<td>n/a</td>
</tr>
<tr>
<td>an llama runs</td>
<td>n/a</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Curse of Dimensionality

Photo credit: newsnshit
Conquer the Curse of Dimensionality - NNLM

- **Neural Net LM**
  - 1-of-V encoding for each word $x_{t-k}$
  - Distributed word representation
    \[ x_{t-k} = W^T x_{t-k} \]
  - Nonlinear hidden layer
    \[ h_t = \tanh(U^T[x_{t-1}; x_{t-2}; ... x_{t-n}] + b) \]
  - Softmax output
    \[ p_t = \text{softmax}(V^T h_t + c) \]

[Bengio et al 2001]
Distributed Word Representation

- NNLM generalises to unseen words/n-grams

[Cho et al 2014]
Context Modeling - RNNLM

- Non Markovian assumption
- RNNLM
  - 1-of-V encoding for each word $x_t$
  - Recurrent transition function
    $$h_t = \tanh(W^Tx_t + U^Th_{t-1} + b)$$
  - Softmax output
    $$p_t = \text{softmax}(V^Th_t + c)$$
- Read, update, predict!
- Can model dependency of arbitrary length

[Mikolov et al 2010]
\[ h_t = \tanh(W^T x_t + U^T h_{t-1} + b) \]
\[ p_t = \text{softmax}(V^T h_t + c) \]
\[ E_3 = -y_3^T \log_{10} p_3 \]
\[
\frac{\partial E_3}{\partial W} = \sum_{k=0}^{3} \frac{\partial E_3}{\partial p_3} \frac{\partial p_3}{\partial h_3} \frac{\partial h_3}{\partial h_k} \frac{\partial h_k}{\partial W} = \sum_{k=0}^{3} \frac{\partial E_3}{\partial p_3} \frac{\partial p_3}{\partial h_3} \left( \prod_{j=k+1}^{3} \frac{\partial h_j}{\partial h_{j-1}} \right) \frac{\partial h_k}{\partial W}
\]
\[
\frac{\partial h_j}{\partial h_{j-1}} = U^T \cdot \text{diag}(\tanh'(m_j))
\]
\[ m_j = W^T x_j + U^T h_{j-1} + b \]

Ignore proof here.
\[ \|U\| \cdot \|\text{diag}(\tanh'(m_j))\| < 1 \]
Vanishing gradient!

[Pascanu et al, 2013]
Learning Long-term Dependency - LSTM

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

- Proposed cell value
  \[ \hat{c}_t = \tanh(W_{wc}x_t + W_{hc}h_{t-1}) \]

- Update cell and hidden layer
  \[ c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t \]
  \[ h_t = o_t \odot \tanh(c_t) \]

[Hochreiter and Schmidhuber, 1997]
How does it prevent vanishing gradient?

Consider memory cell update

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

We can back-prop the gradient by chain rule

\[ \frac{\partial E_t}{\partial C_{t-1}} = \frac{\partial E_t}{\partial C_t} \frac{\partial C_t}{\partial C_{t-1}} = \frac{\partial E_t}{\partial C_t} f_t \]

If \( f_t \) maintains a value of 1, gradient is perfectly propagated.
The meaning of life is ...

- The meaning of life is the tradition of the ancient human reproduction: it is less favorable to the good boy for when to remove her bigger. In the show’s agreement unanimously resurfaced. The wild pasteured with consistent street forests were incorporated by the 15th century BE. In 1996 the primary rapford undergoes an effort that the reserve conditioning, written into Jewish cities, sleepers to incorporate the .St Eurasia that activates the population. Mar??a Nationale, Kelli, Zedlat-Dukastoe, Florendon, Ptu’s thought is. To adapt in most parts of North America, the dynamic fairy Dan please believes, the free speech are much related to the
RNN handwriting synthesis [Graves, 2013]

Turn my candle down again there will

very need anehe bopkshes ha

Mairene Cenerle of hig wraditro

sea Botug a the oorrakers sa

buerne bissest beor lenest

bopes 4 eald bwinef 5 wine cuu

heioet. Y ceelis the gatber me

scyke satet bynow in boryng be a

over 4 hylhe eameis. End. madp
RNN handwriting synthesis [Graves, 2013]

Can we gain control on generated content?

more of national temperament
more of national temperament
more of national temperament
more of national temperament
more of national temperament
more of national temperament
more of national temperament
Q & A
The 3\textsuperscript{rd} Reason: Control!
Integrating across modalities – Conditional RNN

Diagram:

```
| </s> | Eat | serves | British | . | </s> |
```

Diagram arrows:

- From </s> to Eat
- From Eat to serves
- From serves to British
- From British to .
- From . to </s>
Integrating across modalities – Conditional RNN

- Text-to-Text
  - Sequence-to-Sequence Learning [Sutskever et al, 2014]

- Grammar as a foreign language [Vinyals et al, 2015]
Integrating across modalities – Conditional RNN

- Text-to-Text
  - Chinese Poetry Generation [Zhang and Lapata, 2014]

Far off I watch the waterfall plunge to the long river.
Integrating across modalities – Conditional RNN

- Text-to-Image [Graves, 2013]
Integrating across modalities – Conditional RNN

- Image-to-Text
  - Image caption generation [Karpathy and Li, 2015]
Short Conclusion

- I haven’t talked about “Deep Learning for NLG” yet.

- But you know at least why DL is cool for NLP now.
  - Distributed representation – Generalisation
  - Recurrent connection – Long-term Dependency
  - Conditional RNN – Flexibility/Creativity
Q & A
Part II: NLG models

- Gated-based NLG models
- Attention-based NLG models
- Domain Adaptation
- Deep NLG for Dialogue Response Generation
Conditional RNNLM

- Generation conditions on MR
  - Represent MR?

```plaintext
Eat serves British . </s>
```
Inform(name=EAT, food=British)

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

\textbf{dialog act 1-hot representation}

\begin{align*}
\text{SLOT\_NAME} & \quad \text{serves} \quad \text{SLOT\_FOOD} \\
\text{EAT} & \quad \text{serves} \quad \text{British} \\
</s>
\end{align*}

\textbf{delexicalisation}

\textbf{Weight tying}

(Wen et al, 2015a)
Handling Semantic Repetition

- Empirically, semantic repetition is observed.
  - EAT is a great British restaurant that serves British.
  - EAT is a child friendly restaurant in the cheap price range. They also allow kids.

- Deficiency in either model or decoding (or both)

- Mitigation
  - Post-processing rules [Oh & Rudnicky, 2000]
  - Gating mechanism [Wen et al, 2015a & 2015b]
  - Attention [Mei et al, 2016; Wen et al, 2015c]
Recap LSTM gates:

- $i_t = \sigma(W_{wi}x_t + W_{hi}h_{t-1})$
- $x_t$: current input word embedding.
- $h_{t-1}$: sequence embedding up to $t-1$.
- Learn to decide whether the gates should open/close based on generation history.

Can we do the same for learning the gate of semantics (a.k.a. alignments).
- **Original LSTM cell**
  \[i_t = \sigma(W_{wi}x_t + W_{hi}h_{t-1})\]
  \[f_t = \sigma(W_{wf}x_t + W_{hf}h_{t-1})\]
  \[o_t = \sigma(W_{wo}x_t + W_{ho}h_{t-1})\]
  \[\hat{c}_t = \tanh(W_{wc}x_t + W_{hc}h_{t-1})\]
  \[c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t\]
  \[h_t = o_t \odot \tanh(c_t)\]

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

- **Modify Ct**
  \[c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t + \tanh(W_{dc}d_t)\]
Visualization [Wen et al, 2015b]
Cost function \([Wen \ et \ al, \ 2015b]\)

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

- **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 at each time step.
Results [Wen et al, 2015b]

<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>h-lstm</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
Attention Mechanism?
Attentive Caption Generation [Xu et al, 2015]

- A woman is throwing a **frisbee** in a park.
- A dog is standing on a **hardwood floor**.
- A **stop** sign is on a road with a mountain in the background.
- A little girl sitting on a **bed** with a **teddy bear**.
- A group of **people** sitting on a **boat** in the water.
- A giraffe standing in a forest with **trees** in the background.
Attention Mechanism in Neural Networks

- A general form of **differentiable** attention:
  - Given sources $s$ (usually in vector form), determine a distribution $p(s|\theta)$ based on network parameter $\theta$ and take the **expectation** over sources: $g = \sum_s p(s|\theta) s$

- Benefits:
  - Differentiable everywhere (back-prop).
  - Selective focus on part of data that is important.
  - Create short path for gradient flow.
Content-based Attention

- At every generation step $t$
  - Score source $h_j$ by
    \[ e_{tj} = v^T \tanh(W \cdot s_{t-1} + U \cdot h_j) \]
    \[ \alpha_{tj} = \text{softmax}(e_{tj}) \]
  - Take an expectation over sources
    \[ c_t = \sum_j \alpha_{tj} h_j \]

- Everything is differentiable. Back-prop end-to-end!

[Bahdanau et al, 2013]
Neural MT [Bahdanau et al, 2013]
Attentive Encoder-Decoder for NLG

- Slot & value embedding
  \[ z_i = s_i + v_i \]

- Attentive MR representation
  \[ e_{ti} = v^T \tanh(W_{hm} h_{t-1} + W_{zm} z_i) \]
  \[ \alpha_{ti} = \text{softmax}(e_{ti}) \]
  \[ d_t = a \oplus \sum_i \alpha_{ti} z_i \]

- Modified based on Mei et al, 2016.
- Related work: Dusek and Jurcicek 2016

\[ \text{[Wen et al, 2015c]} \]
Attention heat map [Mei et al 2016]

Record details:
- id-0: temperature(time=06-21, min=52, mean=63, max=71)
- id-2: windSpeed(time=06-21, min=8, mean=17, max=23)
- id-3: windDir(time=06-21, mode=SSE)
- id-4: gust(time=06-21, min=0, mean=10, max=30)
- id-5: skyCover(time=6-21, mode=50-75)
- id-10: precipChance(time=06-21, min=19, mean=32, max=73)
- id-15: thunderChance(time=13-21, mode=SCsc)

Figure 3: An example generation for a set of records from WEATHERGOV.
Model Comparison

Restaurant

![Graphs showing BLEU and ERR for different models on Restaurant data.](image)

Laptop

![Graphs showing BLEU and ERR for different models on Laptop data.](image)
Q & A
Domain Adaptation for NLG
Domain Adaptation [Wen et al, 2016a]

- 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 0 1 0 0 0 0 0 1 0 0 0 0 0 1 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

  \[ \text{delexicalisation} \]

  \(<\text{NAME-value}>\) is a \(<\text{WEIGHT-value}> <\text{TYPE-value}>\) with a \(<\text{MEMEORY-value}> <\text{MEMORY-slot}>\)

  \[ \text{counterfeiting} \]

  \(<\text{NAME-value}>\) is a \(<\text{FAMILY-value}> <\text{TYPE-value}>\) with a \(<\text{SCREEN-value}> <\text{SCREEN-slot}>\)

*A possible realisation in TV (target) domain:*

  Apollo 73 is a U76 television with a 29-inch screen
Data counterfeiting – Results [Wen et al, 2016a]
Discriminative Training [Wen et al, 2016a]

- Explore model capacity and correct it.

- 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
Disc. Training – Results [Wen et al, 2016a]

(a) Effect of DT on BLEU

(b) Effect of DT on slot error rate
Q & A
Deep NLG for Dialogue Response Generation
Traditional Dialogue Systems

- Step-by-step query processing
Neural Dialogue Systems

- How about integrating the whole thing?
A Neural Conversational Model

- **Seq2Seq**

  [Vinyals and Le, 2015]

**Human:** what do you think about tesla?
**Machine:** he’s a good conductor.
**Human:** what do you think about bill gates?
**Machine:** he’s a good man.
**Human:** What do you think about messi?
**Machine:** he’s a great player.
**Human:** what do you think about cleopatra?
**Machine:** oh, she’s very regal.
**Human:** what do you think about england during the reign of elizabeth?
**Machine:** it was a great place.

**Human:** what is the purpose of life?
**Machine:** to serve the greater good.
**Human:** what is the purpose of living?
**Machine:** to live forever.
**Human:** what is the purpose of existence?
**Machine:** to find out what happens when we get to the planet earth.
**Human:** where are you now?
**Machine:** i’m in the middle of nowhere.
**Human:** what is the purpose of dying?
**Machine:** to have a life.
**Human:** what is the purpose of being intelligent?
**Machine:** to find out what it is.
**Human:** what is the purpose of emotions?
**Machine:** i don’t know.
Hierarchical RNN for Dialogue [Serban et al, 2016]

Reference (U₁, U₂) | MAP | Target (U₃)
---|---|---
U₁: yeah, okay.  
U₂: well, I guess I’ll be going now. | i’ll see you tomorrow. | yeah. 
U₁: oh. <continued_utterance> oh.  
U₂: what’s the matter, honey? | i don’t know. | oh. 
U₁: it’s the cheapest.  
U₂: then it’s the worst kind? | no, it’s not. | they’re all good, sir. 
U₁: <person>! what are you doing?  
U₂: shut up! c’mon. | what are you doing here? | what are you that crazy?
Other Works

- Neural Responding Machine [Shang et al, 2015]
- Diversity-Promoting Decoding [Li et al, 2016a]
- Persona Modeling [Li et al, 2016b]
- Deep RL for Response Generation [Li et al, 2016c]

Issues:
- Generic responses
- No consistency/control
- Cannot achieve goal (no goal)
- Evaluation
Task-Oriented Dialogue as Conditional Generation
Can I have Korean. Little Seoul serves great Korean.
Can I have <v.food> serves great <v.food>.

A Network-based End-to-End Trainable Task-Oriented Dialogue System, Wen et al, 2016b
Can I have <v.food>

<v.name> serves great <v.food>.

A Network-based End-to-End Trainable Task-Oriented Dialogue System, Wen et al, 2016b
A Network-based End-to-End Trainable Task-Oriented Dialogue System, Wen et al, 2016b
Can I have <v.food> 

Intent Network

Can I have <v.food> 

Generation Network

<v.name> serves great <v.food> .

Database Accessing

Belief Tracker

MySQL query: "Select * where food=Korean"

A Network-based End-to-End Trainable Task-Oriented Dialogue System, Wen et al, 2016b
I have

Korean

0.7

BriOsh

0.2

French

0.1

…

Belief Tracker

MySQL query:
“Select * where food=Korean”

Database Operator

A Network-based End-to-End Trainable Task-Oriented Dialogue System, Wen et al, 2016b
A Network-based End-to-End Trainable Task-Oriented Dialogue System, Wen et al, 2016b
Example dialogues

Human: Restaurant in any area that serves Vietnamese food.

Machine: Thanh Binh serves Vietnamese food type. Is there anything else I can help you with?

Human: What is the phone number and post code?

Machine: The number is 01223 362456 and the area code is C. B 3, 0 A.F. Is there anything else I can do for you?

Human: Goodbye.

Machine: Thank you for using the Cambridge restaurant system.

Human: Hi, I’m hungry for some Indonesian food.

Machine: I am sorry, there are no Indonesian restaurants. Would you like a different food?

Human: Oh no. How about Indian?

Machine: There are 9 restaurants serving Indian types of food. Do you have a preference on side of town?

Human: East side please.

Machine: Sitar Tandoori is a Indian restaurant in the East side of town. Would you like their information?

Human: Yes, the address and phone number please.

Machine: Sitar Tandoori is located at 43 High Street Cherry Hinton Cherry Hinton and their phone number is 01223 249955.

Human: Okay, thank you.

Machine: Thank you, goodbye.
Visualising action embedding
Benefits of DL in complex NLP systems

- **Distributed representation** – “AGAIN” Generalisation
- **Recurrent connection** – Learning “RAW” inputs
- **Conditional RNN** – “MULTIMODAL” sources

DL allows us to build complex NLP learning systems like ever before.

It is ambitious to learn EVERYTHING

- Figure out what should be (shouldn’t) learned.

RL for online fine-tuning? [Su et al 2016].
Q & A
Part III: Codes

- Example codes for implementing deep NLG models in Theano
RNNLG is an open source benchmark toolkit for Natural Language Generation (NLG) in spoken dialogue system application domains. It is released by Tsung-Hsien (Shawn) Wen from Cambridge Dialogue Systems Group under Apache License 2.0.

**Requirement**

You need to have the following package to run the program:

- Theano 0.8.2 and accompanying packages such as numpy, scipy ...
- NLTK 3.0.0

[https://github.com/shawnwun/RNNLG](https://github.com/shawnwun/RNNLG)
RNNLG – Benchmark toolkit for Neural NLG

- **Summary**
  - Implementation: Python 2.7, Theano 0.8.2, NLTK 3.0.0
  - 4 benchmark datasets, 6 counterfeited datasets.
  - 6 baseline models, 2 training/decoding strategies.

- **Including works in the following publications:**
Hands-on
Simple Hands-On Session

- Download code at https://github.com/shawnwun/RNNLG
- Make sure you have
  - Theano 0.8.2, NLTK 3.0.0, python 2.7
- Testing Baselines:
  ```python
  python main.py -config config/ngram.cfg -mode ngram
  python main.py -config config/knn.cfg -mode knn
  ```
- Training SC-LSTM (run in background):
  ```python
  python main.py -config config/sclstm.cfg -mode train
  python main.py -config config/sclstm.cfg -mode test
  ```
Toolkit Navigation
Example codes for Implementing Deep NLG models
Working with Theano is like working as plumbers

- **Compilation time:** define i/o mapping

- **Run time:** follow the forward pipe to compute output; follow the back-prop pipe to update parameters.
Connecting water pipes

[RNNLG toolkit, nn/sclstm.py]

def _recur(self, w_t, y_t, sv_tml, h_tml, c_tml, a):
    
    # input word embedding
    wv_t = T.nnet.sigmoid(self.Wemb[w_t,:])
    # compute ig, fg, og together and slice it
    gates_t = T.dot( T.concatenate([wv_t,h_tml,sv_tml],axis=1),self.Wgate)
    ig = T.nnet.sigmoid(gates_t[:,0:self.dh])
    fg = T.nnet.sigmoid(gates_t[:,self.dh:self.dh*2])
    og = T.nnet.sigmoid(gates_t[:,self.dh*2:self.dh*3])
    # compute reading rg
    rg = T.nnet.sigmoid(T.dot(
        T.concatenate([wv_t,h_tml,sv_tml],axis=1),self.Wrgate))
    
    # compute proposed cell value
    cx_t = T.tanh(T.dot(T.concatenate([wv_t,h_tml],axis=1),self.Wcx))
    # update DA 1-hot vector
    sv_t = rg*sv_tml
    # update lstm internal state
    c_t = ig*cx_t + fg*c_tml + \n    T.tanh(T.dot(T.concatenate([a,sv_t],axis=1),self.Wfc))
    # obtain new hiddne layer
    h_t = og*T.tanh(c_t)
    # compute output distribution target word prob
    o_t = T.nnet.softmax( T.dot(h_t,self.Wo) )
    p_t = o_t[T.arange(self.db),y_t]

    return sv_t, h_t, c_t, p_t

    i_t = \sigma(W_{wi}x_t + W_{hi}h_{t-1})
    f_t = \sigma(W_{wf}x_t + W_{hf}h_{t-1})
    o_t = \sigma(W_{wo}x_t + W_{ho}h_{t-1})
    r_t = \sigma(W_{wr}x_t + W_{hr}h_{t-1})

    \hat{c}_t = \tanh(W_{wc}x_t + W_{hc}h_{t-1})
    d_t = r_t \odot d_{t-1}
    c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t + \tanh(W_{dc}d_t)
    h_t = o_t \odot \tanh(c_t)
    p_t = \text{softmax}(W_{ho}h_t)
Define inputs/outputs

Input placeholders

```python
# input tensor variables
w_idxes = T.imatrix('w_idxes')
sv    = T.imatrix('sv')
s     = T.imatrix('s')
v     = T.imatrix('v')

# cutoff for batch and time
cutoff_f = T.imatrix('cutoff_f')
cutoff_b = T.iscalar('cutoff_b')

# regularization and learning rate
lr     = T.scalar('lr')
reg    = T.scalar('reg')
```

Output cost, gradient, update function

```python
if self.gentype=='sclstm':
    self.cost, cutoff_logp = \
        self.generator.unroll(a,sv,w_idxes,cutoff_f,cutoff_b)

# gradients and updates
gradients = T.grad(clip_gradient(self.cost,1),self.params)
updates = OrderedDict((p, p-lr*g+reg*p) \n    for p, g in zip(self.params, gradients))
```
Part IV: Conclusion
Conclusion

- The three pillars of DL for NLG/NLP
  - Distributed representation – Generalisation.
  - Recurrent connection – Long-term Dependency.
  - Conditional RNN – Flexibility/Creativity.

- The last one is the key to many interesting applications in DL today.
Conclusion

- Useful techniques in DL for NLG
  - Learnable gates
  - Attention mechanism

- Generating longer/complex sentences.

- Phrase dialogue as conditional generation problem
  - Conditioning on raw input sentence: chat-bot
  - Conditioning on both structured and unstructured sources: a task-completing dialogue system!

- More interesting works to be done!
References

**NLG 101**

- “Evaluating evaluation methods for generation in the presence of variation”, Amanda Stent, Matthew Marge, Mohit Singhai, CICLing 2005
- “Training a sentence planner for spoken dialogue using boosting”, Marilyn A. Walker, Owen C. Rambow, Monica Rogati, Computer Speech and Language 2002
- “Training a statistical surface realiser from automatic slot labelling”, Heriberto Cuayáhuitl and Nina Dethlefs and Helen Hastie and Xingkun Liu, IEEE SLT 2014
- “Training a Natural Language Generator From Unaligned Data”, Ondrej Dusek, Filip Jurcicek, ACL 2015
References

**Neural Networks**

- “Recurrent neural network based language model”, Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Honza Cernocky, Sanjeev Khudanpur, InterSpeech 2010
- “Long Short-Term Memory”, Sepp Hochreiter and Jurgen Schmidhuber, Neural Computation 1997
References

Text Generation


Handwriting Generation

- “Generating Sequences With Recurrent Neural Networks”, Alex Graves, arXiv preprint:1308.0850, 2013

Poetry Generation

- “Chinese Poetry Generation with Recurrent Neural Networks”, Xingxing Zhang, Mirella Lapata, EMNLP 2014.

Image Generation

- “DRAW: A Recurrent Neural Network For Image Generation” Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, Daan Wierstra, ICML 2015.
References

**Machine Translation**

- “Sequence to Sequence Learning with Neural Networks”, Ilya Sutskever, Oriol Vinyals, Quoc V. Le, NIPS 2014.

**Image Caption Generation**

Natural Language Generation

N2N Response Generation (chitchat)

References

Dialogue Response Generation (goal-oriented)

Parsing

Code Generation
Thank you! Questions?

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