UNIVERSITY OF CAMBRIDGE

Deep Learning for NLG

Tsung-Hsien (Shawn) Wen
thw28@cam.ac.uk

Dialogue Systems Group

Part I: Overview

- Basic concepts and techniques in DL for NLG
- Recent progress of DL in NLG-related topics

NLG 101

- 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]

Correlation	Adequacy	Fluency
BLEU	0.388	-0.492

[Stent et al, 2005]

  - Human Evaluation

Template-based NLG

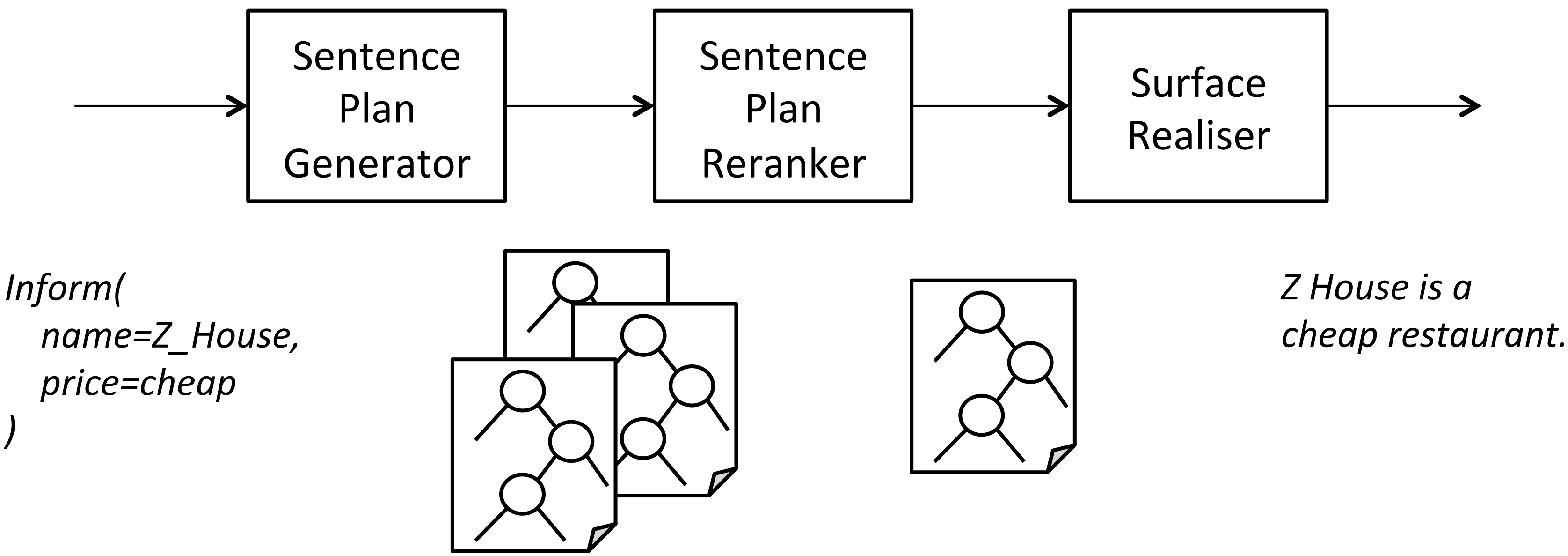
- Define a set of rules to map MR to NL
  - Pros: simple, error-free, easy to control
  - Cons: time consuming, scalability

confirm()	"Please tell me more about the product your 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

$$p(\mathrm{X}|\mathrm{d}) = \sum_{t} \mathrm{p}(x_t|x_0, x_1, \dots x_{t-1}, \mathrm{d})$$

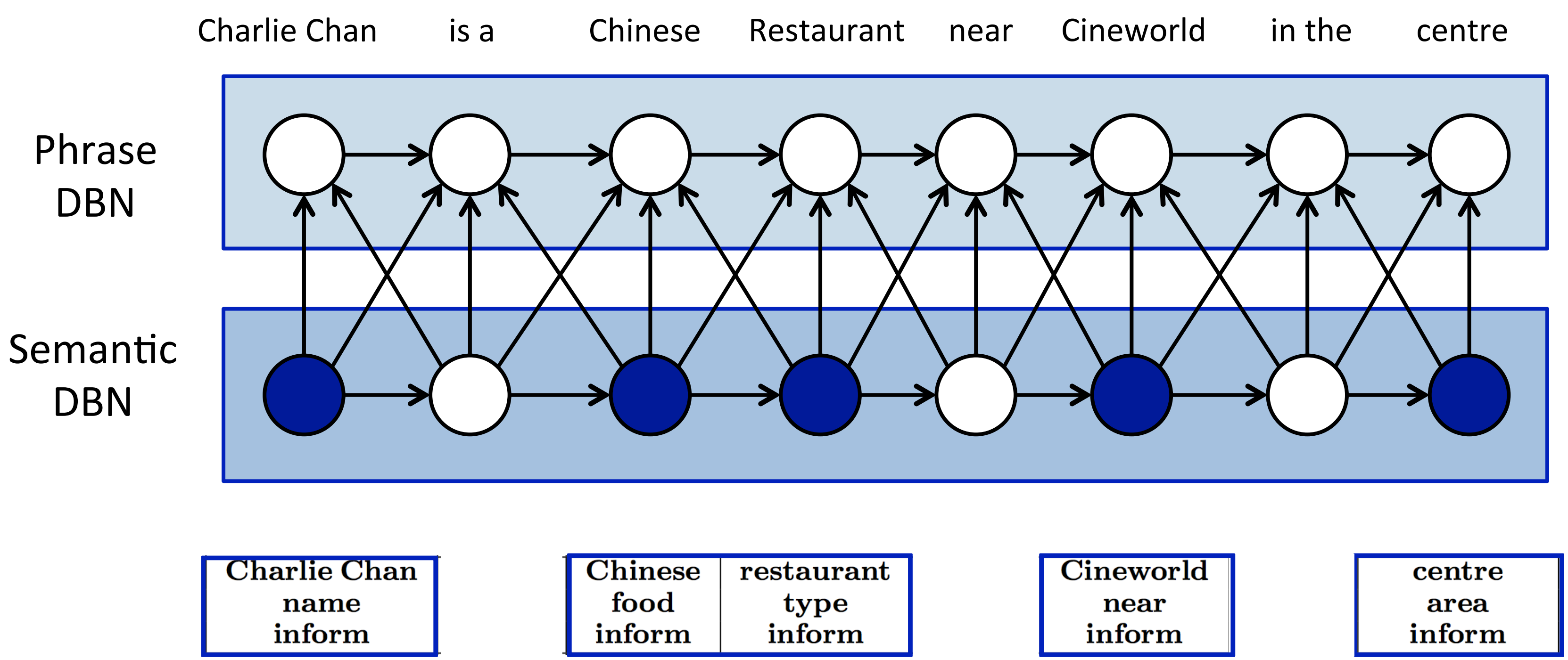
  - Decoding

$$\mathrm{X}^* = \underset{\mathrm{X}}{\mathrm{argmax}}\ p(\mathrm{X}|\mathrm{d})$$

  - 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

r_t	s_t	h_t	l_t
<s>	START	START	START
The Rice Boat	inform(name(X))	X	inform(name)
is a	inform	inform	EMPTY
restaurant	inform(type(restaurant))	restaurant	inform(type)
in the	inform(area)	area	inform
riverside	inform(area(riverside))	riverside	inform(area)
area	inform(area)	area	inform
that	inform	inform	EMPTY
serves	inform(food)	food	inform
French	inform(food(French))	French	inform(food)
food	inform(food)	food	inform
</s>	END	END	END

Q & A

Neural Networks

NN basics

- Artificial Neuron

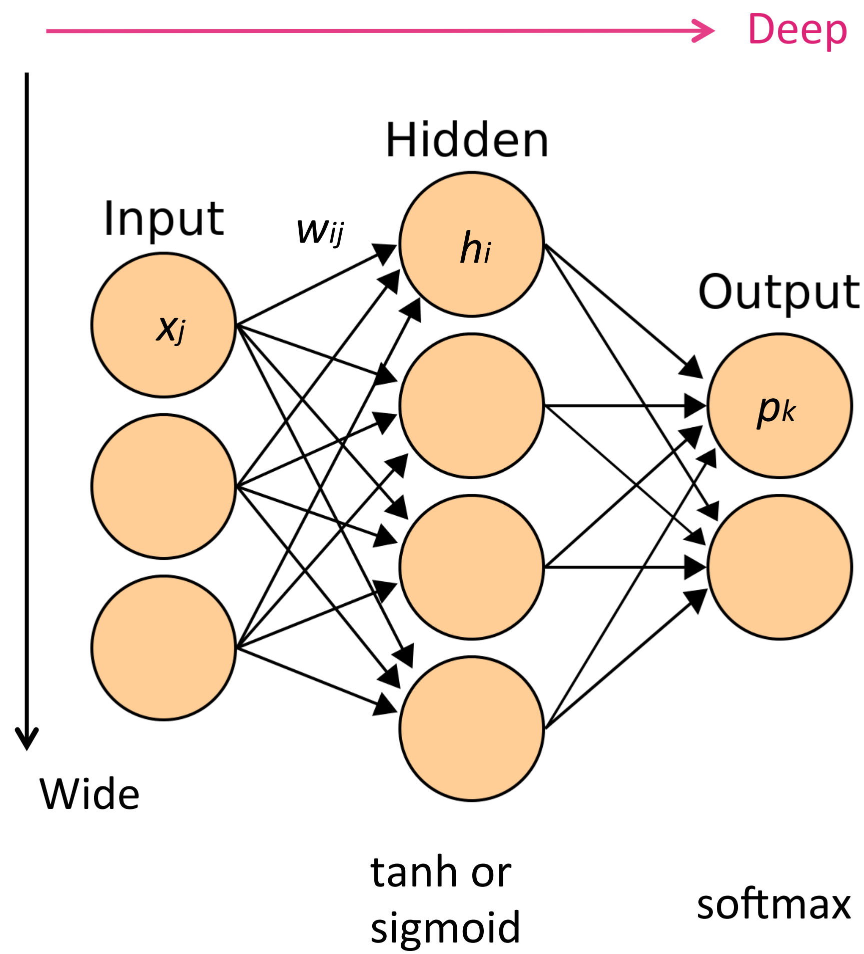
$$h_i = \sigma(\sum_j \omega_{ij} x_j + b_i)$$

output

Activation function

parameter

input



- Loss function

$$\mathcal{L}(\theta) = -\mathbf{y}^{\mathrm{T}} \log \mathbf{p}$$

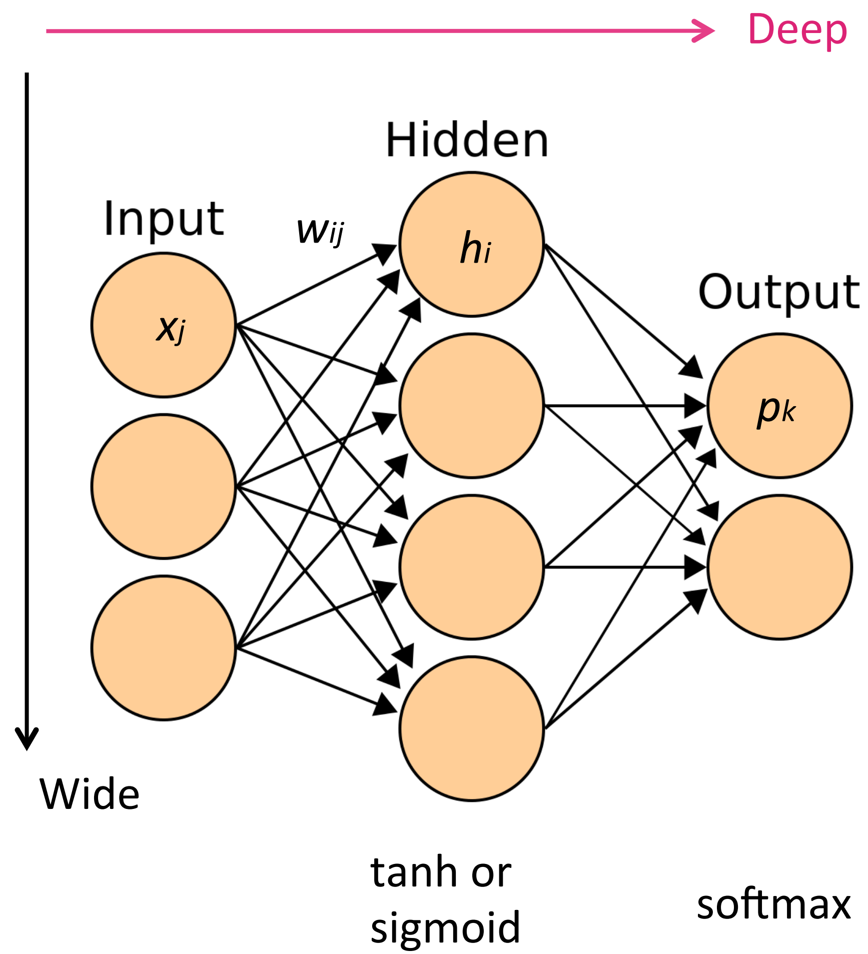
- Back-propagation

$$\frac{\partial \mathcal{L}}{\partial \omega_{ij}} = \sum_k \frac{\partial \mathcal{L}}{\partial p_k} \frac{\partial p_k}{\partial h_i} \frac{\partial h_i}{\partial \omega_{ij}}$$

NN basics

- Gradient descent

$$\omega'_{ij} = \omega_{ij} - \alpha \frac{\partial \mathcal{L}}{\partial \omega_{ij}}$$



3 reasons why DL for NLP/NLG

- Generalisation
- Context Modeling
- Control

N-gram Language Modeling

- How likely is a sentence?
  - N-gram LM

$$p(x_1, x_2, \dots, x_T) = \prod_{t=1}^{T} p(x_t | x_1, \dots x_{t-1}) \approx \prod_{t=1}^{T} p(x_t | x_{t-n}, \dots x_{t-1})$$

  - Markovian assumption
  - Collect statistics from a large corpus:

$$p(x_t | x_{t-n}, \dots x_{t-1}) = \frac{count(x_{t-n}, \dots x_{t-1}, x_t)}{count(x_{t-n}, \dots x_{t-1})}$$

N-gram Language Modeling

- The data sparsity problem
  - Vocab size V
  - Possible n-grams $|V|^n$
- Ways to mitigate:
  - Smoothing, backoff
- But still, lack of generalisation



N-gram	logP
camel	-2.0014
camel is	-2.5426
camel is like	-3.4456
…	…
alpaca	n/a
alpaca is	n/a
alpaca is a	n/a
…	…
llama	n/a
an llama	n/a
an llama runs	n/a
…	

Curse of Dimensionality

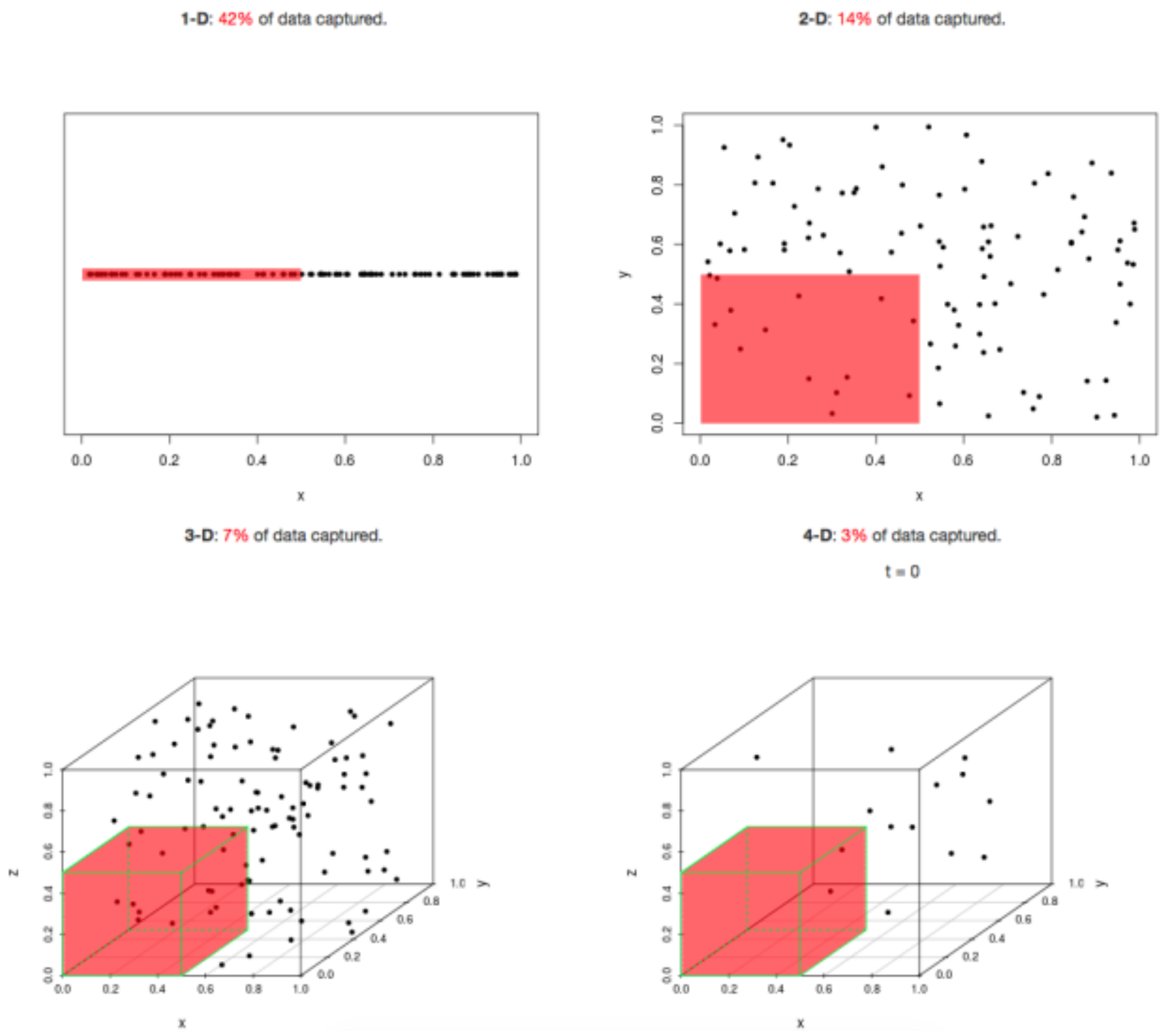


Photo credit: newsnshit

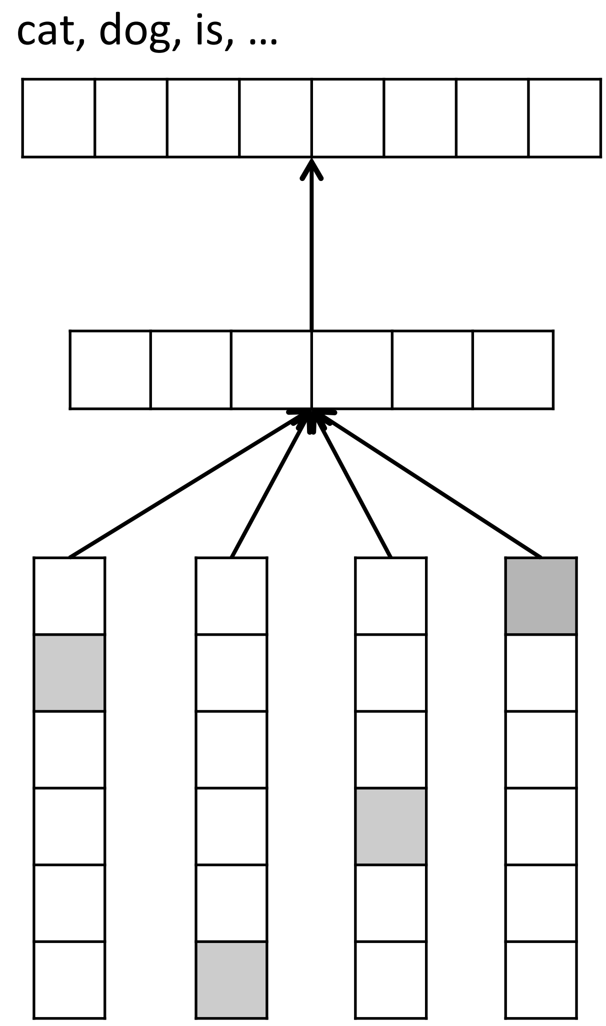
Conquer the Curse of Dimensionality - NNLM

- Neural Net LM
  - 1-of-V encoding for each word $x_{t\text{-}k}$
  - Distributed word representation

    $$\mathbf{x}_{t-k} = \mathbf{W}^{\mathrm{T}} x_{t-k}$$
  - Nonlinear hidden layer

    $$\mathbf{h}_t = \tanh(\mathbf{U}^{\mathrm{T}}[\mathbf{x}_{t-1}; \mathbf{x}_{t-2}; \dots \mathbf{x}_{t-n}] + \mathbf{b})$$
  - Softmax output

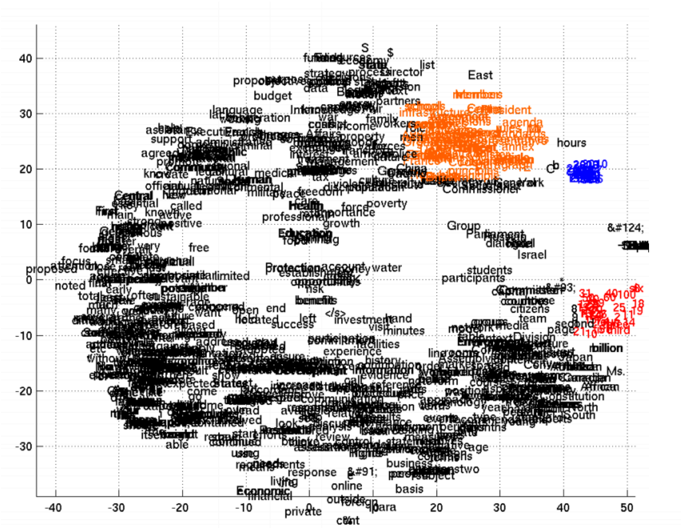
    $$\mathbf{p}_t = \mathrm{softmax}(\mathbf{V}^{\mathrm{T}}\mathbf{h}_t + \mathbf{c})$$

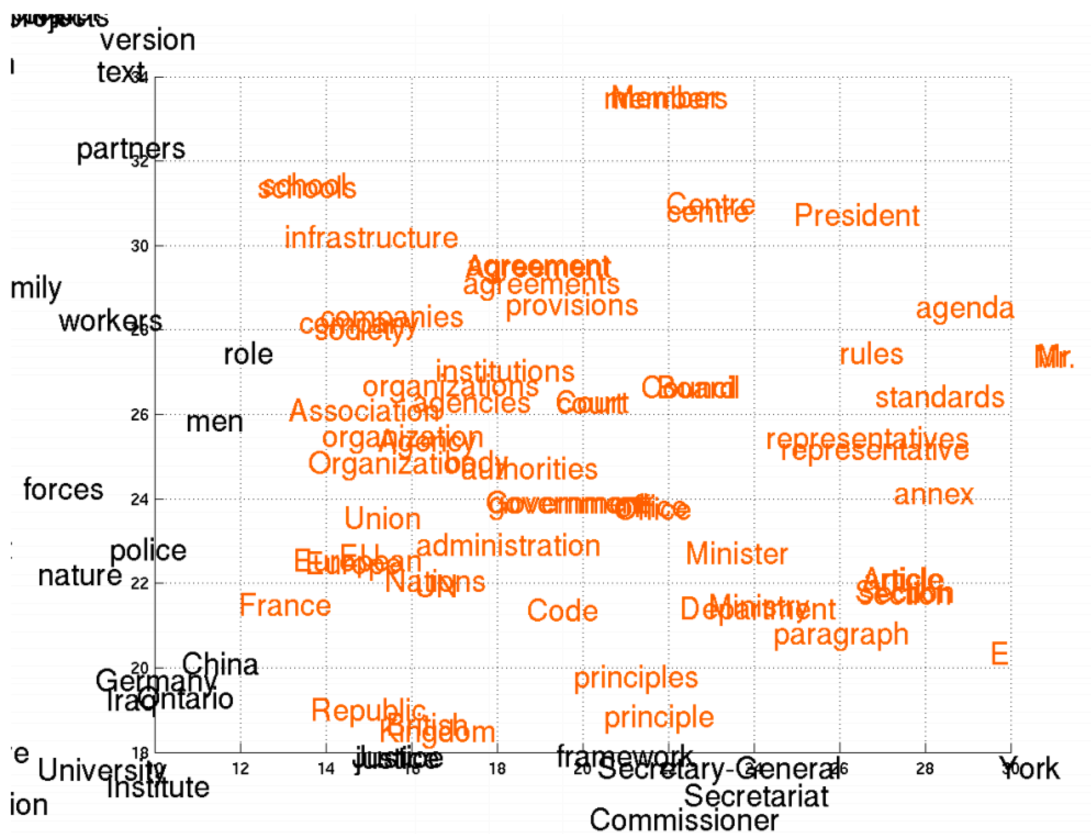


[*Bengio et al 2001*]

Distributed Word Representation

- NNLM generalises to unseen words/n-grams





[*Cho et al 2014*]

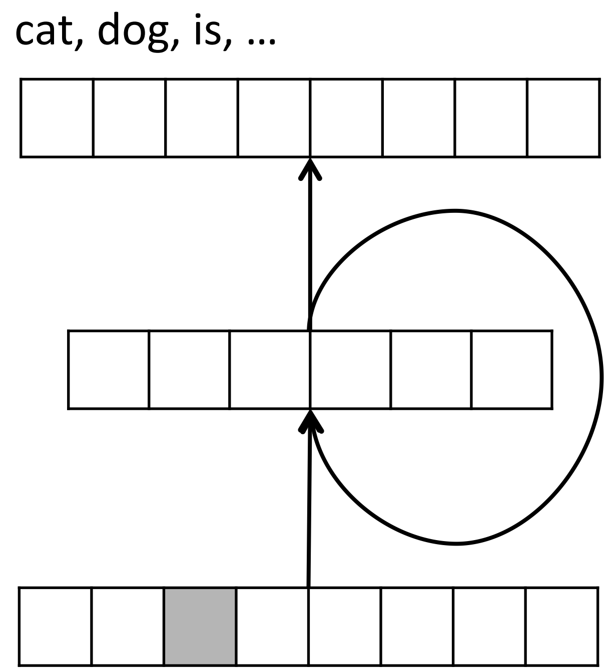
Context Modeling - RNNLM

- Non Markovian assumption
- RNNLM
  - 1-of-V encoding for each word x_t
  - Recurrent transition function

    $$\mathbf{h}_t = \tanh(\mathbf{W}^\mathrm{T}\mathbf{x}_t + \mathbf{U}^\mathrm{T}\mathbf{h}_{t-1} + \mathbf{b})$$

  - Softmax output

    $$\mathbf{p}_t = \mathrm{softmax}(\mathbf{V}^\mathrm{T}\mathbf{h}_t + \mathbf{c})$$



[*Mikolov et al 2010*]

  - Read, update, predict!
  - Can model dependency of arbitrary length

RNN Optimisation & Vanishing Gradient

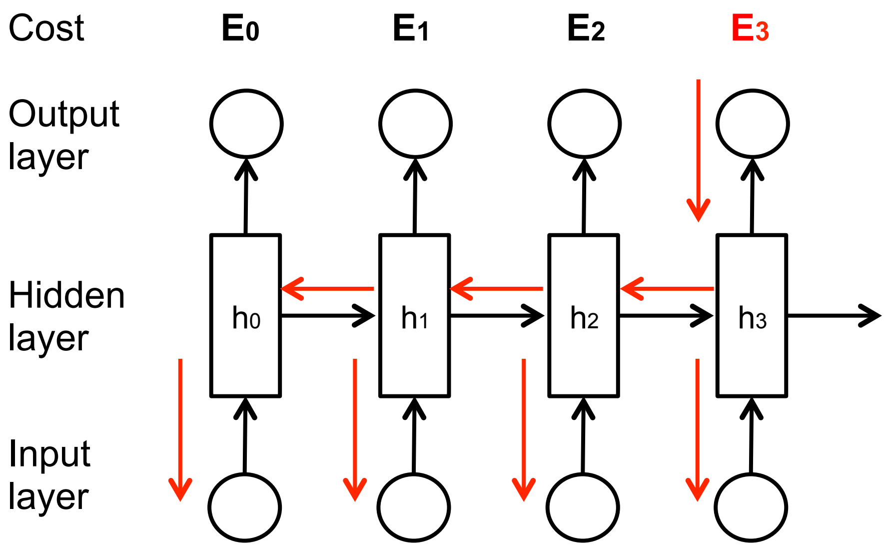
$$\mathbf{h}_t = \tanh(\mathbf{W}^{\mathrm{T}}\mathbf{x}_t + \mathbf{U}^{\mathrm{T}}\mathbf{h}_{t-1} + \mathbf{b})$$

$$\mathbf{p}_t = \mathrm{softmax}(\mathbf{V}^{\mathrm{T}}\mathbf{h}_t + \mathbf{c})$$

$$E_3 = -\mathbf{y}_3^{\mathrm{T}} \log_{10} \mathbf{p}_3$$

$$\frac{\partial E_3}{\partial \mathbf{W}} = \sum_{k=0}^{3} \frac{\partial E_3}{\partial \mathbf{p}_3} \frac{\partial \mathbf{p}_3}{\partial \mathbf{h}_3} \frac{\partial \mathbf{h}_3}{\partial \mathbf{h}_k} \frac{\partial \mathbf{h}_k}{\partial \mathbf{W}}$$

$$= \sum_{k=0}^{3} \frac{\partial E_3}{\partial \mathbf{p}_3} \frac{\partial \mathbf{p}_3}{\partial \mathbf{h}_3} \left(\prod_{j=k+1}^{3} \frac{\partial \mathbf{h}_j}{\partial \mathbf{h}_{j-1}} \right) \frac{\partial \mathbf{h}_k}{\partial \mathbf{W}}$$



$$\frac{\partial \mathbf{h}_j}{\partial \mathbf{h}_{j-1}} = \mathbf{U}^{\mathrm{T}} \cdot \mathrm{diag}(\tanh'(\mathbf{m}_j))$$

Jacobian Matrix

$$\mathbf{m}_j = \mathbf{W}^{\mathrm{T}}\mathbf{x}_j + \mathbf{U}^{\mathrm{T}}\mathbf{h}_{j-1} + \mathbf{b}$$

Ignore proof here.

$$\|\mathbf{U}\| \cdot \left\|\mathrm{diag}\left(\tanh'(\mathbf{m}_j)\right)\right\| < 1$$

Vanishing gradient !

[*Pascanu et al,2013*]

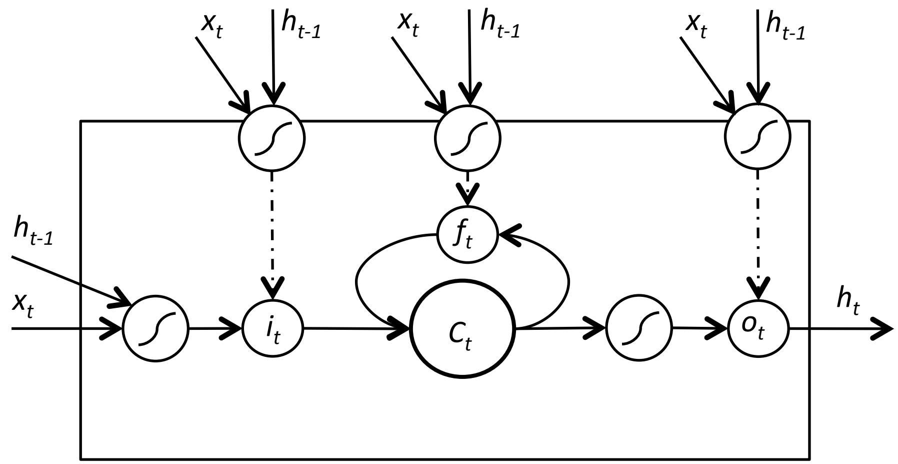
Learning Long-term Dependency - LSTM

- Sigmoid gates

$$\mathbf{i}_t = \sigma(\mathbf{W}_{wi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1})$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{wf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1})$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{wo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1})$$



[*Hochreiter and Schmidhuber*, 1997]

- Proposed cell value

$$\hat{\mathbf{c}}_t = \tanh(\mathbf{W}_{wc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1})$$

- Update cell and hidden layer

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \hat{\mathbf{c}}_t$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t)$$

Learning Long-term Dependency - LSTM

- How does it prevent vanishing gradient?
  - Consider memory cell update

    $$\mathbf{C}_t = \mathbf{i}_t \odot \hat{\mathbf{C}}_t + \mathbf{f}_t \odot \mathbf{C}_{t-1}$$

  - We can back-prop the gradient by chain rule

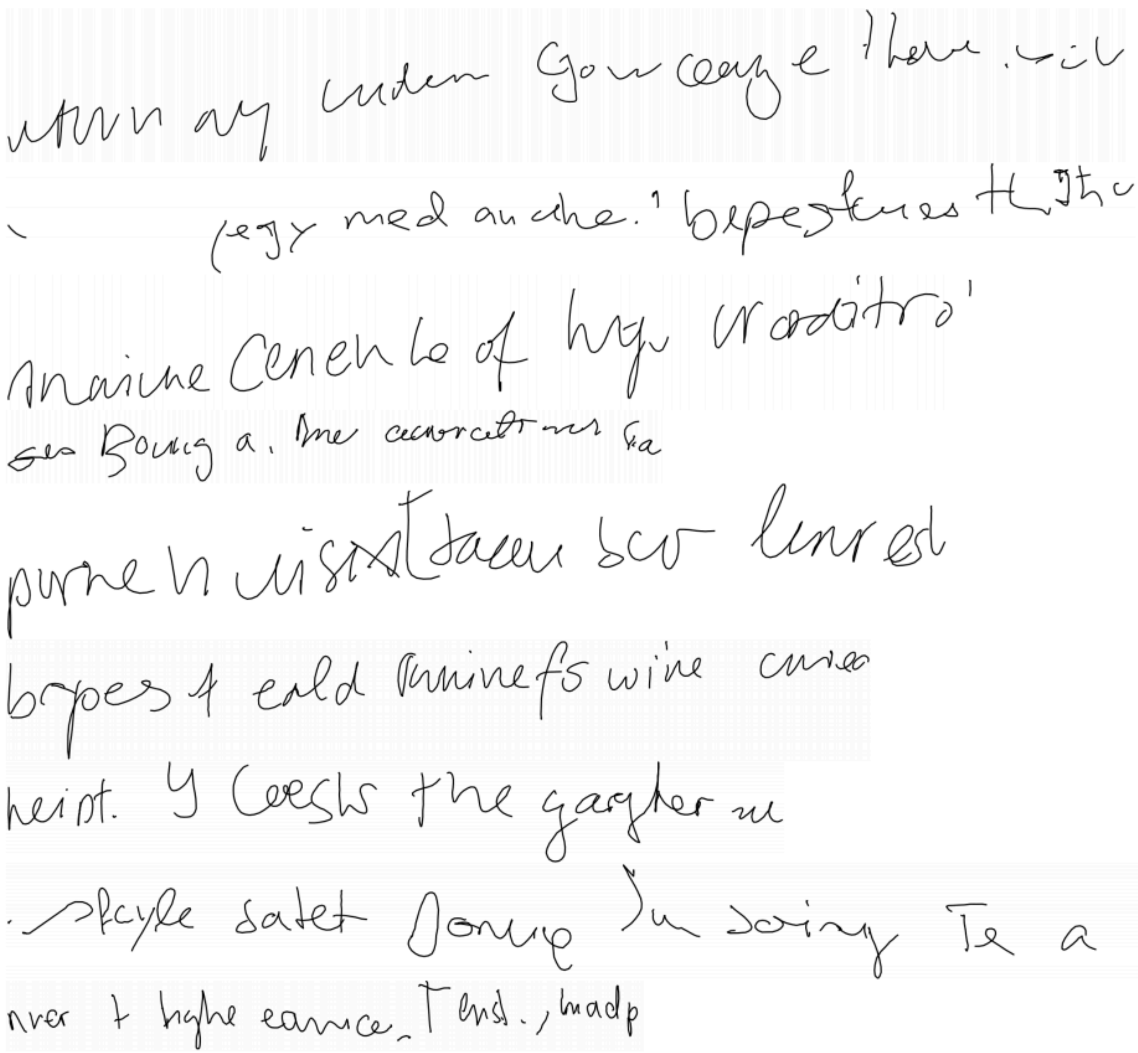
    $$\frac{\partial E_t}{\partial C_{t-1}} = \frac{\partial E_t}{\partial C_t}\frac{\partial C_t}{\partial C_{t-1}} = \frac{\partial E_t}{\partial C_t} f_t$$

  - If f_t maintains a value of 1, gradient is perfectly propagated.

RNNLM Text Generation [*Sutskever et al 2011*]

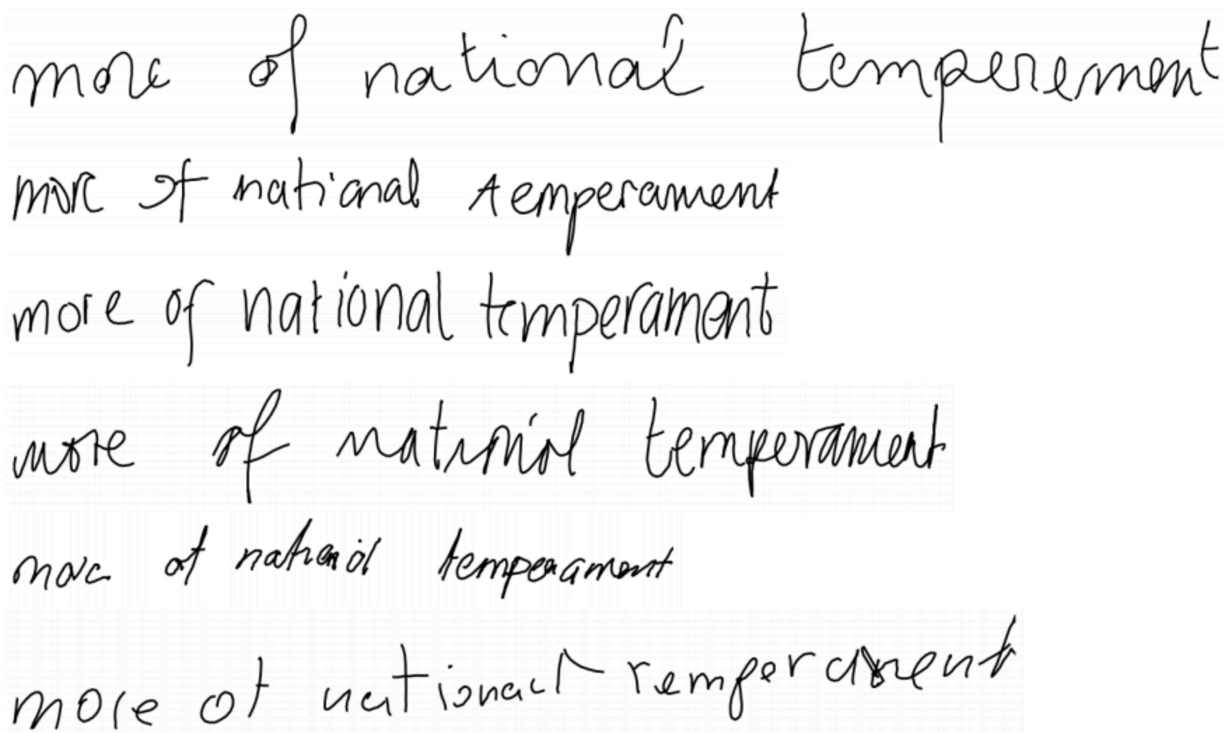
- **The meaning of life is …**
- *The meaning of life is the tradition of the ancient* ***human reproduction****: it is less favorable to the good boy for when to remove her bigger. In the show's agreement unanimously resurfaced. The wild pasteured with consistent street forests were incorporated by 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*]



RNN handwriting synthesis [*Graves, 2013*]

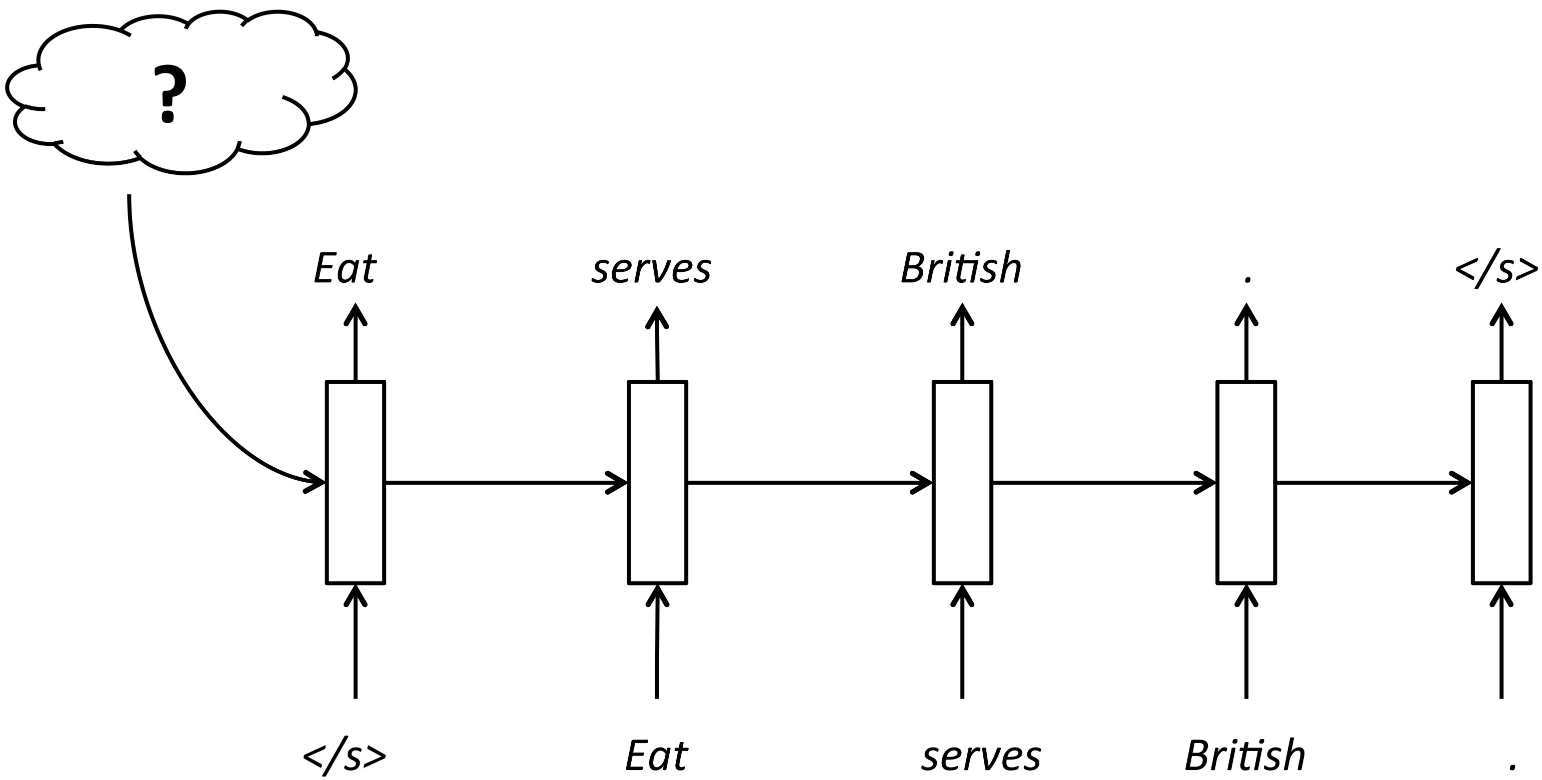
- Can we gain control on generated content?



Q & A

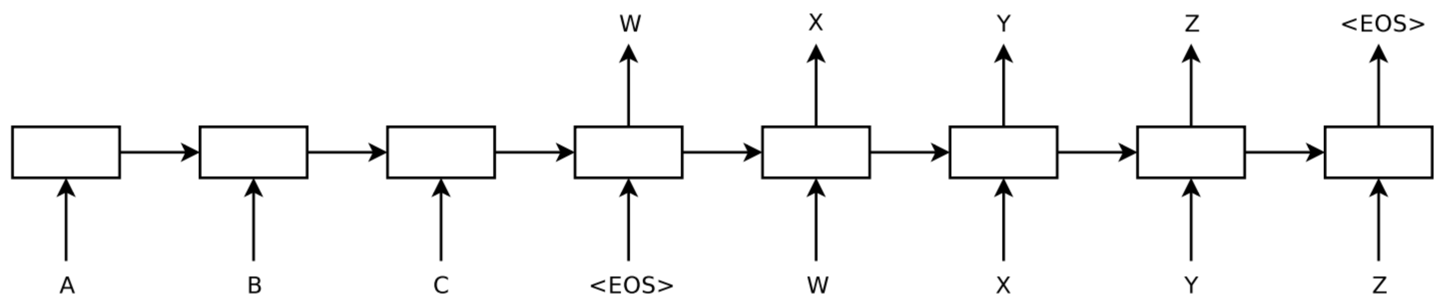
The 3rd Reason: Control!

Integrating across modalities – Conditional RNN

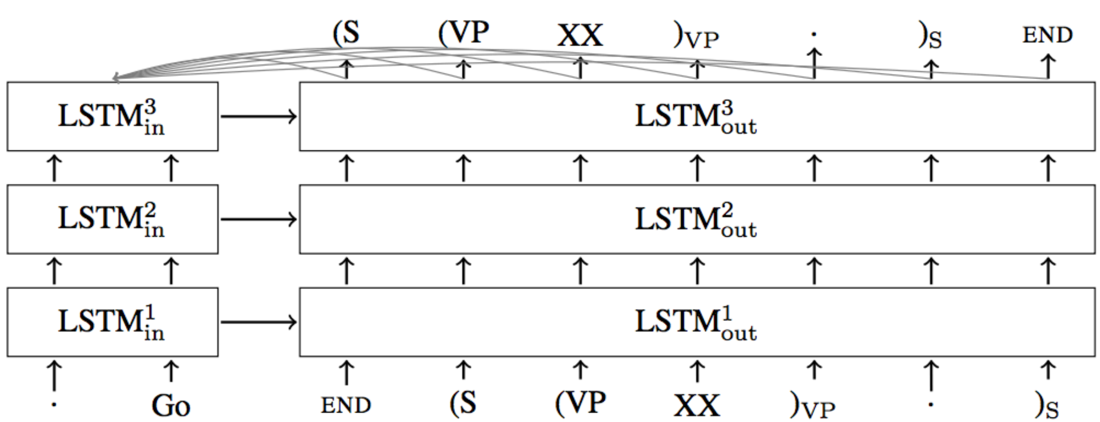


Integrating across modalities – Conditional RNN

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

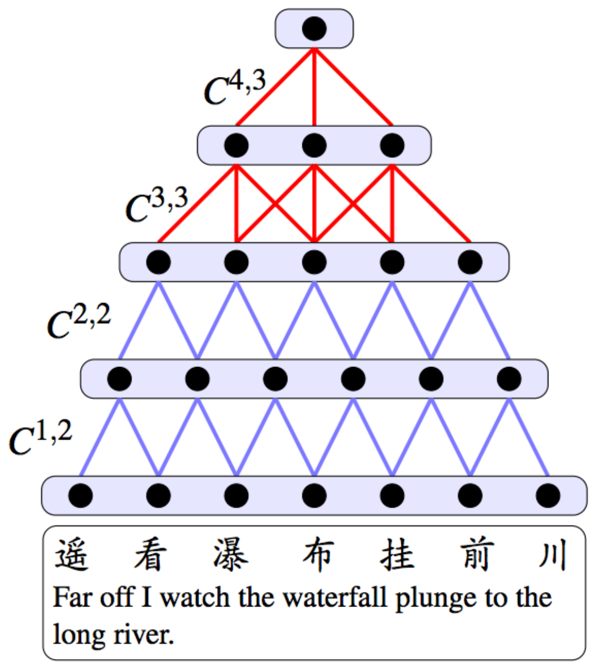


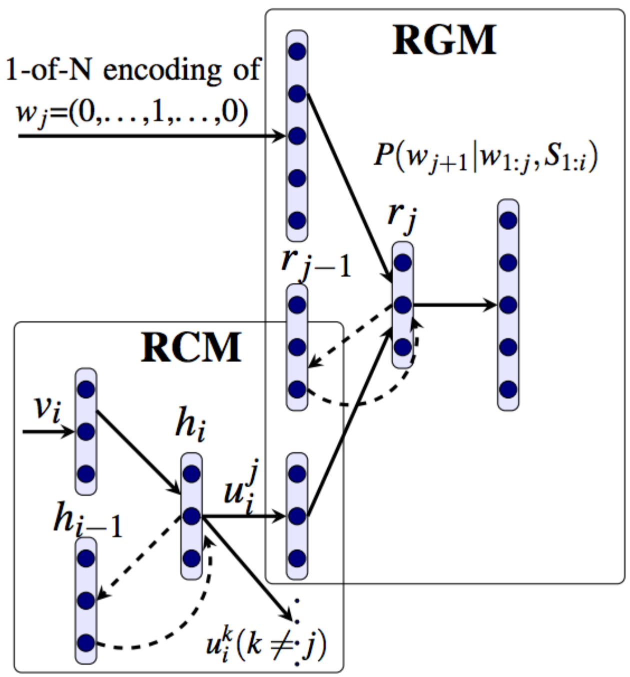
  - Grammar as a foreign language [*Vinyals et al, 2015*]



Integrating across modalities – Conditional RNN

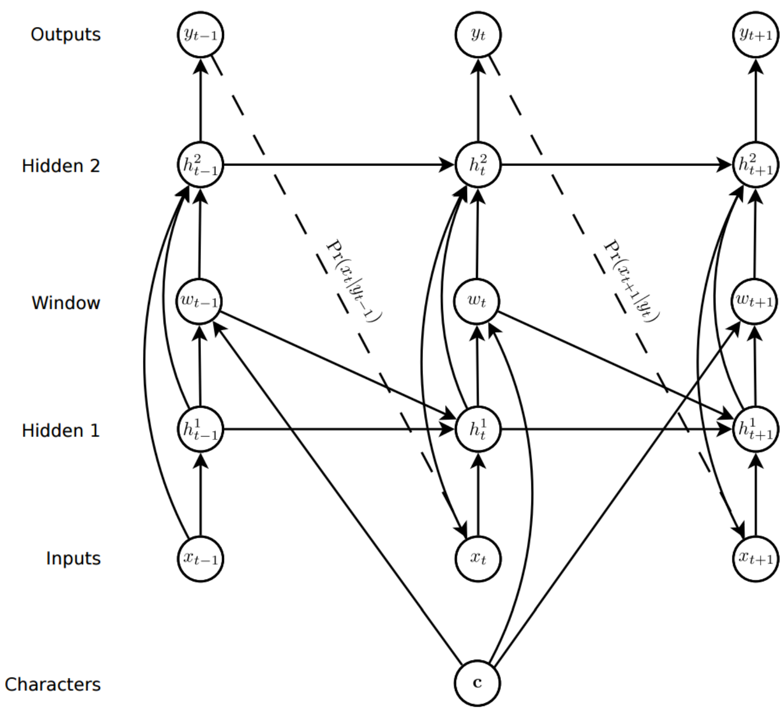
- Text-to-Text
  - Chinese Poetry Generation [Zhang *and Lapata, 2014*]

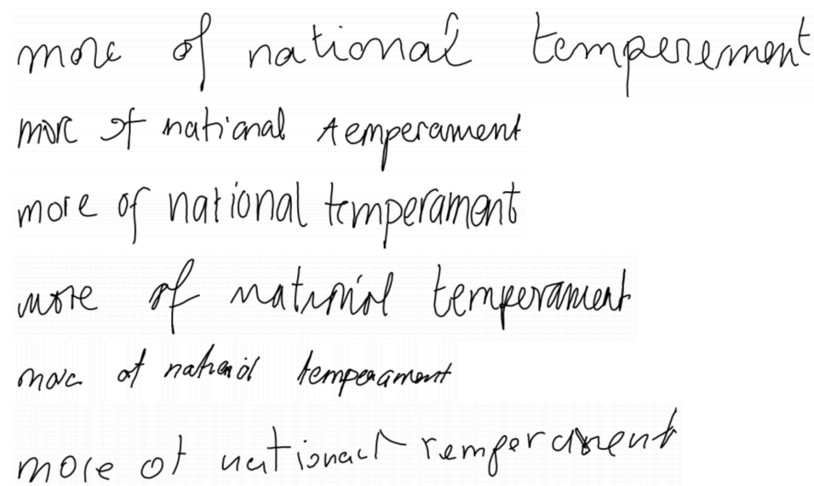




Integrating across modalities – Conditional RNN

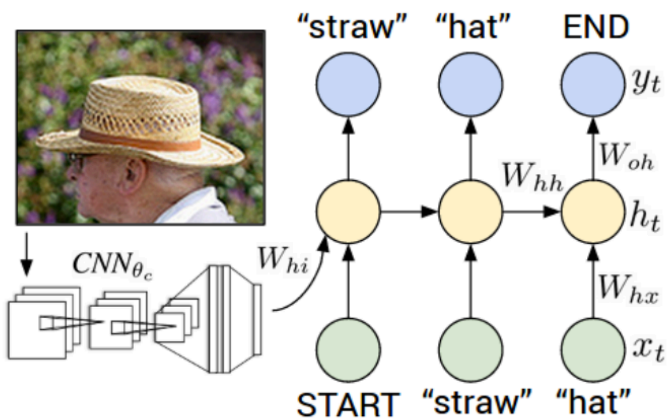
- Text-to-Image [*Graves, 2013*]





Integrating across modalities – Conditional RNN

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




man in black shirt is playing guitar.


construction worker in orange safety vest is working on road.


two young girls are playing with lego toy.


boy is doing backflip on wakeboard.

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

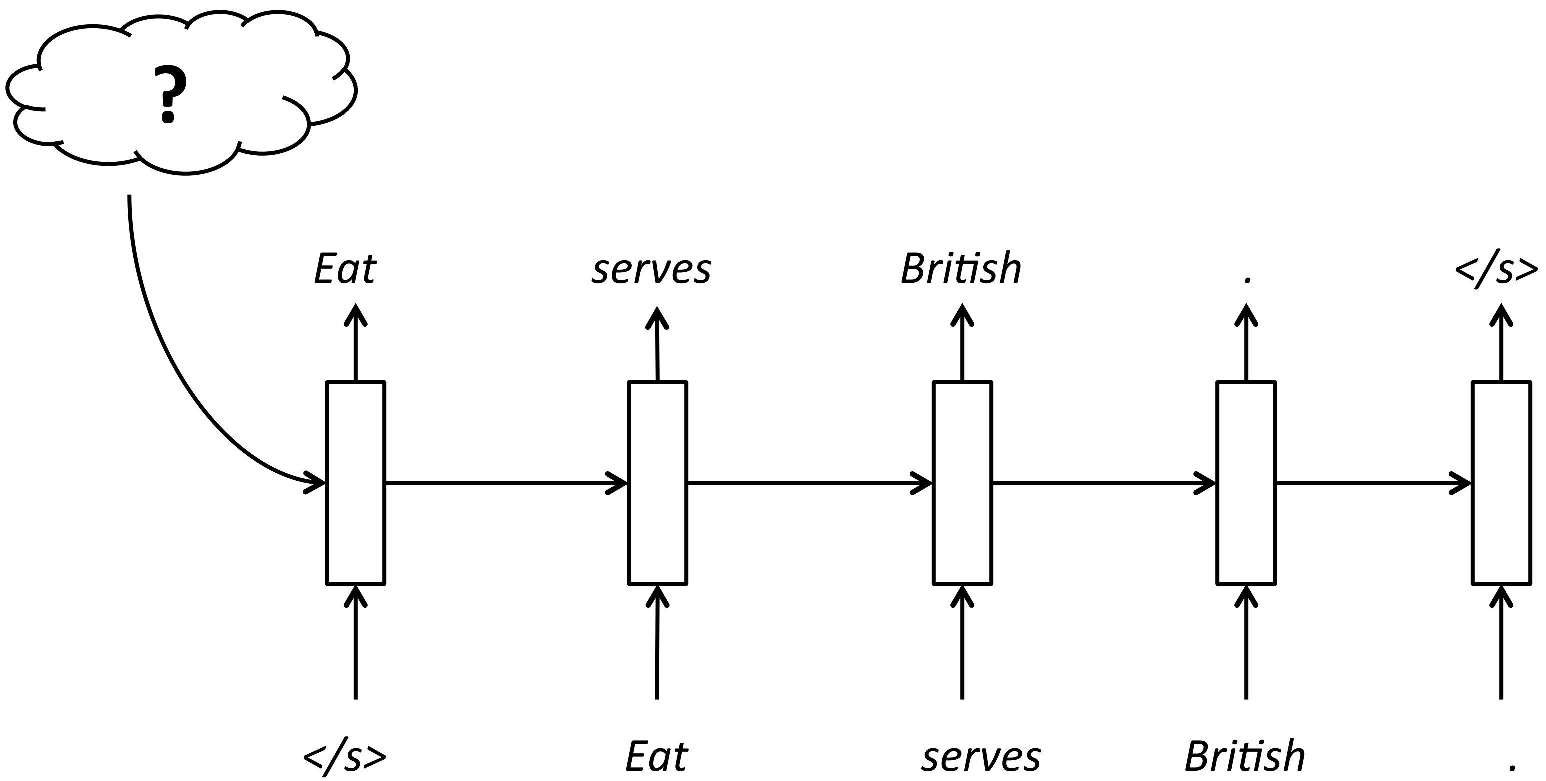
Q & A

Part II: NLG models

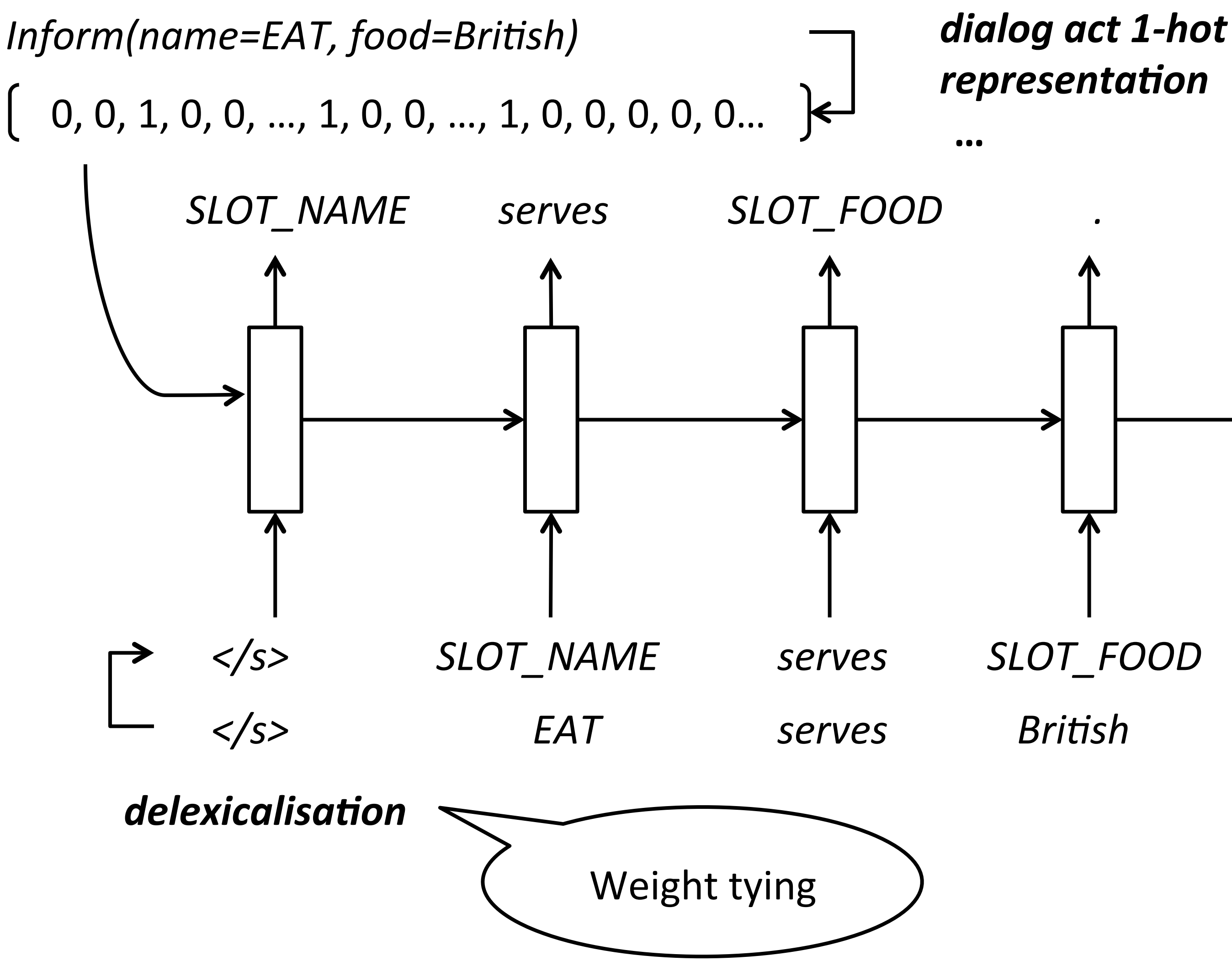
- Gated-based NLG models
- Attention-based NLG models
- Domain Adaptation
- Deep NLG for Dialogue Response Generation

Conditional RNNLM

- Generation conditions on MR
  - Represent MR?



RNN Language Generator



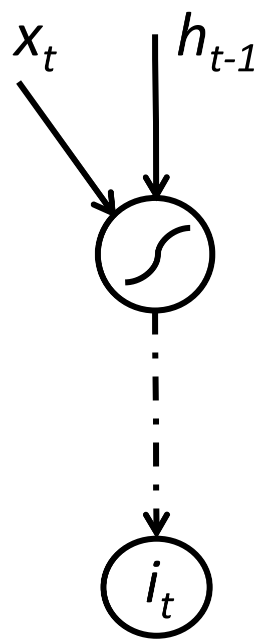
(Wen et al, 2015a)

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*]

Learning to Control Gates [*Wen et al, 2015b*]

- Recap LSTM gates:
  - $\mathbf{i}_t = \sigma(\mathbf{W}_{wi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1})$
  - 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).

SC-LSTM [*Wen et al, 2015b*]

- Original LSTM cell

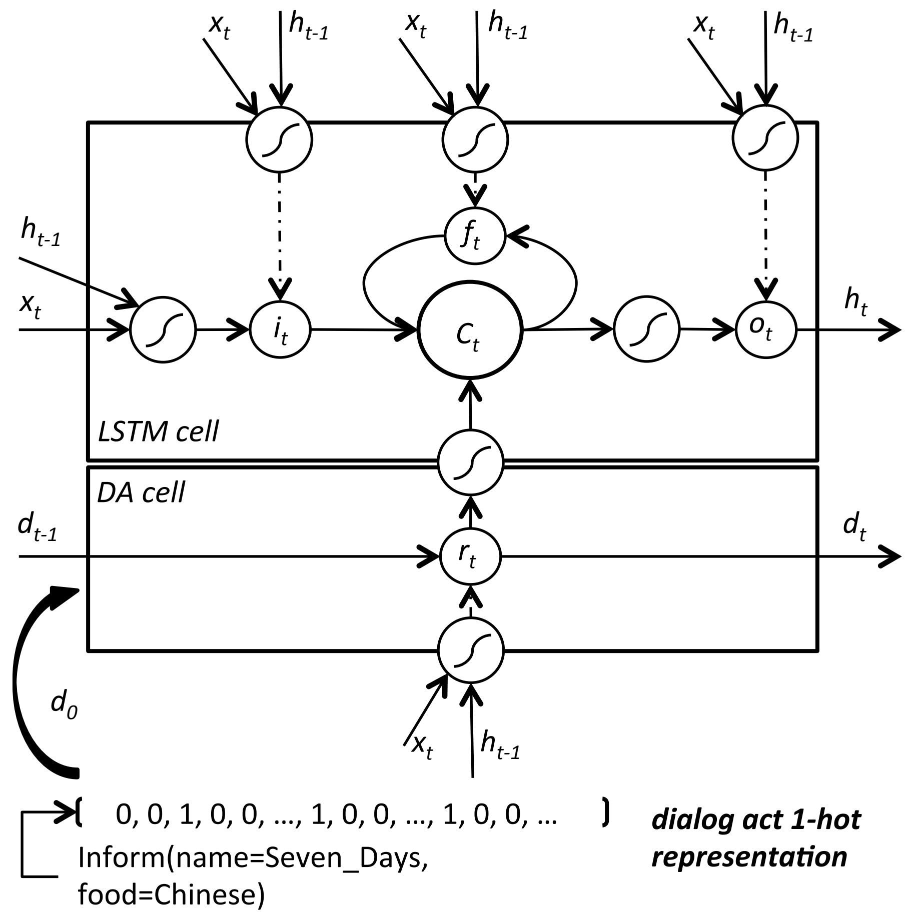
$$\mathbf{i}_t = \sigma(\mathbf{W}_{wi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1})$$
$$\mathbf{f}_t = \sigma(\mathbf{W}_{wf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1})$$
$$\mathbf{o}_t = \sigma(\mathbf{W}_{wo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1})$$
$$\hat{\mathbf{c}}_t = \tanh(\mathbf{W}_{wc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1})$$
$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \hat{\mathbf{c}}_t$$
$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t)$$

- DA cell

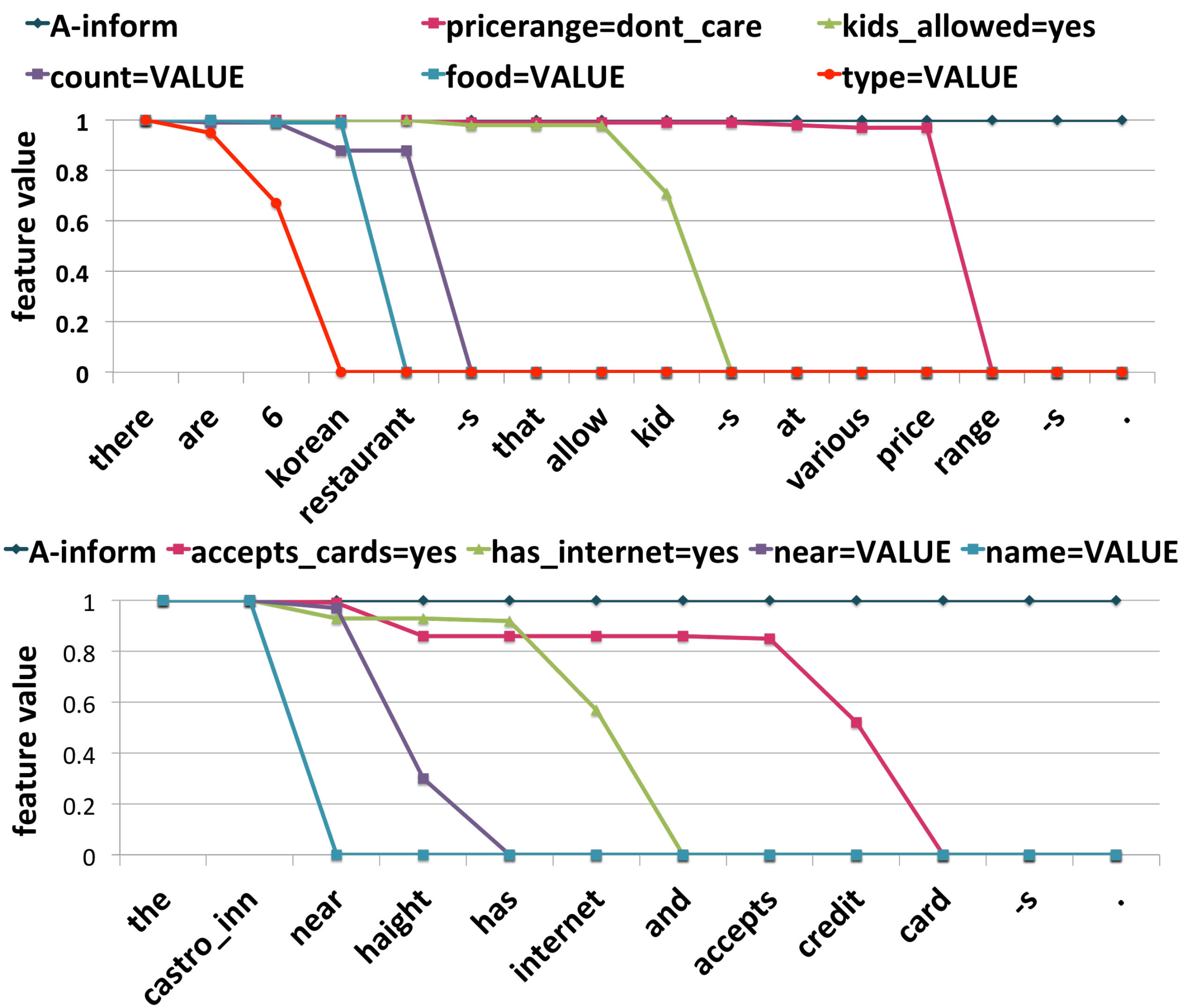
$$\mathbf{r}_t = \sigma(\mathbf{W}_{wr}\mathbf{x}_t + \mathbf{W}_{hr}\mathbf{h}_{t-1})$$
$$\mathbf{d}_t = \mathbf{r}_t \odot \mathbf{d}_{t-1}$$

- Modify Ct

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \hat{\mathbf{c}}_t + \tanh(\mathbf{W}_{dc}\mathbf{d}_t)$$



Visualization [*Wen et al, 2015b*]

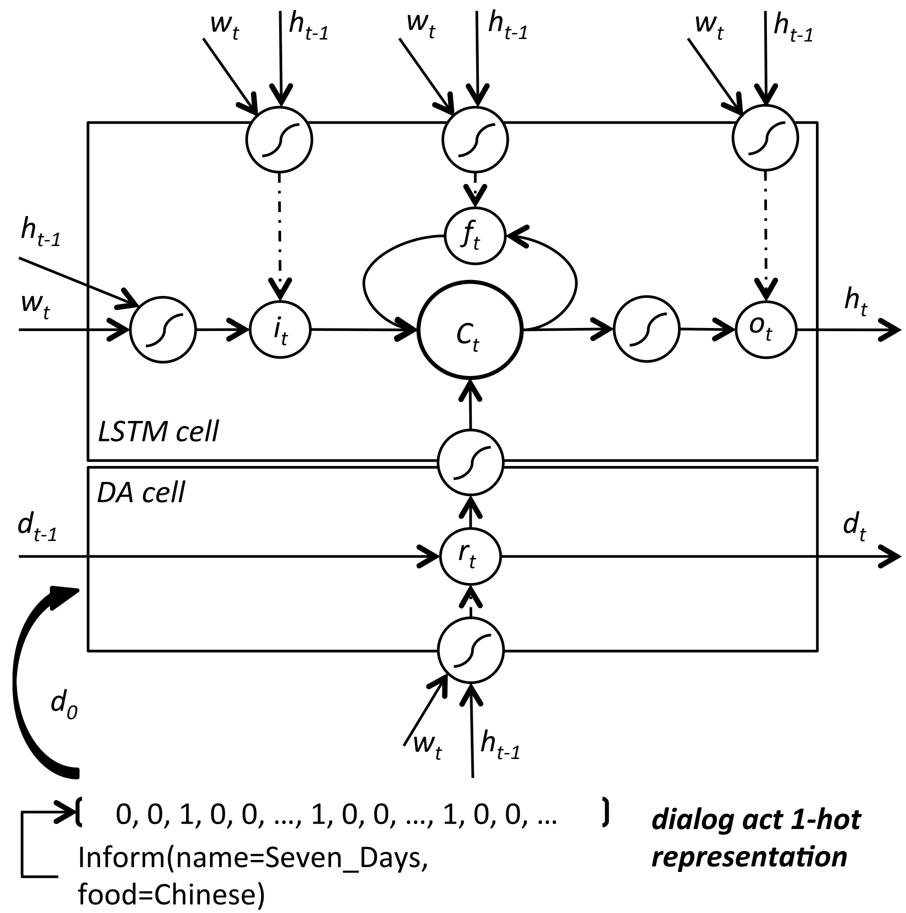


Cost function [*Wen et al, 2015b*]

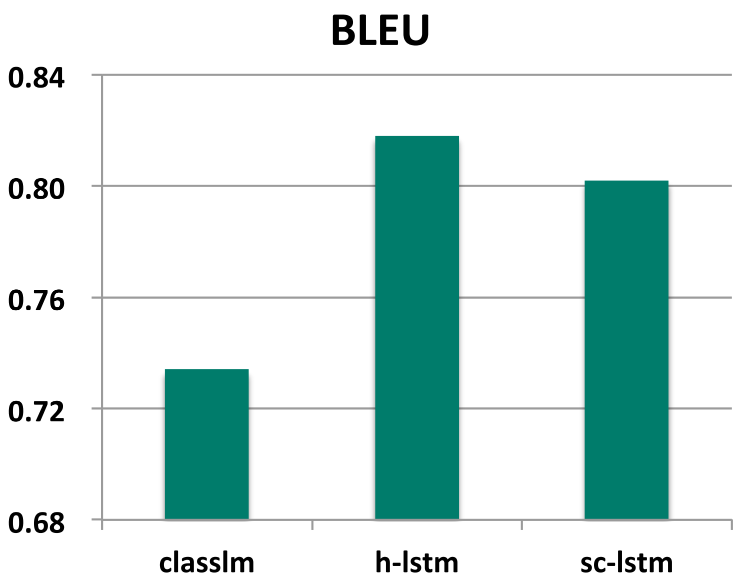
- Cost function

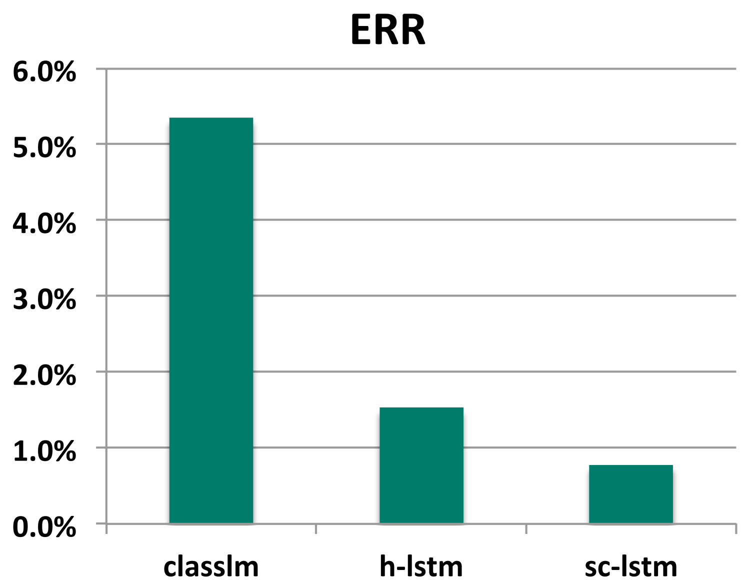
$$\mathcal{L}(\theta) = -\sum_t \mathbf{y}_t^{\mathrm{T}} \log \mathbf{p}_t + \|\mathbf{d}_T\| + \sum_{t=0}^{T-1} \eta \xi^{\|\mathbf{d}_{t+1} - \mathbf{d}_t\|}$$

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



Results [*Wen et al, 2015b*]



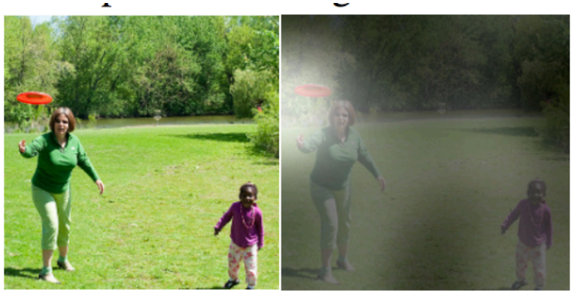


Method	**Informativeness**	**Naturalness**
sc-lstm	2.59	2.50
h-lstm	2.53	2.42^{*}
classlm	2.46^{**}	2.45

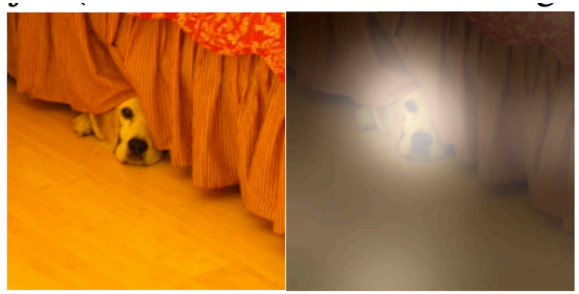
$^{*}p < 0.05$ $^{**}p < 0.005$

Attention Mechanism?

Attentive Caption Generation [*Xu et al, 2015*]

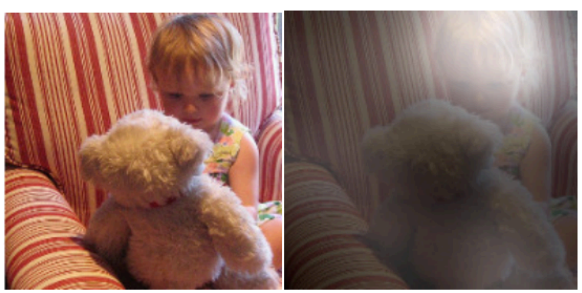


A woman is throwing a frisbee in a park.

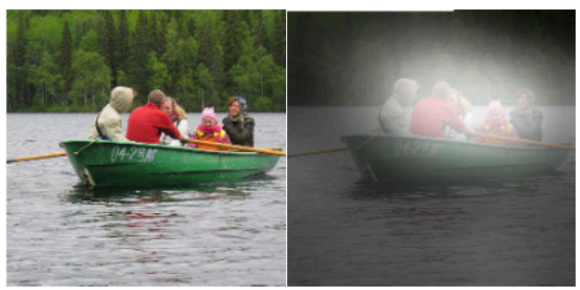


A dog is standing on a hardwood floor.

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|θ)** based on network parameter θ and take the **expectation** over sources: $\mathbf{g} = \sum_{\mathbf{s}} p(\mathbf{s}|\theta)\,\mathbf{s}$

- 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 hj by

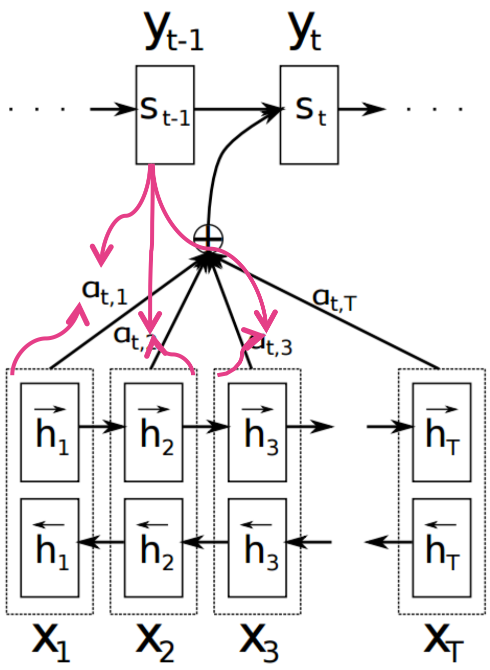
    $$e_{tj} = \mathbf{v}^{\mathrm{T}} \tanh(\mathbf{W} \cdot \mathbf{s}_{t-1} + \mathbf{U} \cdot \mathbf{h}_j)$$

    $$\alpha_{tj} = \mathrm{softmax}(e_{tj})$$

  - Take an expectation over sources

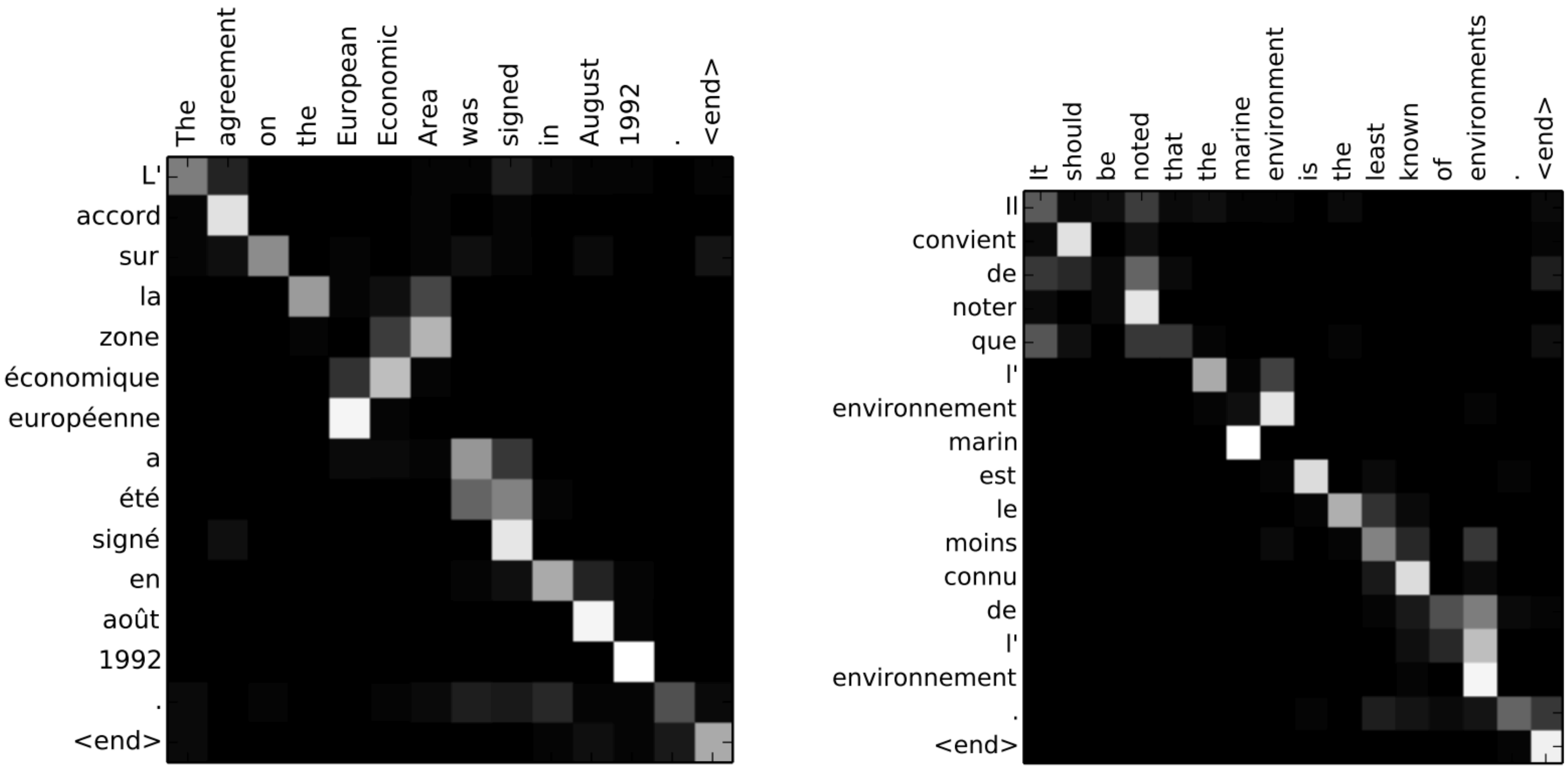
    $$\mathbf{c}_t = \sum_j \alpha_{tj}\, \mathbf{h}_j$$

[*Bahdanau et al,2013*]



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

Neural MT [*Bahdanau et al,2013*]



Attentive Encoder-Decoder for NLG

- Slot & value embedding

$$\mathbf{z}_i = \mathbf{s}_i + \mathbf{v}_i$$

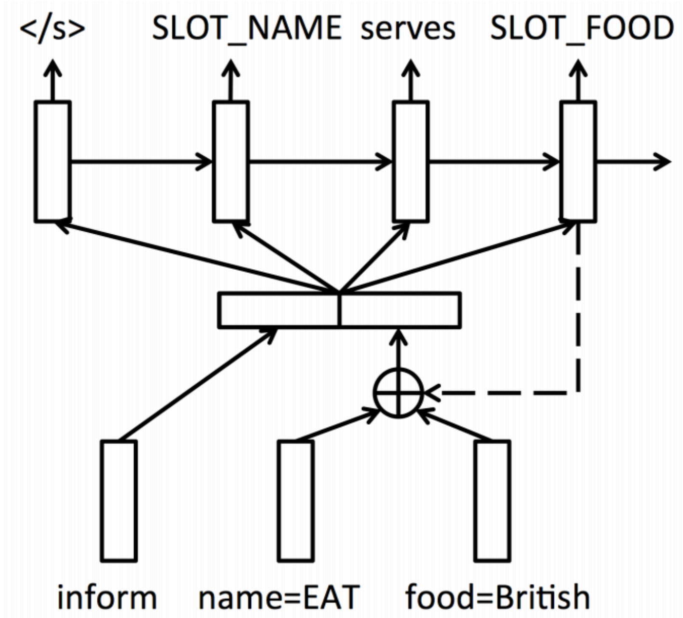
- Attentive MR representation

$$e_{ti} = \mathbf{v}^{\mathrm{T}} \tanh(\mathbf{W}_{hm}\mathbf{h}_{t-1} + \mathbf{W}_{zm}\mathbf{z}_i)$$

$$\alpha_{ti} = \mathrm{softmax}(e_{ti})$$

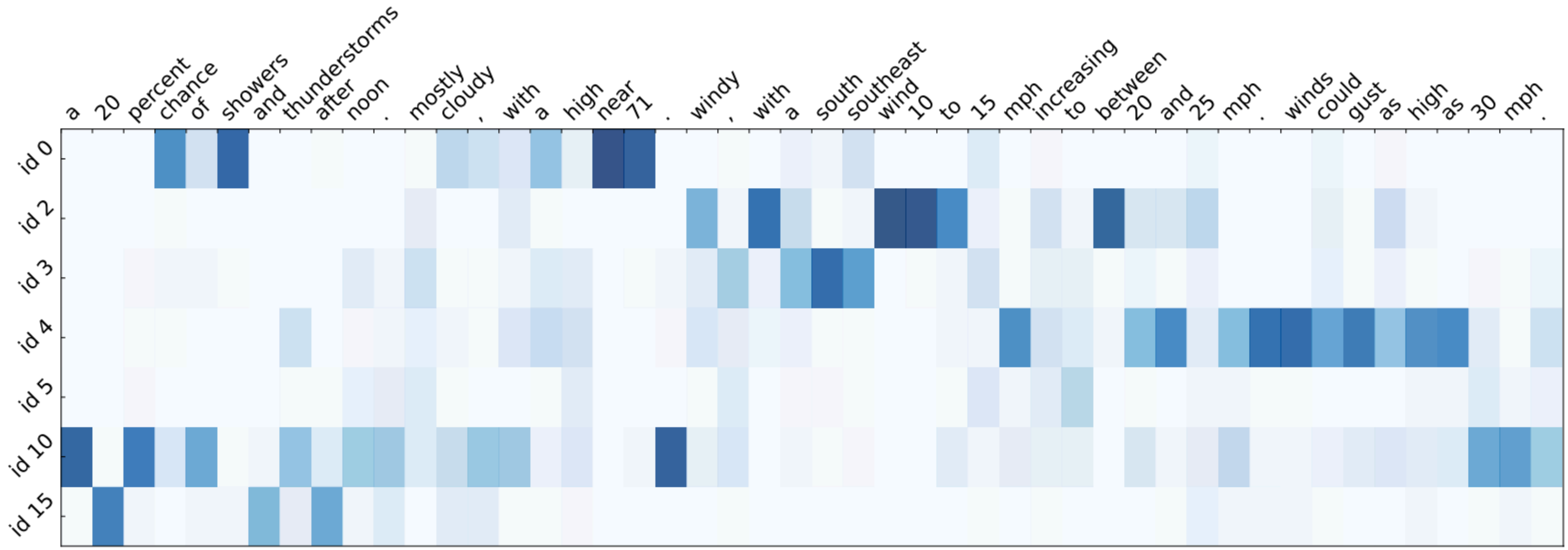
$$\mathbf{d}_t = \mathbf{a} \oplus \sum_i \boldsymbol{\alpha}_{ti}\mathbf{z}_i$$

[*Wen et al,2015c*]



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

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);

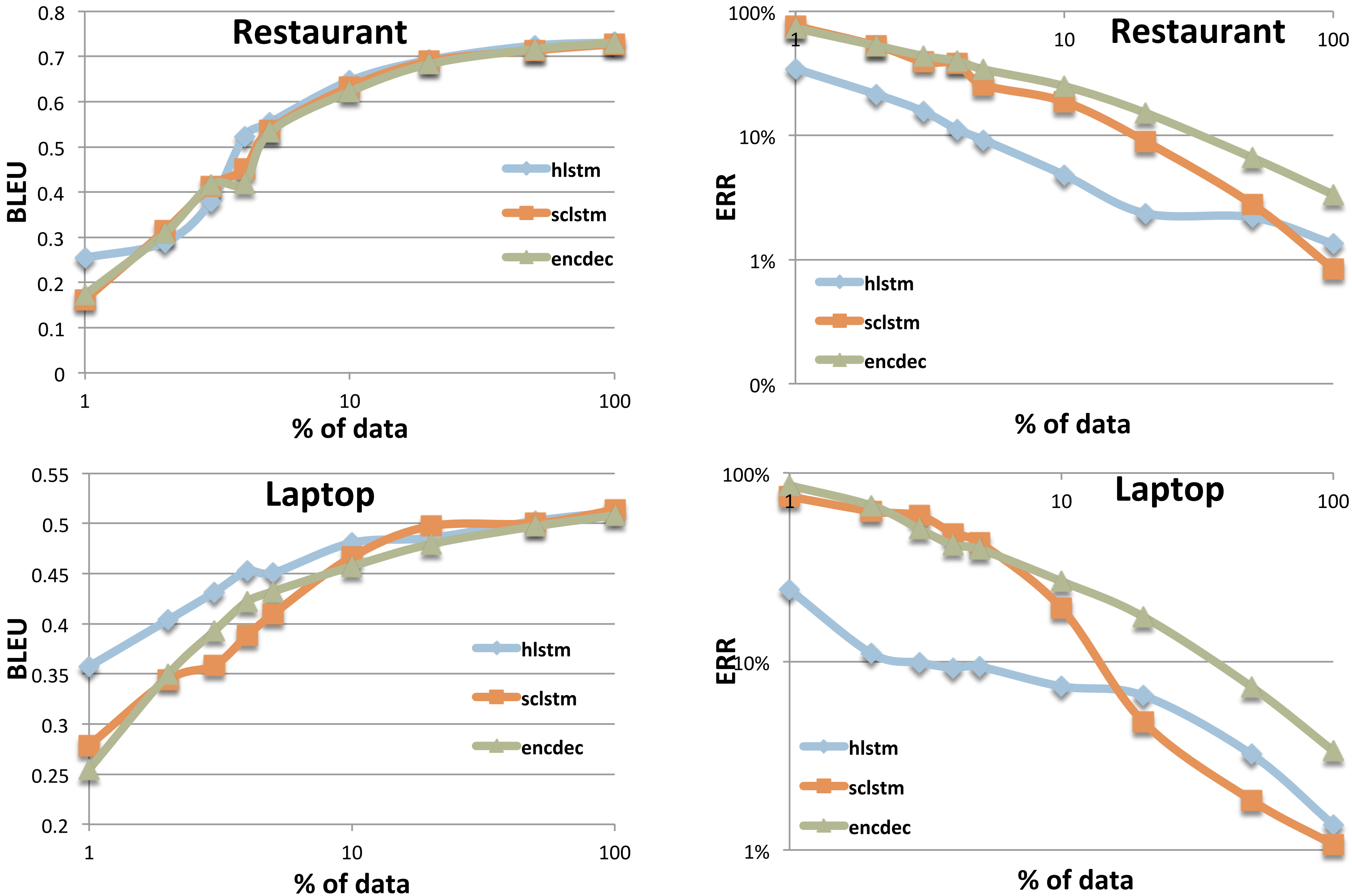
id-3: windDir(time=06-21, mode=SSE); id-4: gust(time=06-21, min=0, mean=10, max=30);

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=SChc)

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

Model Comparison

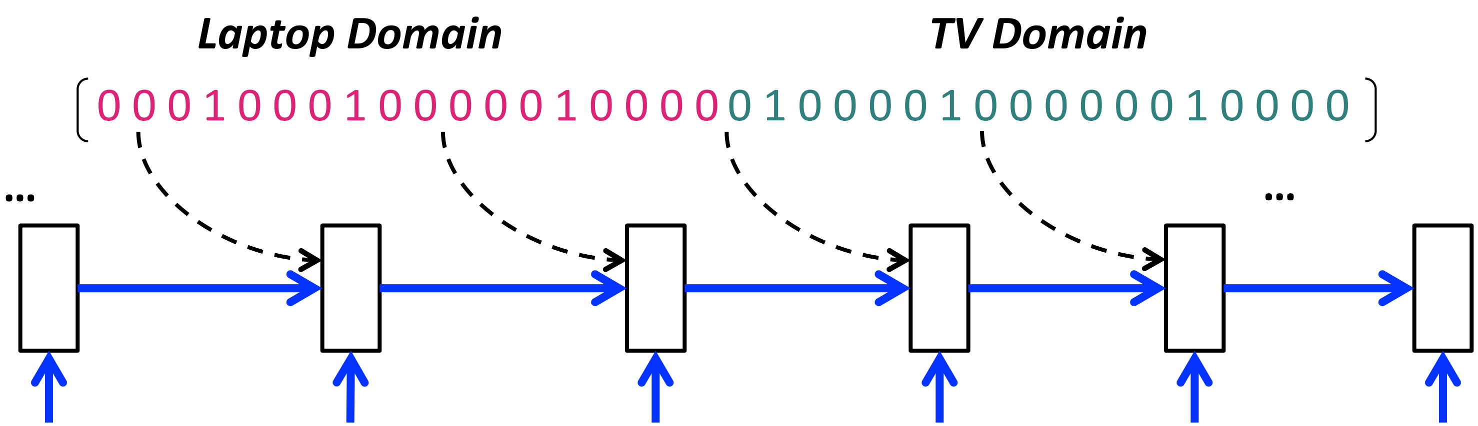


Q & A

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



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> <MEMORY-slot>

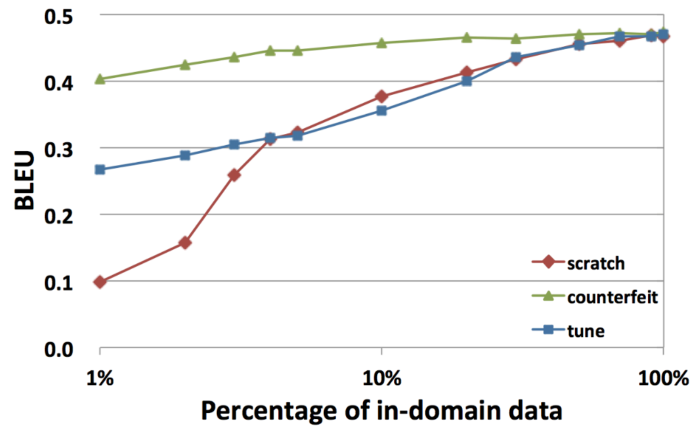
counterfeiting

<NAME-value> is a <FAMILY-value> <TYPE-value> with a <SCREEN-value> <SCREEN-slot>

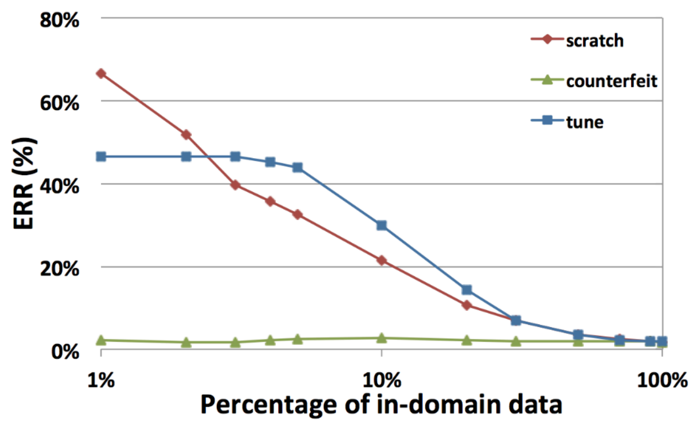
A possible realisation in TV (target) domain:

Apollo 73 is a U76 television with a 29-inch screen

Data counterfeiting – Results [Wen et al, 2016a]



(a) BLEU score curve

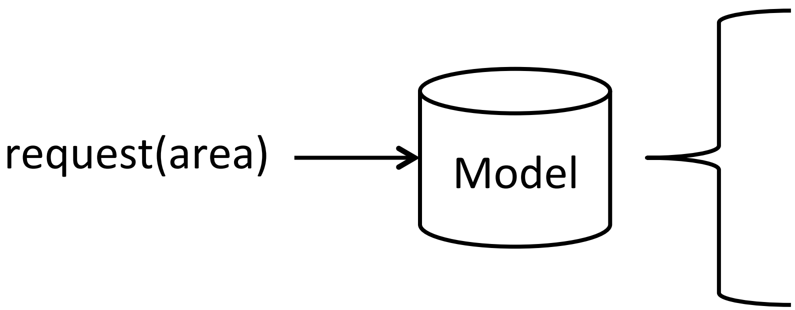


(b) Slot error rate curve

Laptop to TV

Discriminative Training [Wen et al, 2016a]

- Explore model capacity and correct it.



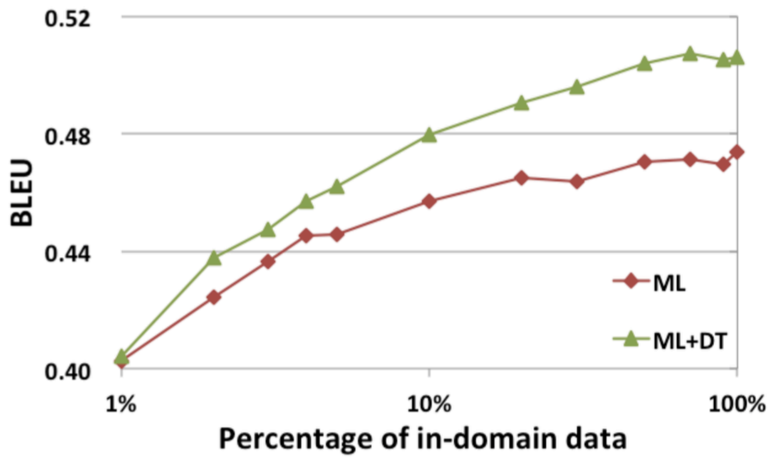
candidates	***score***
What area do you want?	0.9
What food type do you want?	0.2
Do you want north area?	0.1
Do you have any area in mind?	0.8
What part of town do you want?	1.0

- DT cost function:

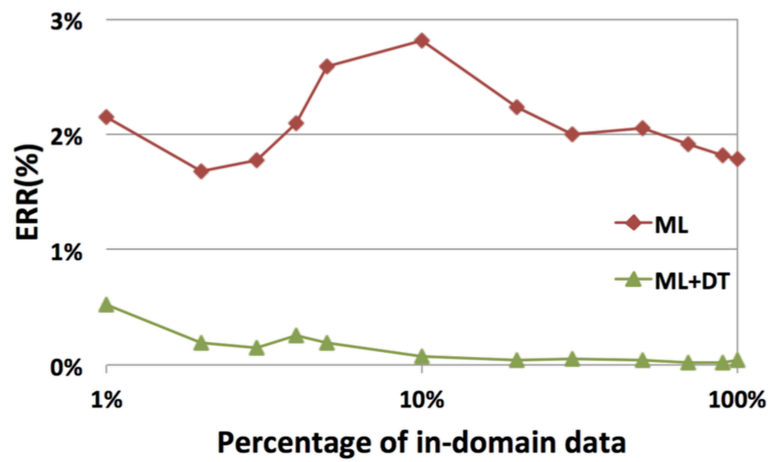
$$F(\theta) = -\mathbb{E}[L(\theta)]$$
$$= -\sum_{\Omega \in Gen(d_i)} p_\theta(\Omega|d_i) L(\Omega, \Omega_i)$$

Ω: candidate sentence
Ωi: reference sentence
di: dialogue act
L(.): scoring function

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



(a) Effect of DT on BLEU



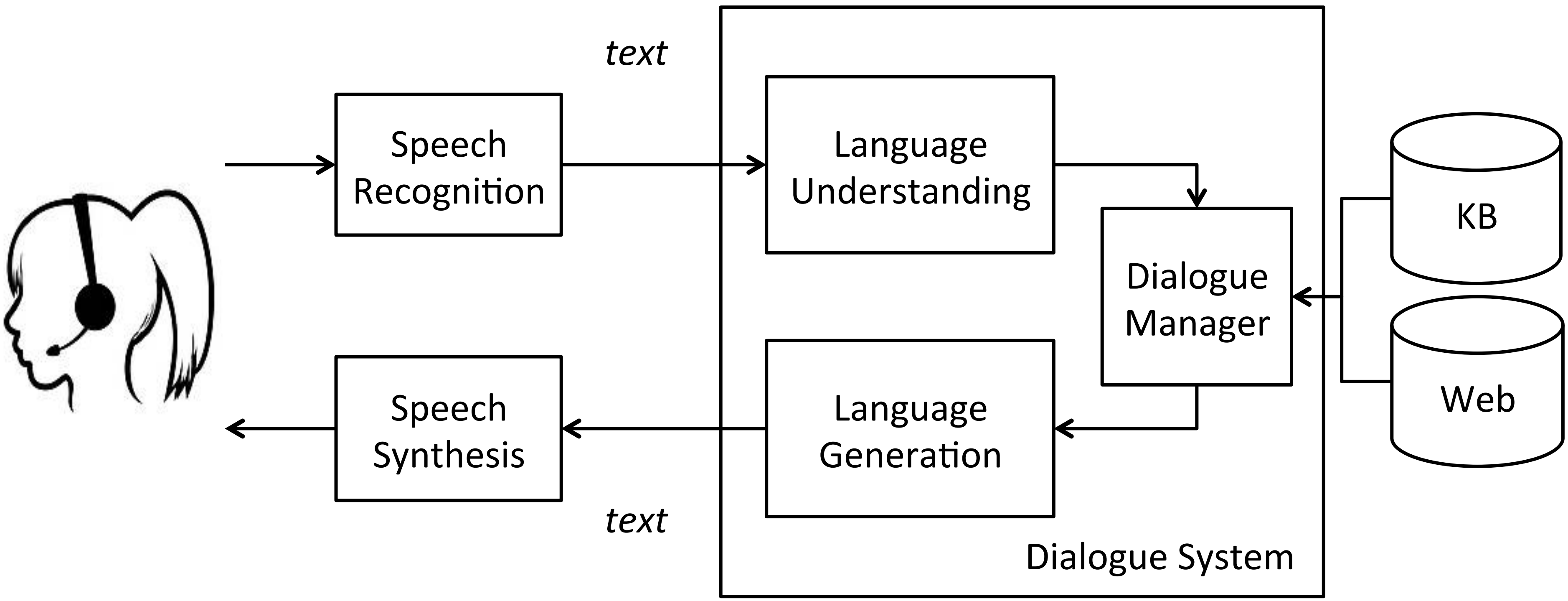
(b) Effect of DT on slot error rate

Q & A

Deep NLG for Dialogue Response Generation

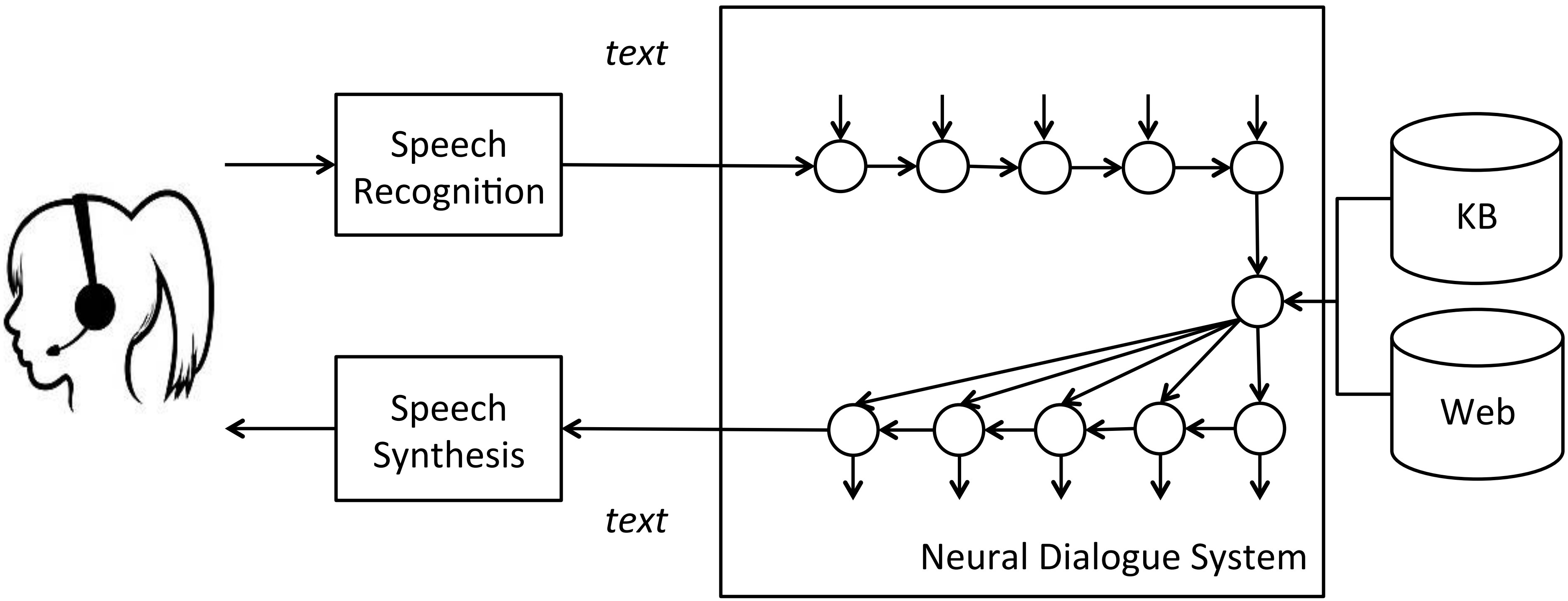
Traditional Dialogue Systems

- Step-by-step query processing



Neural Dialogue Systems

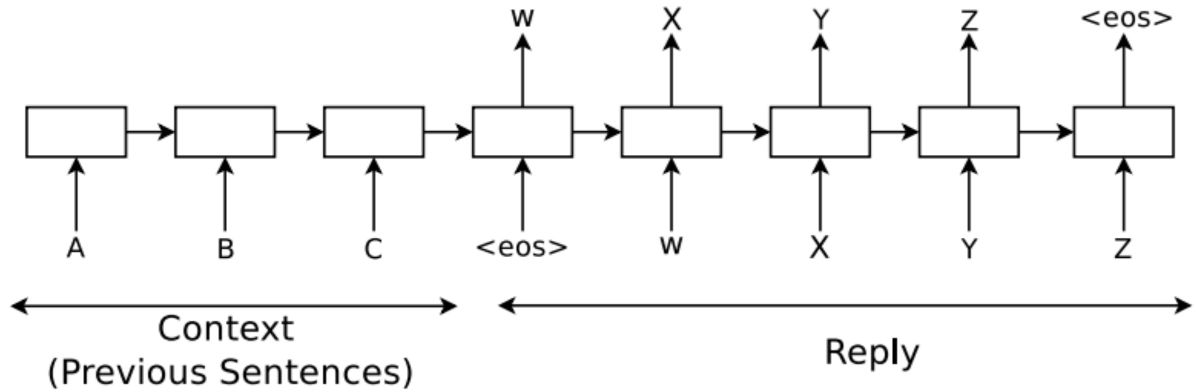
- How about integrating the whole thing?



A Neural Conversational Model

- Seq2Seq

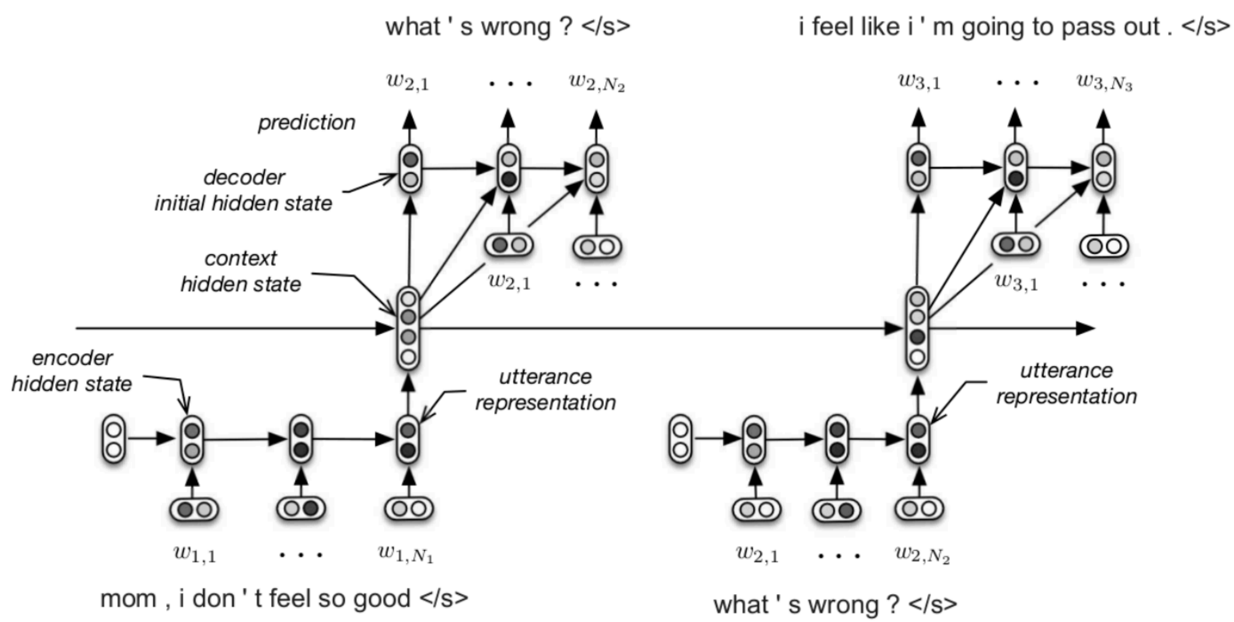
[*Vinyals and Le, 2015*]



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 ($\mathbf{U_1}$, $\mathbf{U_2}$)	**MAP**	**Target** ($\mathbf{U_3}$)
U_1: yeah , okay . U_2: well , i guess i ' ll be going now .	i ' ll see you tomorrow .	yeah .
U_1: oh . <continued_utterance> oh . U_2: what ' s the matter , honey ?	i don ' t know .	oh .
U_1: it ' s the cheapest . U_2: then it ' s the worst kind ?	no , it ' s not .	they ' re all good , sir .
U_1: <person> ! what are you doing ? U_2: 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 .

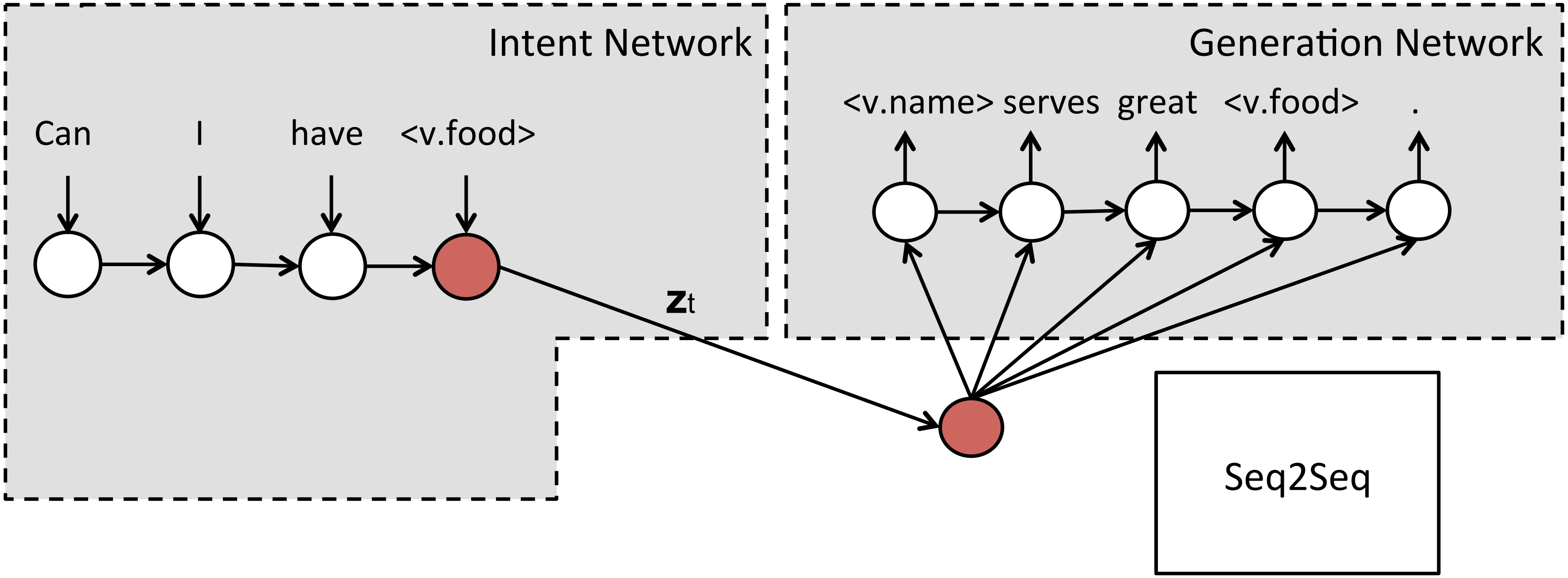
A Network-based End-to-End Trainable Task-Oriented Dialogue System, Wen et al, 2016b

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

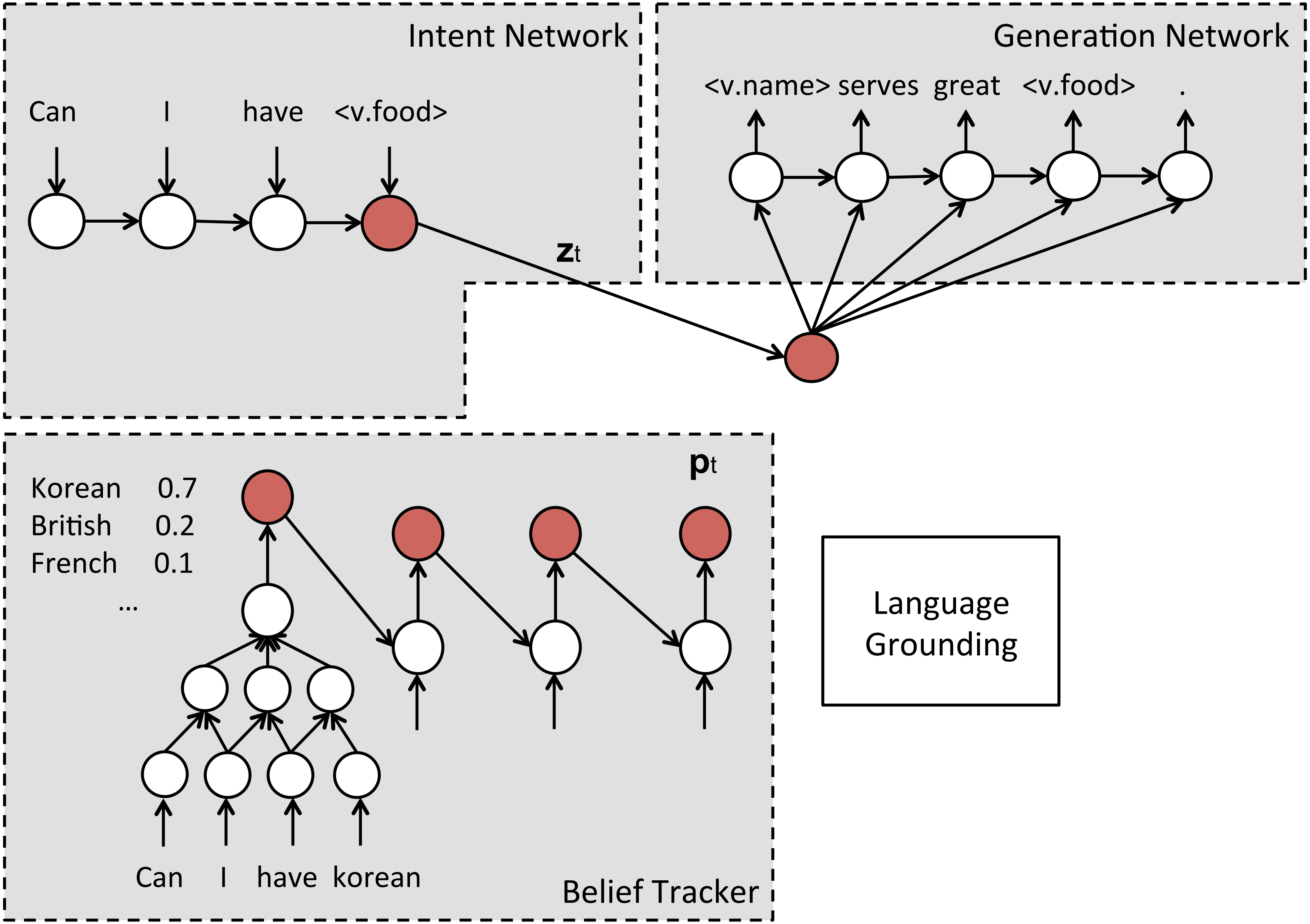
Can I have <v.food>

Delexicalisation

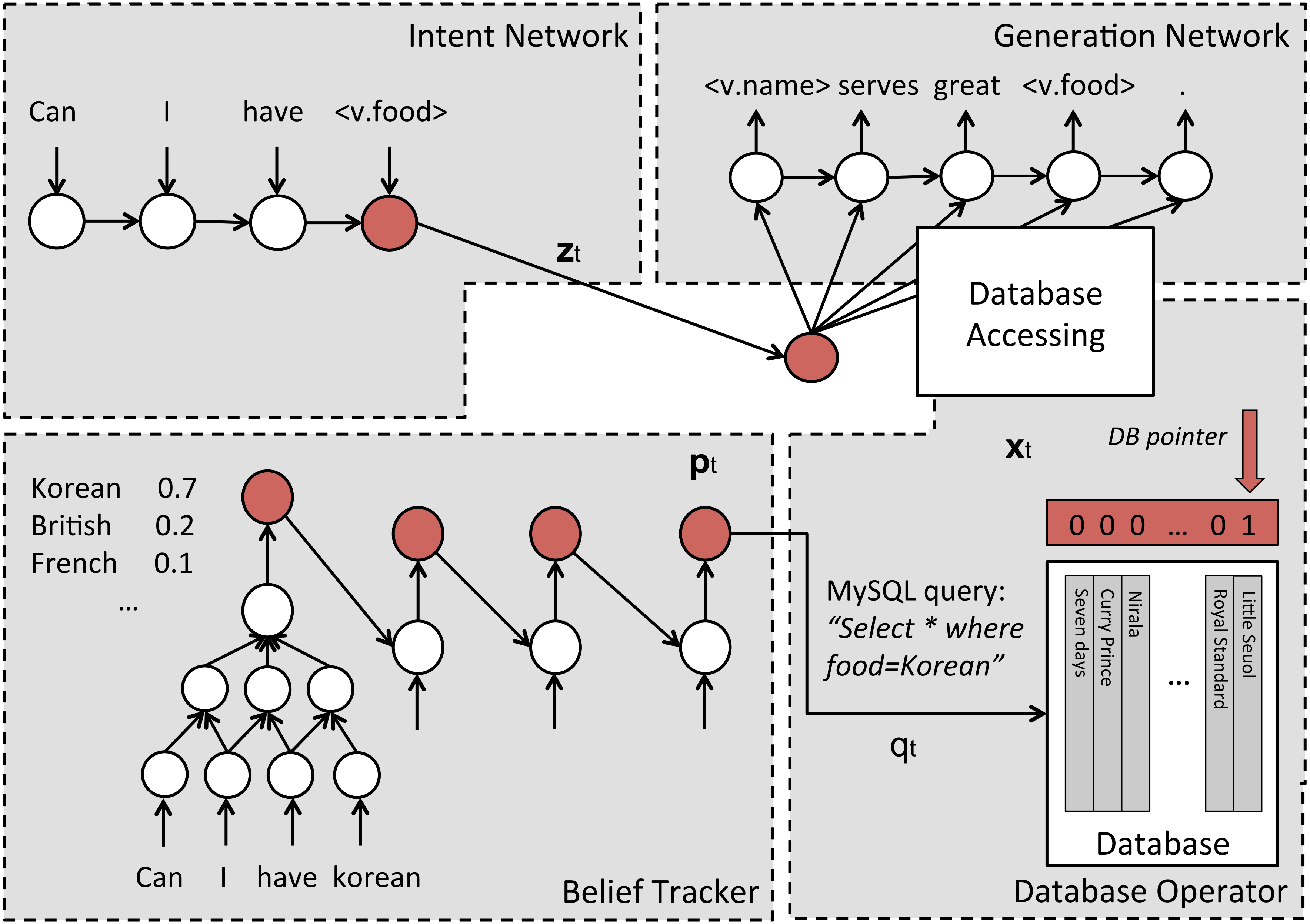
A Network-based End-to-End Trainable Task-Oriented Dialogue System, Wen et al, 2016b



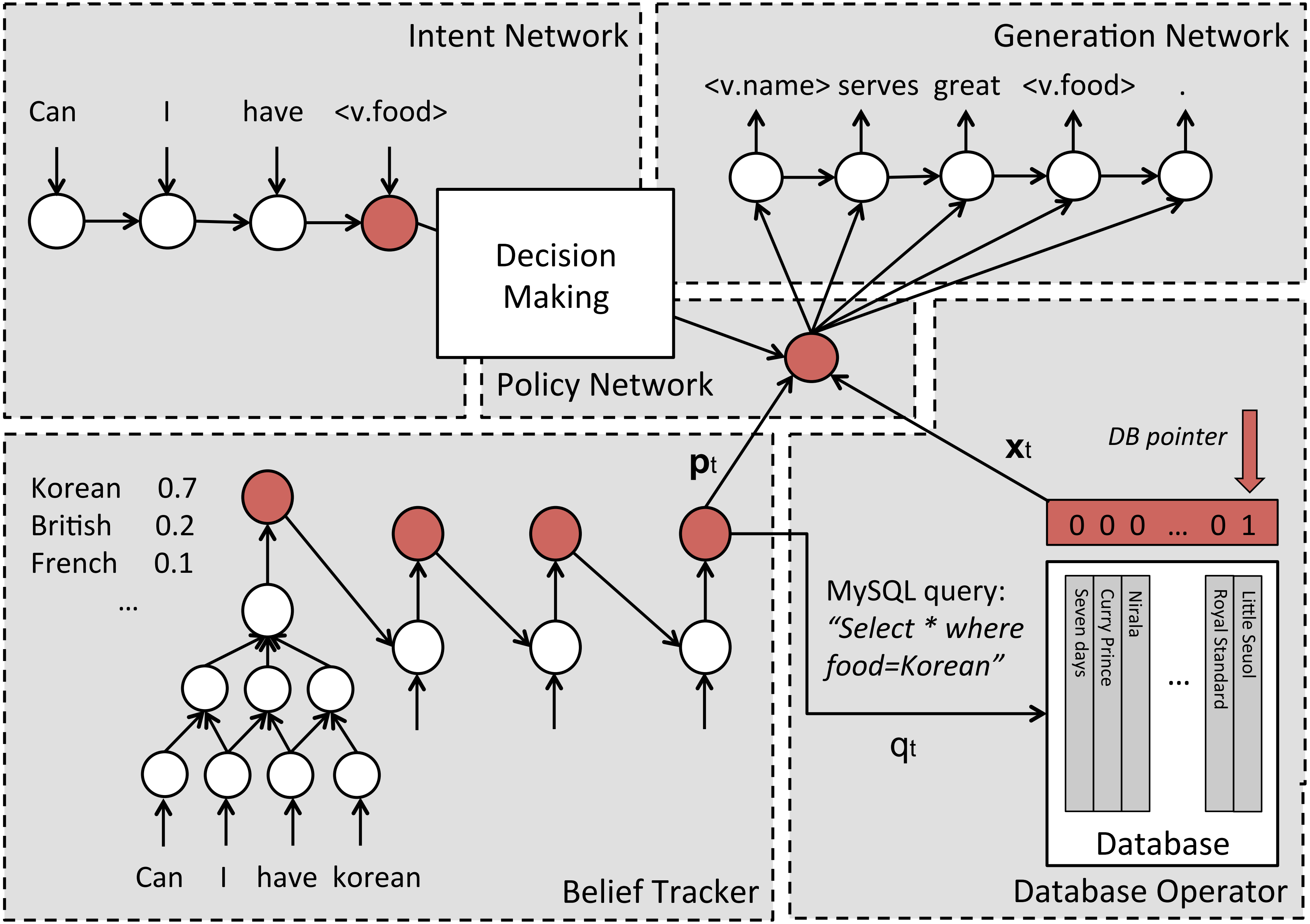
A Network-based End-to-End Trainable Task-Oriented Dialogue System, Wen et al, 2016b



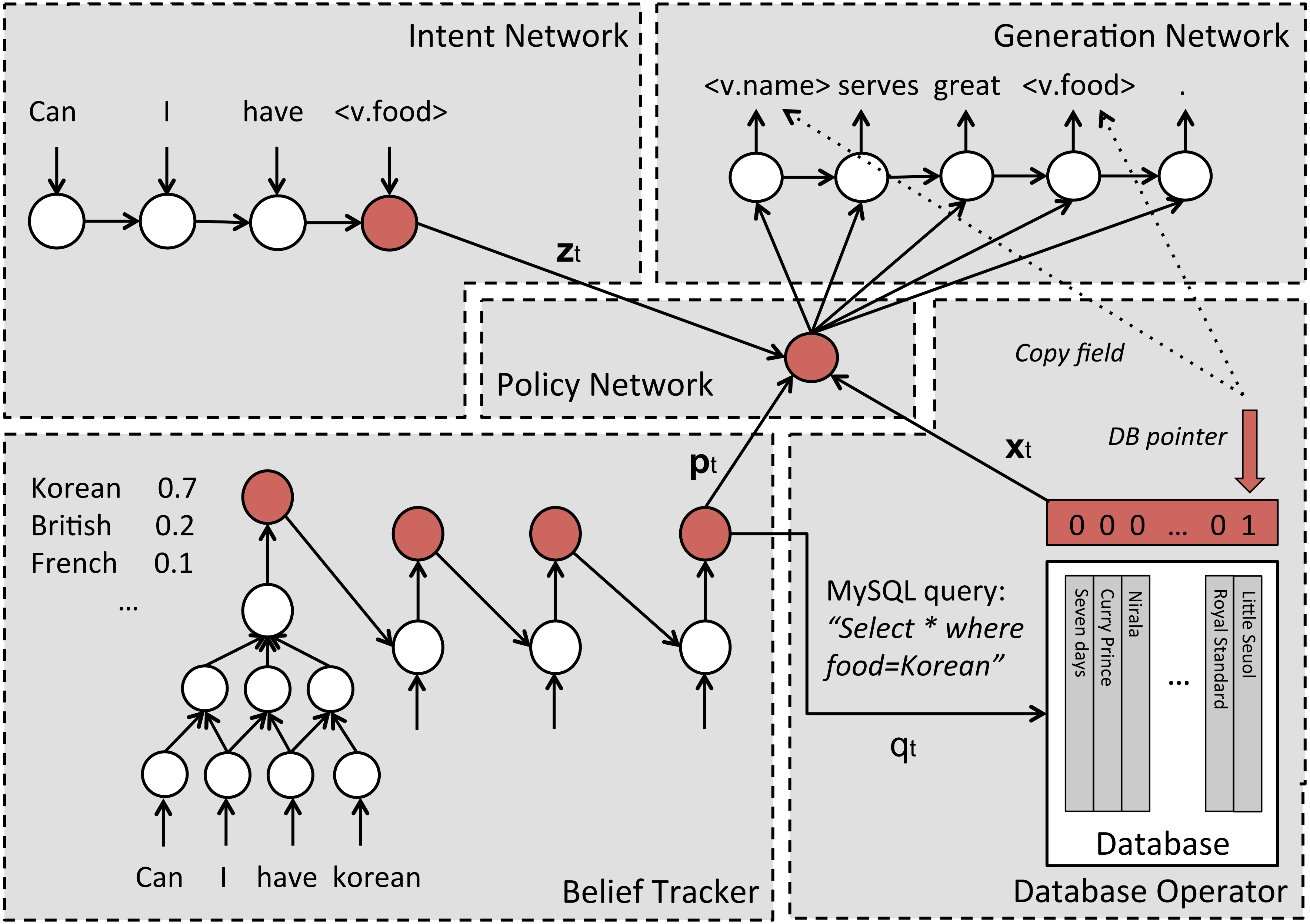
A Network-based End-to-End Trainable Task-Oriented Dialogue System, Wen et al, 2016b



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



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



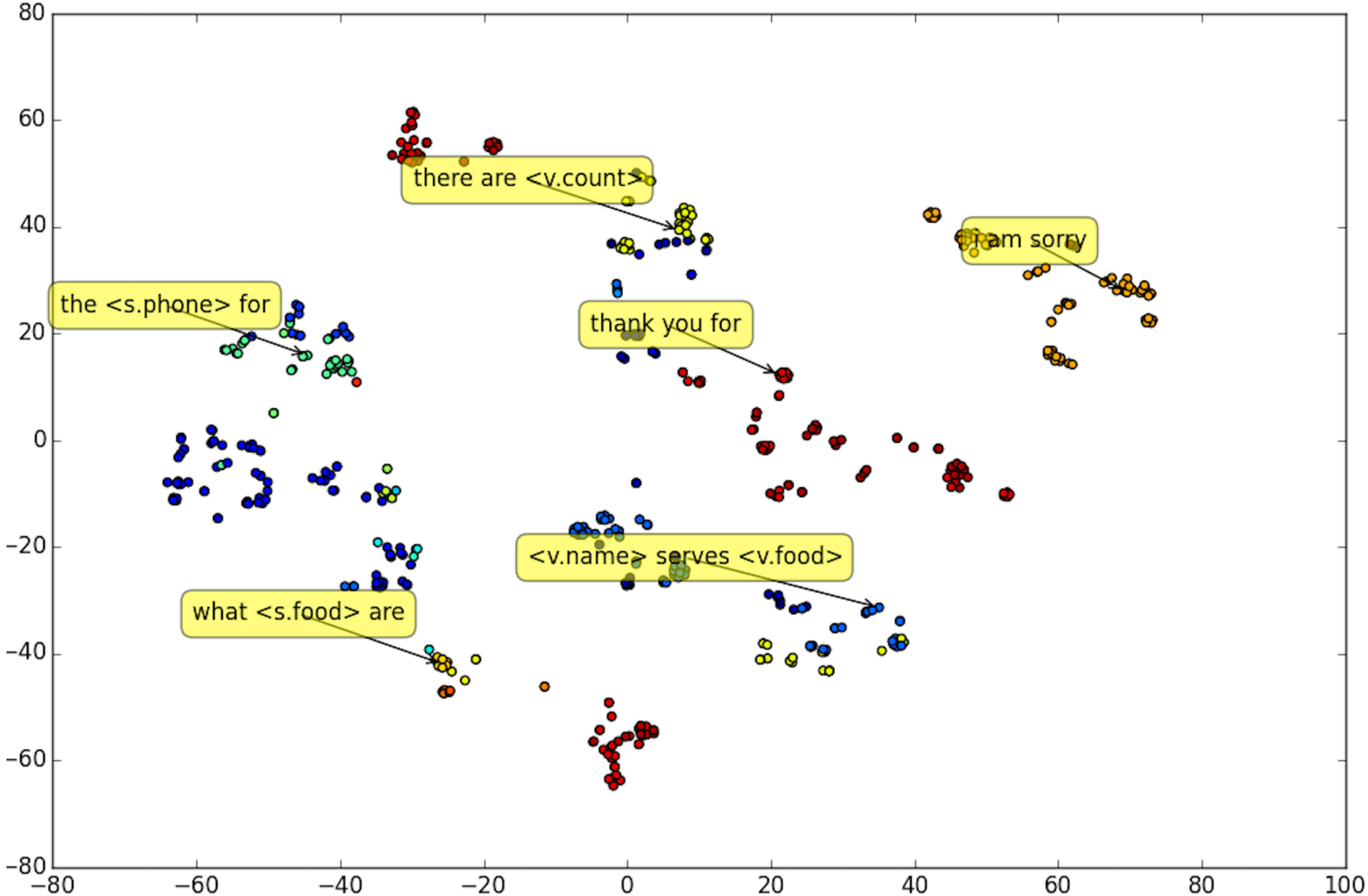
A Network-based End-to-End Trainable Task-Oriented Dialogue System, Wen et al, 2016b

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	:	Good bye
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



Short Conclusion

- Benefits of DL in complex NLP systems
  - **Distributed representation** – "AGAIN" Generalisation
  - **Recurrent connection** – Learning "RAW" inputs
  - **Conditional RNN** – "MULTIMODAL" sources

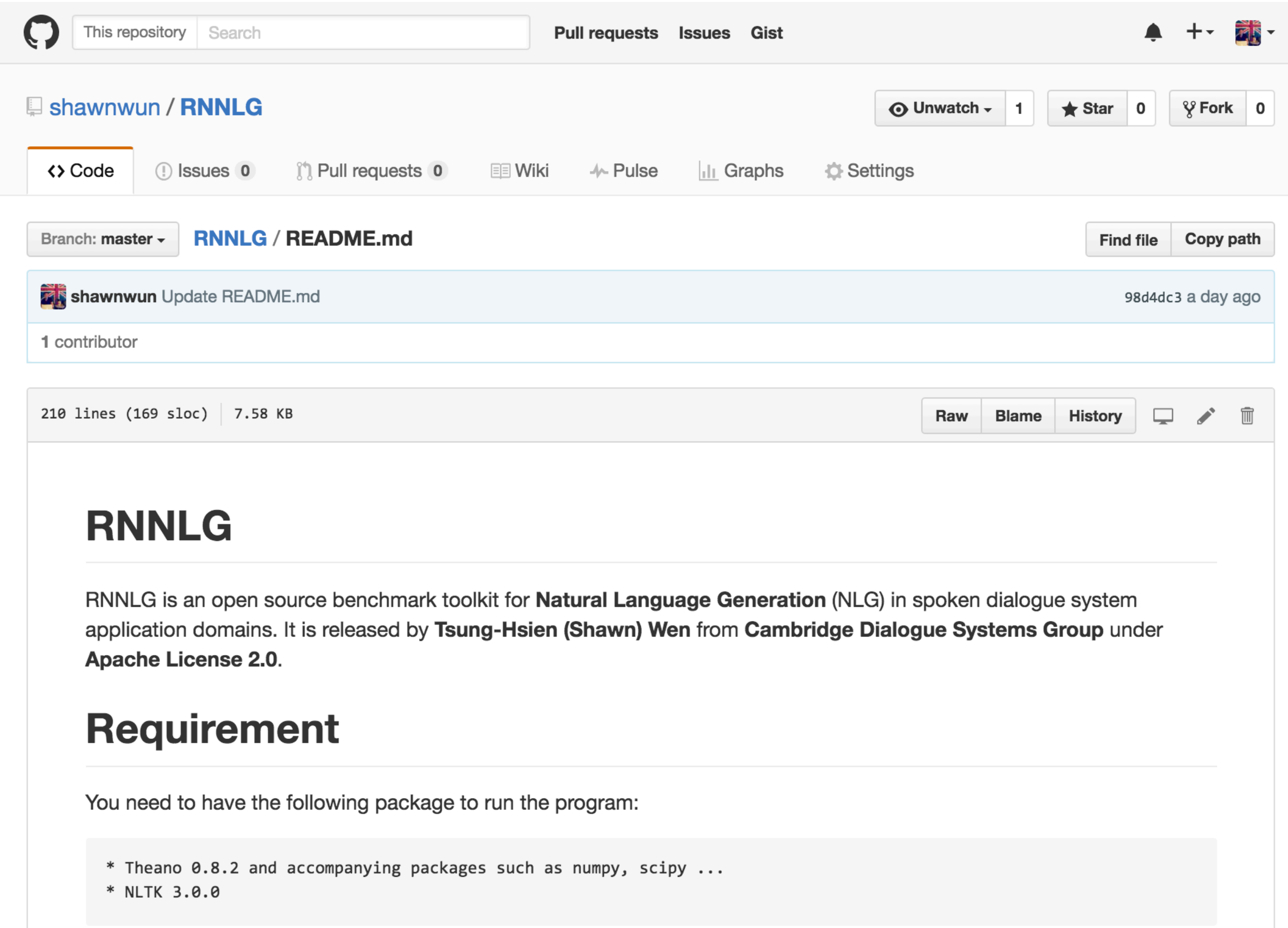
- DL allows us to build complex NLP learning systems like ever before.
- It is ambitious to learn EVERYTHING
  - Figure out what should be (shouldn't) learned.
- RL for online fine-tuning? [Su et al 2016].

Q & A

Part III: Codes

- Example codes for implementing deep NLG models in Theano

RNNLG – Benchmark toolkit for Neural NLG



- 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:
  - ✓ *Stochastic Language Generation in Dialogue using Recurrent Neural Networks with Convolutional Sentence Reranking*, Wen et al, SigDial 2015a.
  - ✓ *Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems*, Wen et al, EMNLP 2015b.
  - ✓ *Toward Multi-domain Language Generation using Recurrent Neural Networks*, Wen et al, NIPS workshop on ML for SLU & Interaction 2015c.
  - ✓ *Multi-domain Neural Network Language Generation for Spoken Dialogue Systems*, Wen et al, NAACL 2016a.

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 main.py -config config/ngram.cfg -mode ngram
python main.py -config config/knn.cfg   -mode knn
```

- Training SC-LSTM (run in background):

```
python main.py -config config/sclstm.cfg -mode train
```

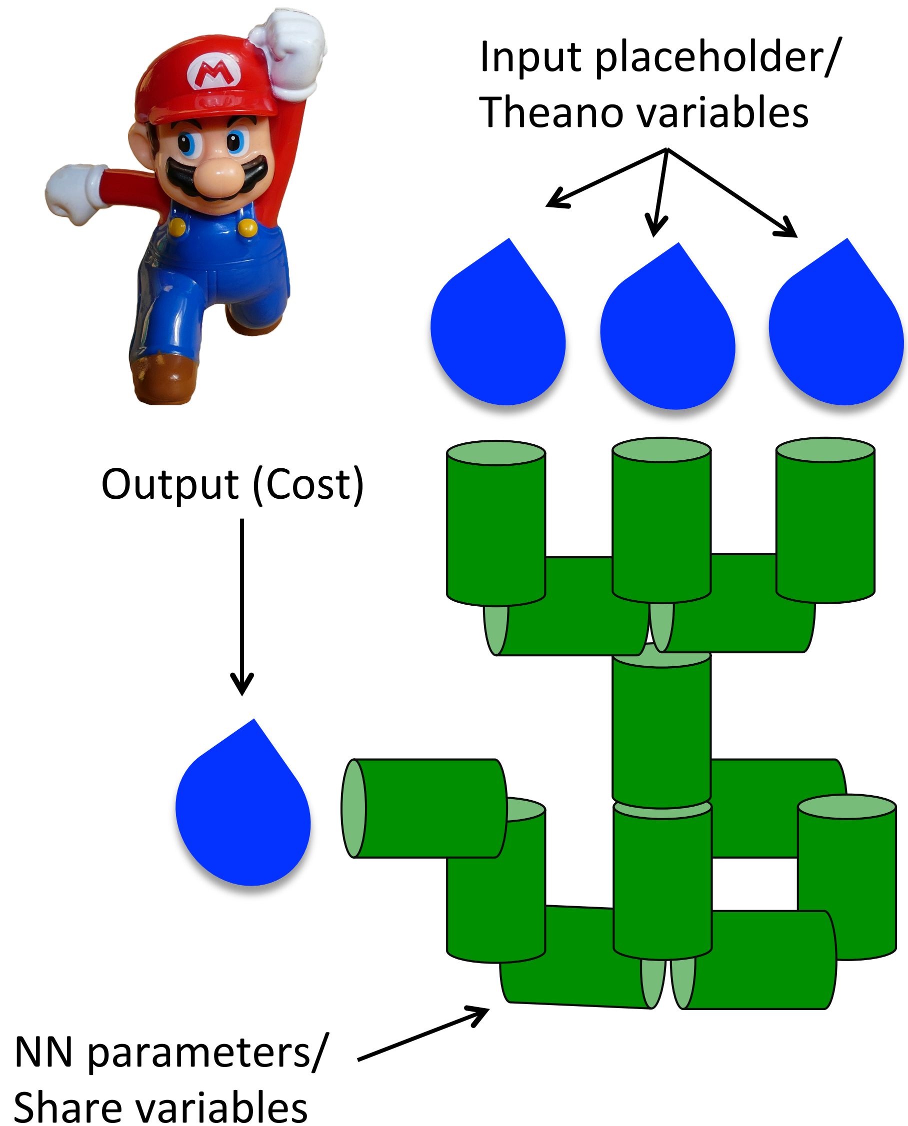
```
python main.py -config config/sclstm.cfg -mode test
```

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_tm1, h_tm1, c_tm1, 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_tm1,sv_tm1],axis=1),self.Wgate)
    ig  = T.nnet.sigmoid(gates_t[:,: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_tm1,sv_tm1],axis=1),self.Wrgate))
    # compute proposed cell value
    cx_t= T.tanh(T.dot(T.concatenate([wv_t,h_tm1],axis=1),self.Wcx))
    # update DA 1-hot vector
    sv_t = rg*sv_tm1
    # update lstm internal state
    c_t = ig*cx_t + fg*c_tm1 + \
            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.Who) )
    p_t = o_t[T.arange(self.db),y_t]

    return sv_t, h_t, c_t, p_t
```

$$\mathbf{i}_t = \sigma(\mathbf{W}_{wi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1})$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{wf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1})$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{wo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1})$$

$$\mathbf{r}_t = \sigma(\mathbf{W}_{wr}\mathbf{x}_t + \mathbf{W}_{hr}\mathbf{h}_{t-1})$$

$$\hat{\mathbf{c}}_t = \tanh(\mathbf{W}_{wc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1})$$

$$\mathbf{d}_t = \mathbf{r}_t \odot \mathbf{d}_{t-1}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \hat{\mathbf{c}}_t + \tanh(\mathbf{W}_{dc}\mathbf{d}_t)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t)$$

$$\mathbf{p}_t = \text{softmax}(\mathbf{W}_{ho}\mathbf{h}_t)$$

Define inputs/outputs

[RNNLG toolkit, nn/NNGenerator.py]

Input placeholders

```
# input tensor variables
w_idxes = T.imatrix('w_idxes')
w_idxes = T.imatrix('w_idxes')
a       = T.imatrix('a')
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')
```

Interface between Theano and python

```
# theano functions
self.train = theano.function(
        inputs= [a,sv,s,v, w_idxes, cutoff_f, cutoff_b, lr, reg],
        outputs=-self.cost,
        updates=updates,
        on_unused_input='ignore')
self.test  = theano.function(
        inputs= [a,sv,s,v, w_idxes, cutoff_f, cutoff_b],
        outputs=-self.cost,
        on_unused_input='ignore')
```

Output cost, gradient, update function

```
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 ) \
        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 Automatic Extraction of Rules for Sentence Plan Construction"*, Amanda Stent and Martin Molina, SigDial 2009
- *"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
- *"Conditional Random Fields for Responsive Surface Realisation Using Global Features"*, Nina Dethlefs, Helen Hastie, Heriberto Cuayáhuitl, Oliver Lemon, ACL 2013
- *"Training a statistical surface realiser from automatic slot labelling"*, Heriberto Cuayáhuitl and Nina Dethlefs and Helen Hastie and Xingkun Liu, IEEE SLT 2014
- *"Stochastic Language Generation for Spoken Dialogue Systems"*, Alice H. Oh and Alexander I. Rudnicky, NAACL workshop on Conversational Systems 2000
- *"Phrase-based Statistical Language Generation Using Graphical Models and Active Learning"*, Francois Mairesse, Milica Gasic, Filip Jurcicek, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young, ACL 2010
- *"Training a Natural Language Generator From Unaligned Data"*, Ondrej Dusek, Filip Jurcicek, ACL 2015

References

Neural Networks

- *"A Neural Probabilistic Language Model"*, Yoshua Bengio, Rejean Ducharme, Pascal Vincent, NIPS 2001
- *"Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation"*, Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio, EMNLP 2014
- *"Recurrent neural network based language model"*, Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Honza Cernocky,Sanjeev Khudanpur, InterSpeech 2010
- *"On the difficulty of training recurrent neural networks*", Razvan Pascanu, Tomas Mikolov, Yoshua Bengio, ICML 2013
- *"Long Short-Term Memory"*, Sepp Hochreiter and Jurgen Schmidhuber, Neural Computation 1997

References

Text Generation

- "*Generating Text with Recurrent Neural Networks*", Ilya Sutskever, James Martens, Geoffrey E. Hinton, ICML 2011.

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.
- "*Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*", Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, Yoshua Bengio, EMNLP 2014.
- "*Neural Machine Translation by Jointly Learning to Align and Translate*", Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio, ICLR 2015.

Image Caption Generation

- "*Deep Visual-Semantic Alignments for Generating Image Descriptions*", Andrej Karpathy, Fei-Fei Li, CVPR 2015.
- "*Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*", Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, Yoshua Bengio, ICML 2015

References

Natural Language Generation

- "*Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking*", Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young, SigDial 2015a.
- "*Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems*", Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young, EMNLP 2015b.
- "*Toward Multi-domain Language Generation using Recurrent Neural Networks*", Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina M. R.-Barahona, Pei-Hao Su, David Vandyke, and Steve Young, NIPS Workshop on ML for SLU 2015c.
- "*Multi-domain neural network language generation for spoken dialogue systems*", Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young, NAACL 2016a.
- "*What to talk about and how? Selective Generation using LSTMs with Coarse-to-Fine Alignment*", Hongyuan Mei, Mohit Bansal, Matthew R. Walter, NAACL 2016.
- "*Sequence-to-Sequence Generation for Spoken Dialogue via Deep Syntax Trees and Strings*", Ondrej Dusek, Filip Jurcicek, ACL 2016.

References

N2N Response Generation (chitchat)

- "*A Neural Conversational Model*", Oriol Vinyals, Quoc V. Le, ICML Deep Learning Workshop 2015.
- "*Neural Responding Machine for Short-Text Conversation*", Lifeng Shang, Zhengdong Lu, Hang Li, ACL 2015.
- "*Hierarchical Neural Network Generative Models for Movie Dialogues*", Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, Joelle Pineau, AAAI 2015.
- "*A Diversity-Promoting Objective Function for Neural Conversation Models*", Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, Bill Dolan, NAACL 2016a.
- "*A Persona-Based Neural Conversation Model*", Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, Bill Dolan, ACL 2016b.
- "*Deep Reinforcement Learning for Dialogue Generation*", Jiwei Li, Will Monroe, Alan Ritter and Dan Jurafsky, EMNLP 2016c.

References

Dialogue Response Generation (goal-oriented)

- "*A Network-based End-to-End Trainable Task-Oriented Dialogue System*", Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young, arXiv preprint:1604.04562, 2016b.
- "*Conditional Generation and Snapshot Learning in Neural Dialogue Systems*", Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young, EMNLP 2016c.
- "*Continuously Learning Neural Dialogue Management*", Pei-Hao Su, Milica Gasic, Nikola Mrksic, Lina Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young, arXiv preprint:1606.02689, 2016.

Parsing

- "*Grammar as a Foreign Language*", Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, Geoffrey Hinton, NIPS 2015.

Code Generation

- "*Latent Predictor Networks for Code Generation*", Wang Ling, Edward Grefenstette, Karl Moritz Hermann, Tomas Kocisky, Andrew Senior, Fumin Wang, Phil Blunsom, ACL 2016.

UNIVERSITY OF CAMBRIDGE

Thank you! Questions?

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