UNIVERSITY OF CAMBRIDGE

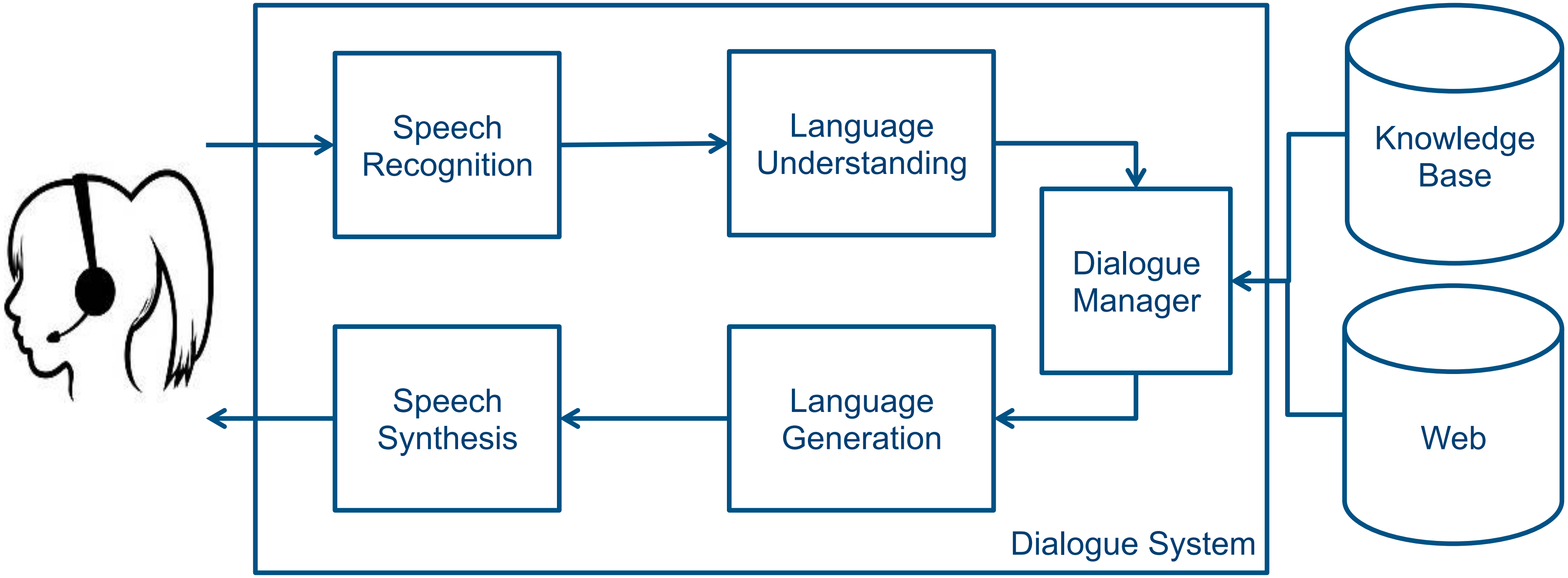
Scalable Neural Language Generation for Open Domain Dialogue Systems

Speaker: Tsung-Hsien Wen

Supervisor: Professor Steve Young

Dialogue Systems Group

Spoken Dialogue System

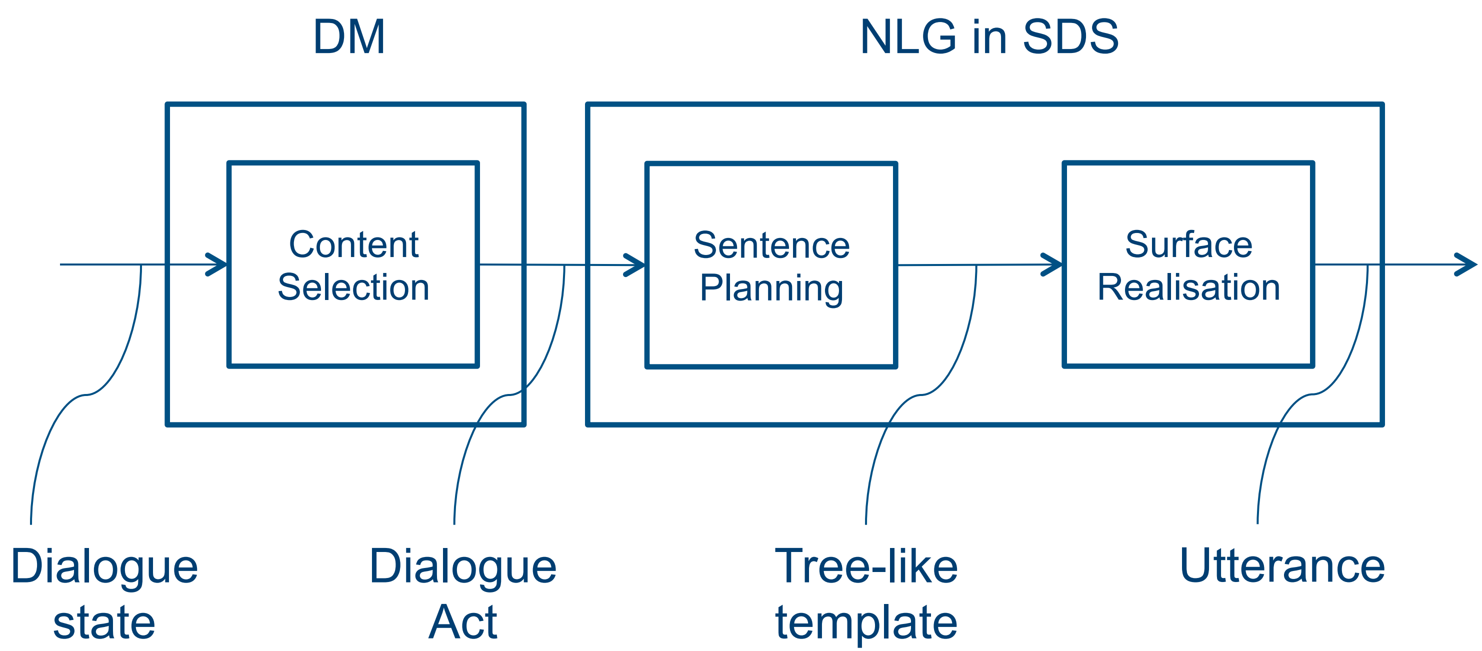


NLG: Problem Definition

- 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



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.

UNIVERSITY OF CAMBRIDGE

Part 1
Heuristically Gated RNN Generator

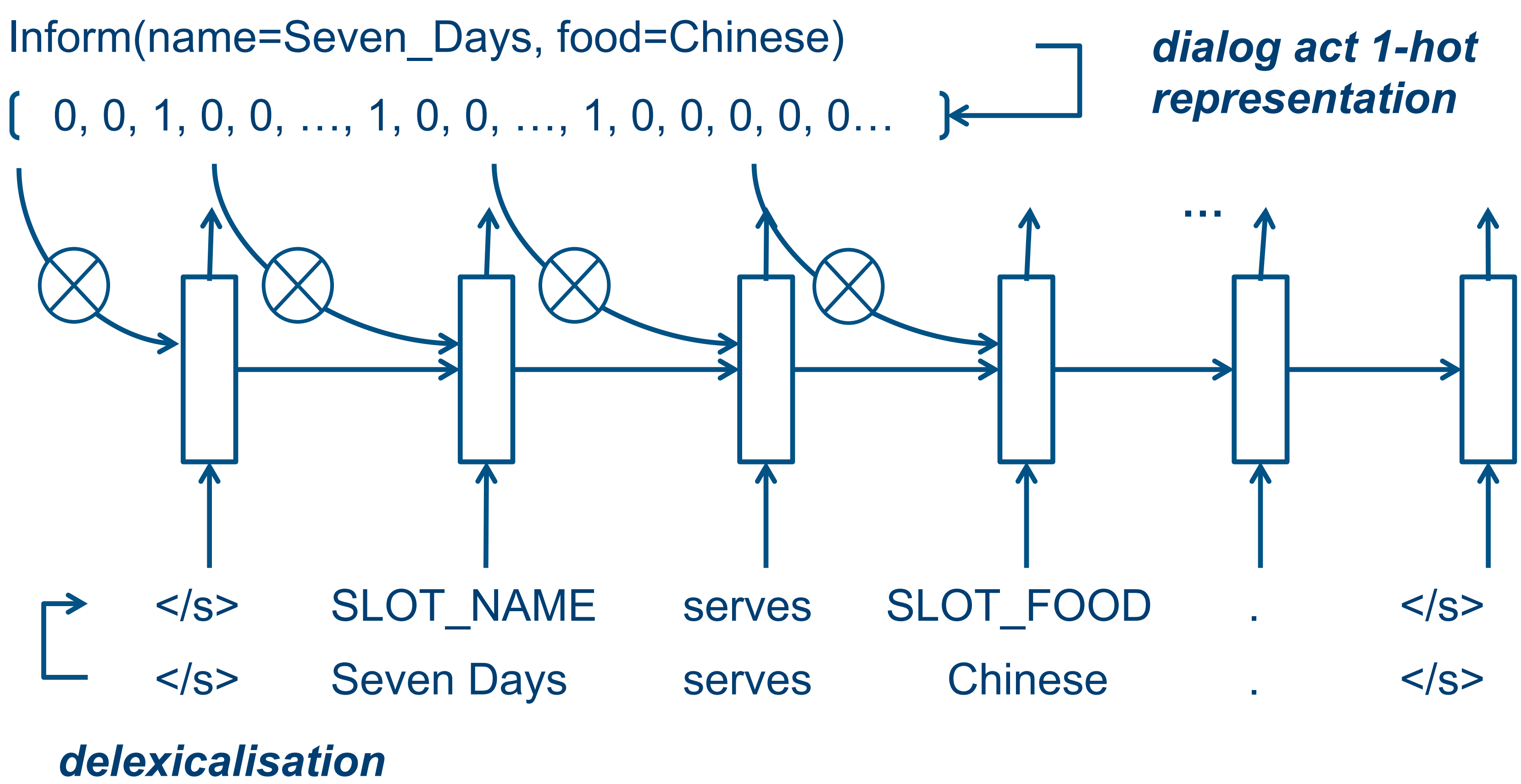
Outline

- 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)



(Mikolov et al 2010)

Recurrent Generation Model (2/2)

- 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.

Feature value	**</s>**	**SLOT_NAME**	**serves**	**SLOT_FOOD**	**.**	**</s>**
NAME	1	1	δ	δ^2	δ^3	δ^4
FOOD	1	1	1	1	δ	δ^2

Outline

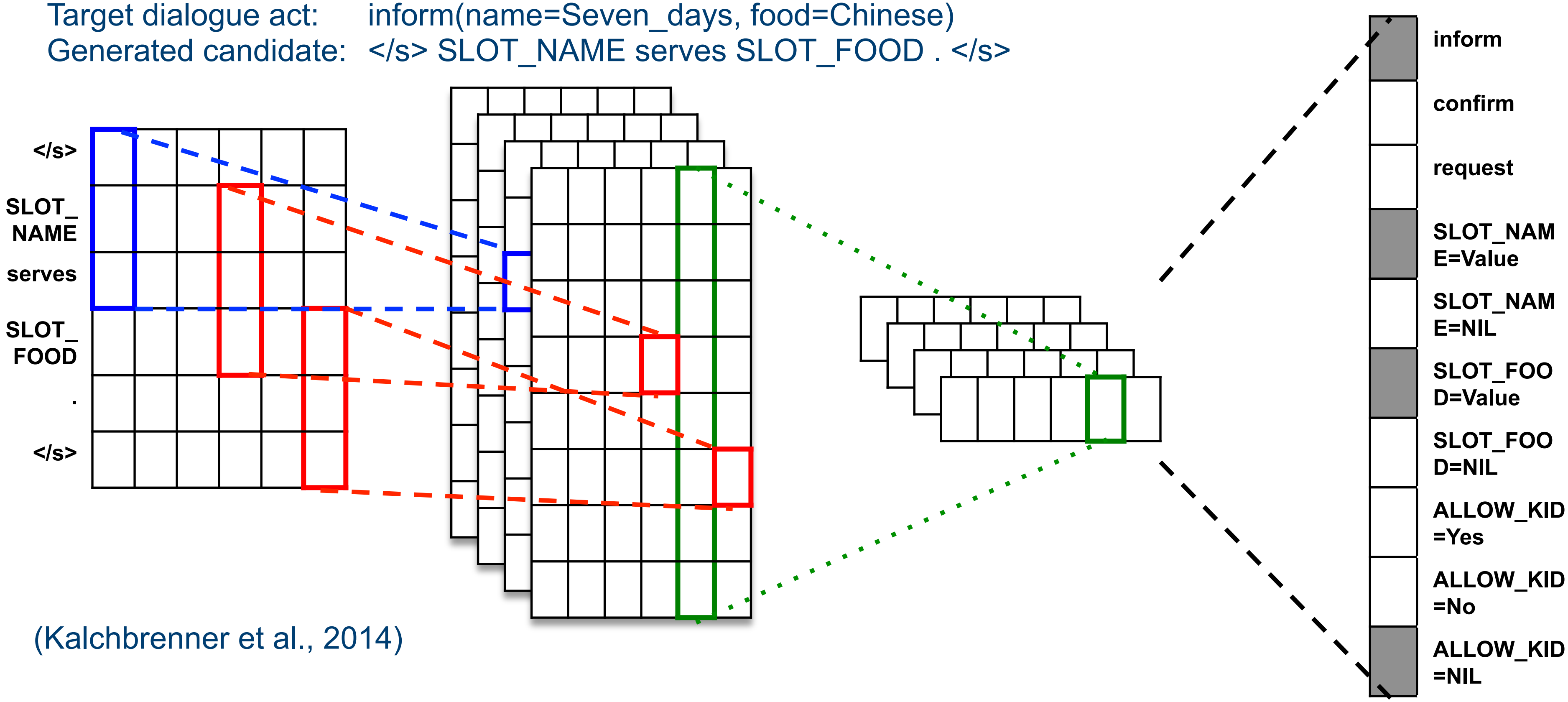
- Recurrent Generation Model
- **Convolutional Semantic Reranker**
- Backward RNN Reranker
- Experiments
  - Setup
  - Automatic Evaluation
  - Human Evaluation

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)

Sentence representation over delexicalised corpus

1-D convolutional layer with multiple feature maps

Average pooling over time

Fully connected layer for classifying dialogue act

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)

BLEU		hdc	knn	classlm	rnn
Selection Beam	1/20	0.440	0.591	0.757	0.777
	5/20	-	-	0.678	0.712

ERR		hdc	knn	classlm	rnn
Selection Beam	1/20	0	17.2	47.8	0
	5/20	-	-	104.6	3.1

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)

Metrics	hdc	rnn1	hdc	rnn5
Info.	3.75	3.81	3.85	3.93*
Nat.	3.58	3.74**	3.57	3.94**
Pref.	44.8%	55.2%*	37.2%	62.8%**
Metrics	**rnn1**	**rnn5**	**classlm5**	**rnn5**
Info.	3.75	3.72	4.02	4.15%*
Nat.	3.67	3.58	3.91	4.02
Pref.	47.5%	52.5%	47.1%	52.9%

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

Human Evaluation (2/3)

Metrics	hdc	rnn1	hdc	rnn5
Info.	3.75	3.81	3.85	3.93*
Nat.	3.58	3.74**	3.57	3.94**
Pref.	44.8%	55.2%*	37.2%	62.8%**
Metrics	**rnn1**	**rnn5**	**classlm5**	**rnn5**
Info.	3.75	3.72	4.02	4.15*
Nat.	3.67	3.58	3.91	4.02
Pref.	47.5%	52.5%	47.1%	52.9%

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

Human Evaluation (2/3)

Metrics	hdc	rnn1	hdc	rnn5
Info.	3.75	3.81	3.85	3.93*
Nat.	3.58	3.74**	3.57	3.94**
Pref.	44.8%	55.2%*	37.2%	62.8%**
Metrics	**rnn1**	**rnn5**	**classlm5**	**rnn5**
Info.	3.75	3.72	4.02	4.15%*
Nat.	3.67	3.58	3.91	4.02
Pref.	47.5%	52.5%	47.1%	52.9%

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

Human Evaluation (2/3)

Metrics	hdc	rnn1	hdc	rnn5
Info.	3.75	3.81	3.85	3.93*
Nat.	3.58	3.74**	3.57	3.94**
Pref.	44.8%	55.2%*	37.2%	62.8%**
Metrics	**rnn1**	**rnn5**	**classlm5**	**rnn5**
Info.	3.75	3.72	4.02	4.15*
Nat.	3.67	3.58	3.91	4.02
Pref.	47.5%	52.5%	47.1%	52.9%

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

Human Evaluation (2/3)

Metrics	hdc	rnn1	hdc	rnn5
Info.	3.75	3.81	3.85	3.93*
Nat.	3.58	3.74**	3.57	3.94**
Pref.	44.8%	55.2%*	37.2%	62.8%**
Metrics	**rnn1**	**rnn5**	**classlm5**	**rnn5**
Info.	3.75	3.72	4.02	4.15%*
Nat.	3.67	3.58	3.91	4.02
Pref.	47.5%	52.5%	47.1%	52.9%

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

Human Evaluation (3/3)

#	**Dialogue act and example realisations of our system, by sampling from top-5 candidates**
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.

UNIVERSITY OF CAMBRIDGE

Part 2
Semantically Controlled LSTM Generator

Dialogue Systems Group

Outline

- **SC-LSTM**
- Deep Model
- Experiments
  - Automatic Evaluation
  - Human Evaluation

SC-LSTM (1/4)

- Original LSTM cell

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

$$\mathbf{f}_t = \sigma(\mathbf{W}_{wf}\mathbf{w}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1}) \quad (2)$$

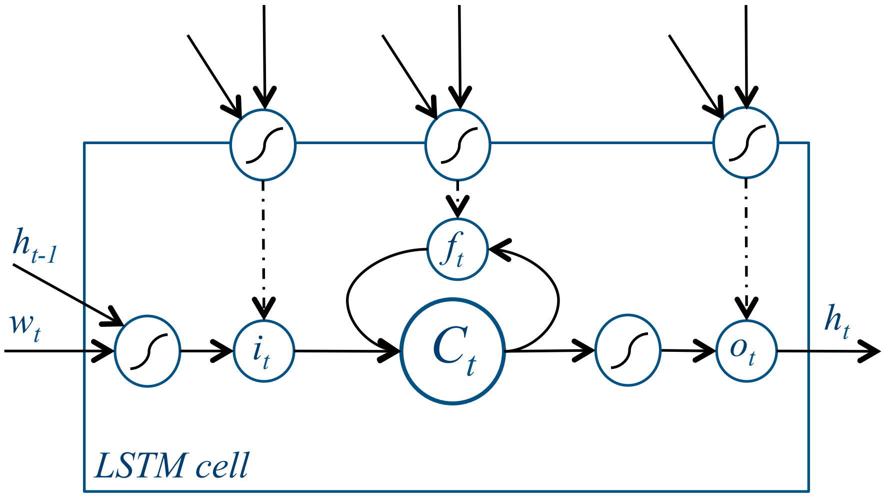
$$\mathbf{o}_t = \sigma(\mathbf{W}_{wo}\mathbf{w}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1}) \quad (3)$$

$$\hat{\mathbf{c}}_t = tanh(\mathbf{W}_{wc}\mathbf{w}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1}) \quad (4)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \hat{\mathbf{c}}_t \quad (5)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot tanh(\mathbf{c}_t) \quad (6)$$

(Hochreiter and Schmidhuber, 1997)



SC-LSTM (2/4)

- Original LSTM cell

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

$$\mathbf{f}_t = \sigma(\mathbf{W}_{wf}\mathbf{w}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1}) \quad (2)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{wo}\mathbf{w}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1}) \quad (3)$$

$$\hat{\mathbf{c}}_t = tanh(\mathbf{W}_{wc}\mathbf{w}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1}) \quad (4)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \hat{\mathbf{c}}_t \quad (5)$$

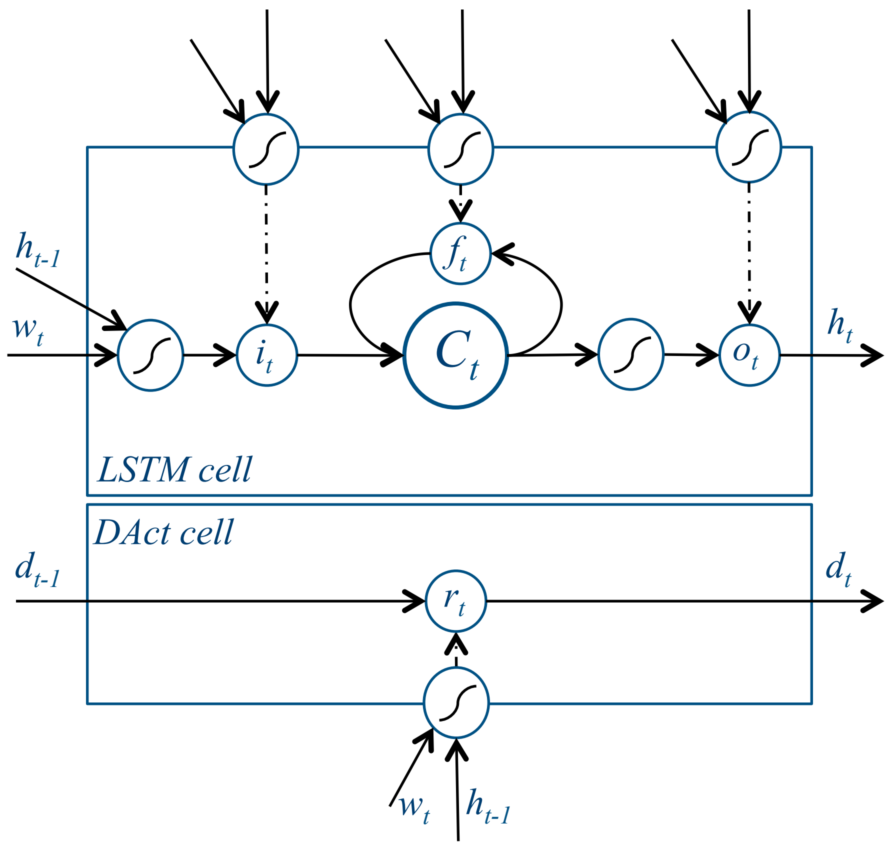
$$\mathbf{h}_t = \mathbf{o}_t \odot tanh(\mathbf{c}_t) \quad (6)$$

- DA cell

$$\mathbf{r}_t = \sigma(\mathbf{W}_{wr}\mathbf{w}_t + \alpha\mathbf{W}_{hr}\mathbf{h}_{t-1}) \quad (7)$$

$$\mathbf{d}_t = \mathbf{r}_t \odot \mathbf{d}_{t-1} \quad (8)$$

(Hochreiter and Schmidhuber, 1997)



SC-LSTM (3/4)

- Original LSTM cell

$$\mathbf{i}_t = \sigma(\mathbf{W}_{wi}\mathbf{w}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1}) \quad (1)$$
$$\mathbf{f}_t = \sigma(\mathbf{W}_{wf}\mathbf{w}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1}) \quad (2)$$
$$\mathbf{o}_t = \sigma(\mathbf{W}_{wo}\mathbf{w}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1}) \quad (3)$$
$$\hat{\mathbf{c}}_t = tanh(\mathbf{W}_{wc}\mathbf{w}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1}) \quad (4)$$
$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \hat{\mathbf{c}}_t \quad (5)$$
$$\mathbf{h}_t = \mathbf{o}_t \odot tanh(\mathbf{c}_t) \quad (6)$$

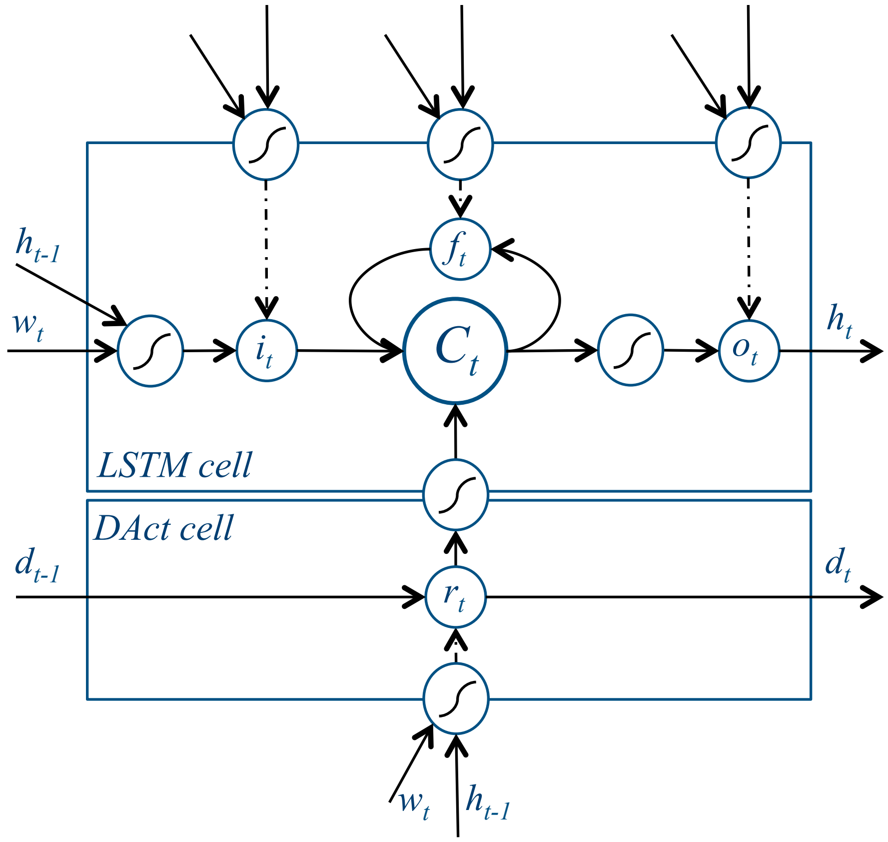
- DA cell

$$\mathbf{r}_t = \sigma(\mathbf{W}_{wr}\mathbf{w}_t + \alpha\mathbf{W}_{hr}\mathbf{h}_{t-1}) \quad (7)$$
$$\mathbf{d}_t = \mathbf{r}_t \odot \mathbf{d}_{t-1} \quad (8)$$

- Modify eq. (6) to

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \hat{\mathbf{c}}_t + tanh(\mathbf{W}_{dc}\mathbf{d}_t) \quad (9)$$

(Hochreiter and Schmidhuber, 1997)



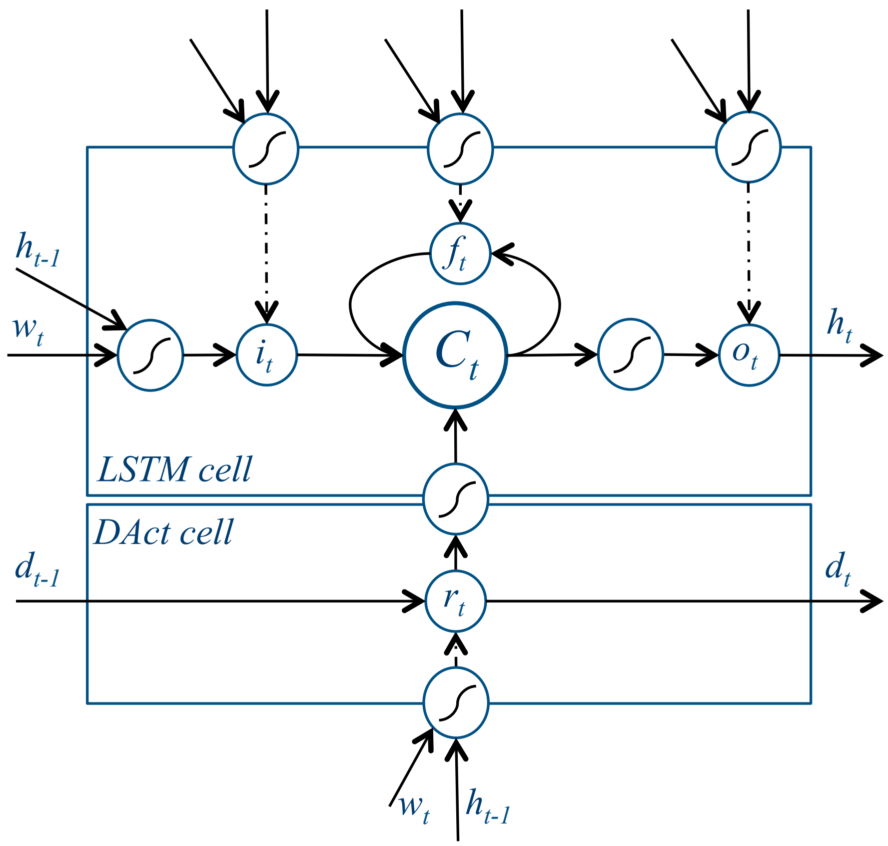
SC-LSTM (4/4)

- Cost function

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

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

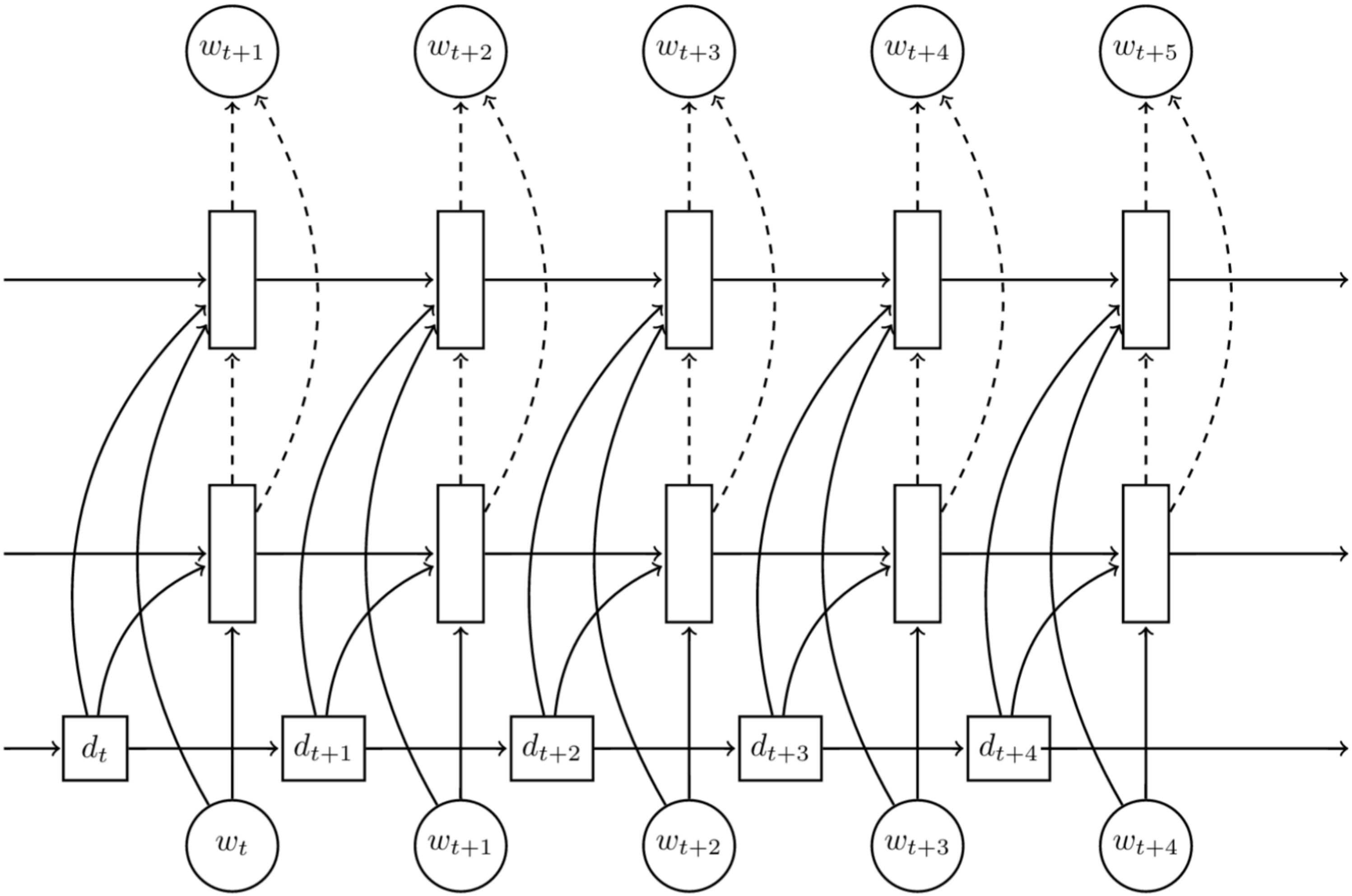
(Hochreiter and Schmidhuber, 1997)



Outline

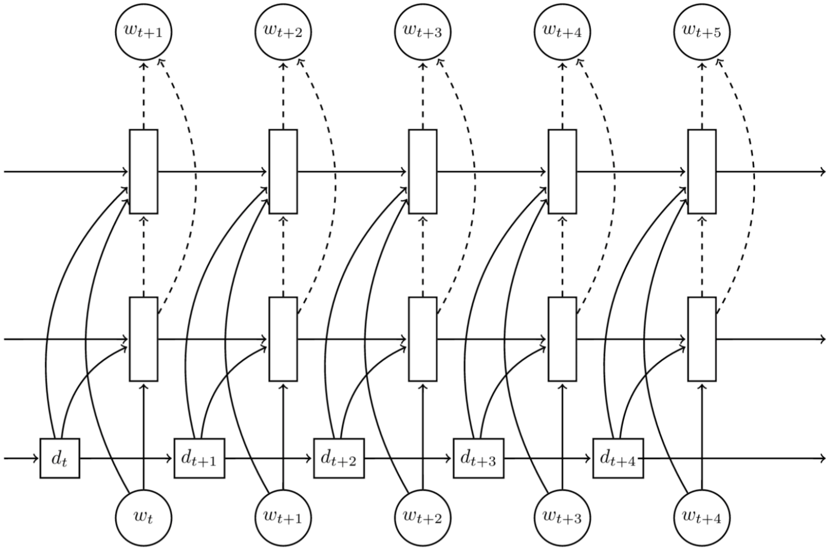
- SC-LSTM
- **<u>Deep Model</u>**
- Experiments
  - Automatic Evaluation
  - Human Evaluation

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

$$\mathbf{r}_t = \sigma(\mathbf{W}_{wr}\mathbf{w}_t + \alpha \mathbf{W}_{hr}\mathbf{h}_{t-1}) \qquad (7)$$

- To

$$\mathbf{r}_t = \sigma(\mathbf{W}_{wr}\mathbf{w}_t + \sum_l \alpha_l \mathbf{W}^l_{hr}\mathbf{h}^l_{t-1}) \qquad (12)$$

Outline

- SC-LSTM
- Deep Model
- **Experiments**
  - **Automatic Evaluation**
  - Human Evaluation

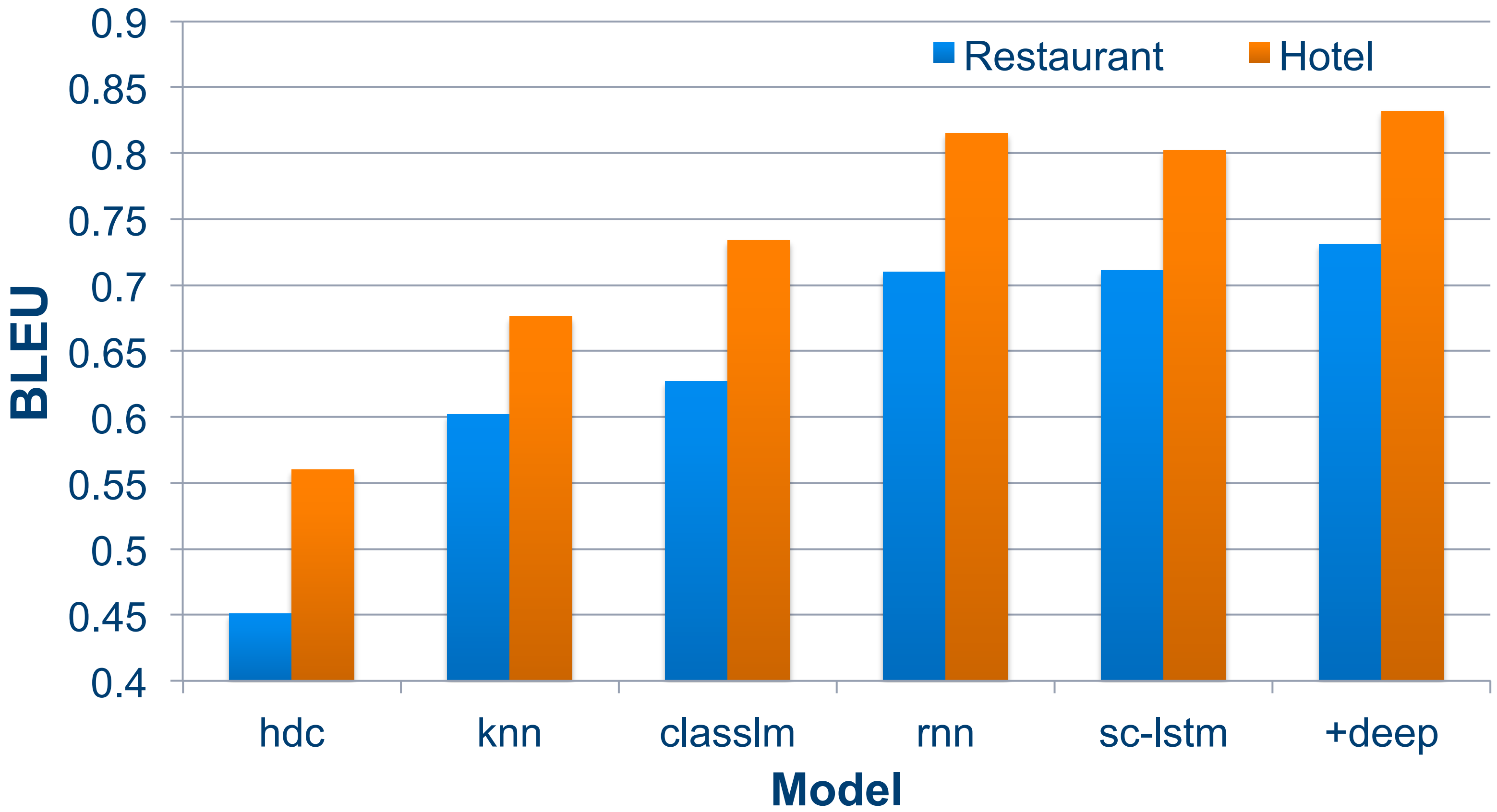
Automatic Evaluation (1/3)

- 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:

	SF Restaurant	SF Hotel
act type	inform, inform_only, reject, confirm, select, request, reqmore, goodbye	
shared	name, type, *pricerange, price, phone, address, postcode, *area, *near	
specific	*food *goodformeal ***kids-allowed**	***hasinternet** ***acceptscards** ***dogs-allowed**

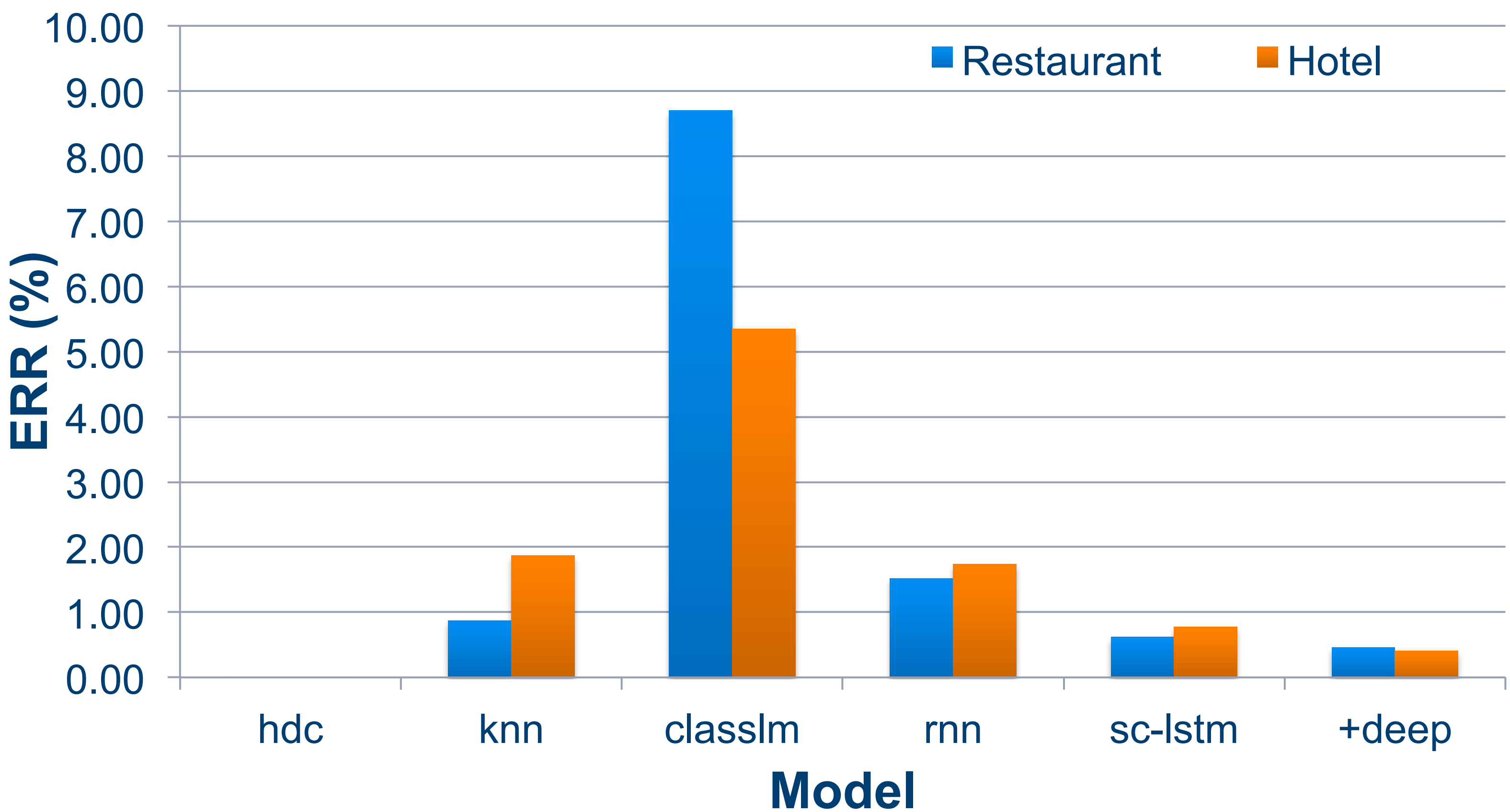
bold=binary slots, *=slots can take “don’t care” value

Automatic Evaluation (2/3)



Selection scheme : 5/20

Automatic Evaluation (3/3)



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 *classlm*, *rnn w/*, *sc-lstm* and +*deep*
- Metrics
  - Informativeness, Naturalness, Preference

Human Evaluation (2/3)

Method	Informativeness	Naturalness
+deep	2.58	**2.51**
sc-lstm	**2.59**	2.50
rnn w/	2.53	2.42^{*}
classlm	2.46^{**}	2.45

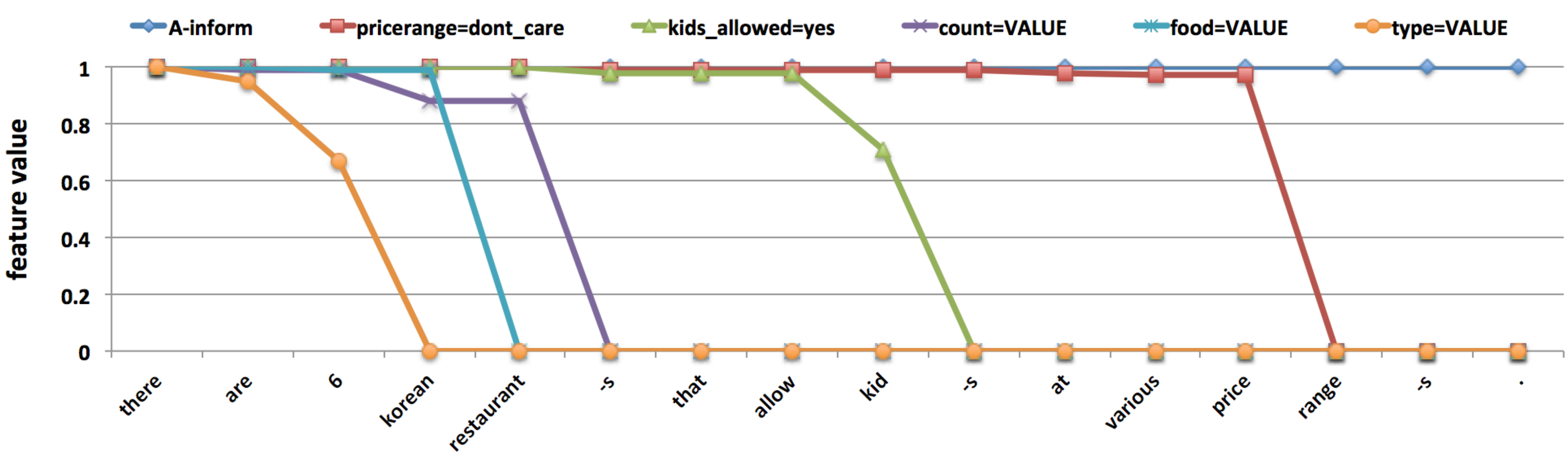
$^{*}p < 0.05$ $^{**}p < 0.005$

Human Evaluation (3/3)

Pref.%	**classlm**	**rnn w/**	**sc-lstm**	**+deep**
classlm	-	46.0	40.9^{**}	37.7^{**}
rnn w/	54.0	-	43.0	35.7^{*}
sc-lstm	59.1^{*}	57	-	47.6
+deep	62.3^{**}	64.3^{**}	52.4	-

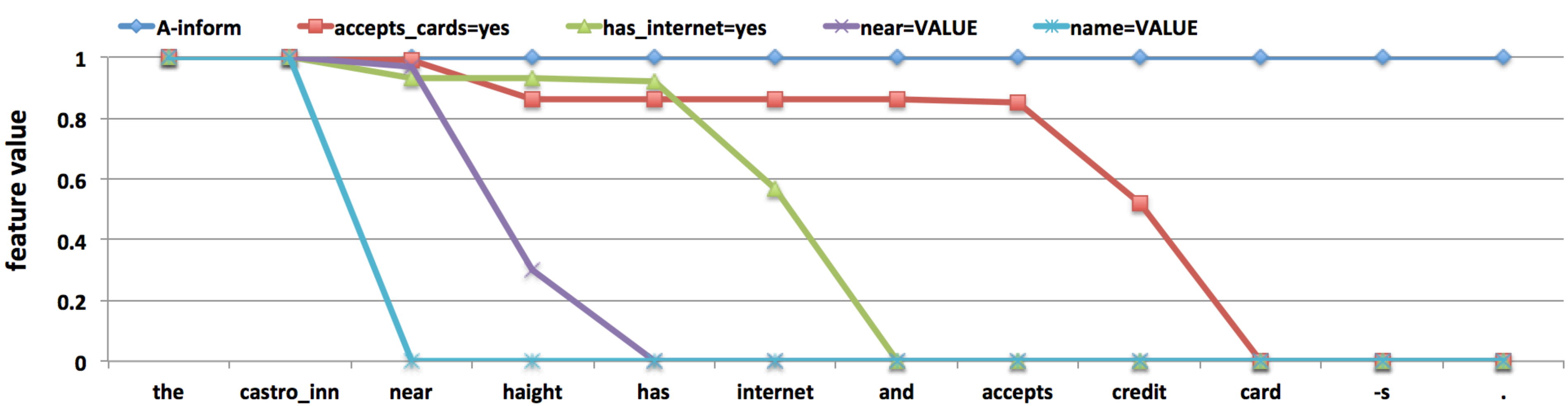
$^{*}p < 0.05$ $^{**}p < 0.005$

Example



A-inform
pricerange=dont_care
kids_allowed=yes
count=VALUE
food=VALUE
type=VALUE
feature value
1
0.8
0.6
0.4
0.2
0
there
are
6
korean
restaurant
-s
that
allow
kid
-s
at
various
price
range
-s
.

Example


A-inform
accepts_cards=yes
has_internet=yes
near=VALUE
name=VALUE
feature value
1
0.8
0.6
0.4
0.2
0
the
castro_inn
near
haight
has
internet
and
accepts
credit
card
-s
.

UNIVERSITY OF CAMBRIDGE

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

- Tsung-Hsien Wen, Milica Gasic , Dongho Kim, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. In *Proceedings of SIGdial*. Association for Computational Linguistics.
- Tsung-Hsien Wen, Milica Gasic , Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. To be appear In *Proceedings of EMNLP*. Association for Computational Linguistics.

Reference

- Amanda Stent, Matthew Marge, and Mohit Singhai. 2005. Evaluating evaluation methods for generation in the presence of variation. In Proceedings of CICLing 2005.
- Alice H. Oh and Alexander I. Rudnicky. 2000. Stochastic language generation for spoken dialogue systems. In Proceedings of the 2000 ANLP/NAACL Workshop on Conversational Systems.
- Tomas Mikolov, Martin Karafit, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. *In Proceedings on InterSpeech*.
- Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. Proceedings of the 52nd Annual Meeting of ACL.

Reference

- Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*.
- Alex Graves. 2013. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*.

UNIVERSITY OF CAMBRIDGE

Thank you! Questions?

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