Natural language generation

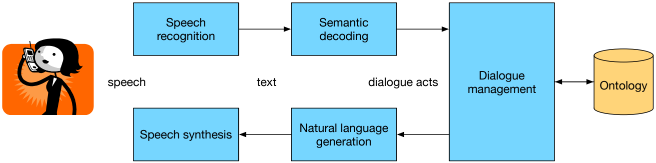
Milica Gašić

Dialogue Systems and Machine Learning Group,
Heinrich Heine University Düsseldorf

Natural language generation

Speech synthesis in dialogue

Natural language generation



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

readabilty 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.

Correlation coefficent	Adequacy	Fluency
BLEU	0.388	-0.492

- 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.

Dialogue act	Delexicalised utterance
confirm(area=$V)	Would you like a restaurant in the $V?
confirm(food=$V)	Would you like a $V restaurant?
confirm(food=$V,area=$W)	Would you like a $V restaurant in the $W ?

Template-based natural language generator

Pros simple, usually error free, controllable

Cons time consuming, rigid, not scalable

Trainable generator [Walker et al., 2002]

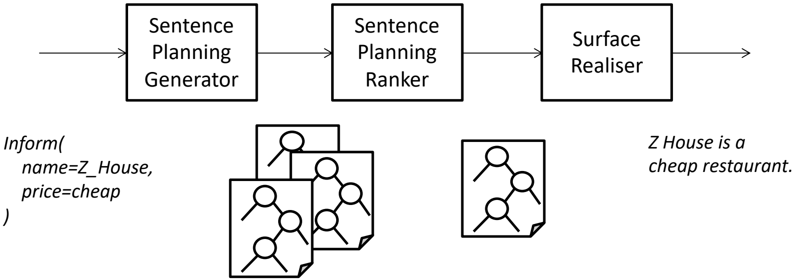
- Divide the problem into a pipeline

  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



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.

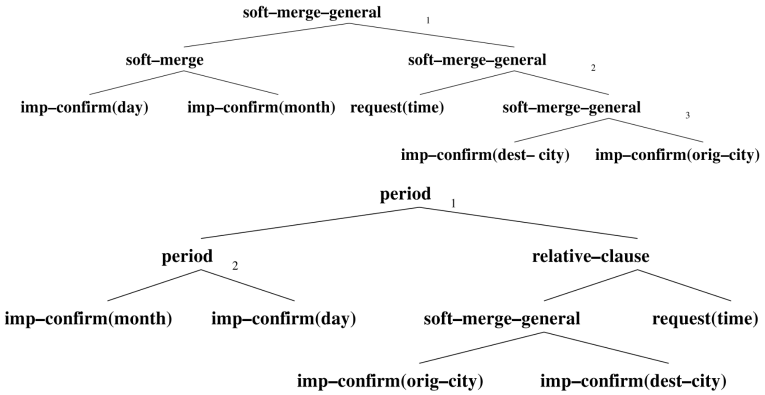
$$F(x) = \sum_i \alpha_i f_i(x)$$

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

where x and y are sentence plans, x is preferred to y, α_i are learnable parameters, and $\mathcal{D}$ 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)



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

$$p(W) = \prod_t p(w_t | w_0, \cdots, w_{t-1})$$

- Class-based language modelling

$$p(W|u) = \prod_t p(w_t | w_0, \cdots, w_{t-1}, u)$$

- 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]

$$
\begin{aligned}
\mathbf{h}_j &= \sigma(W_r \mathbf{h}_{j-1} + W_i \mathbf{w}_j + \mathbf{b}_h) \\
\mathbf{y}_j &= softmax(W_o \mathbf{h}_j + \mathbf{b}_o) \\
\frac{\partial E_3}{\partial W_r} &= \sum_{k=0}^{3} \frac{\partial E_3}{\partial \mathbf{y}_3} \frac{\partial \mathbf{y}_3}{\partial \mathbf{h}_3} \frac{\partial \mathbf{h}_3}{\partial \mathbf{h}_k} \frac{\partial \mathbf{h}_k}{\partial W_r} \\
&= \sum_{k=0}^{3} \frac{\partial E_3}{\partial \mathbf{y}_3} \frac{\partial \mathbf{y}_3}{\partial \mathbf{h}_3} \left(\prod_{j=k+1}^{3} \frac{\partial \mathbf{h}_j}{\partial \mathbf{h}_{j-1}} \right) \frac{\partial \mathbf{h}_k}{\partial W_r} \\
\frac{\partial \mathbf{h}_j}{\partial \mathbf{h}_{j-1}} &= W_r^{\mathrm{T}} diag(\sigma'(W_r \mathbf{h}_{j-1} + W_i \mathbf{w}_j + \mathbf{b}_h)),
\end{aligned}
$$

where $'$ is elementwise derivative. The norm of the last term is smaller than 1 causing vanishing gradient.

LSTM [Hochreiter and Schmidhuber, 1997]

- Sigmoid gates

$$
\begin{aligned}
\mathbf{i}_t &= \sigma(W_{wi}\mathbf{w}_t + W_{hi}\mathbf{h}_{t-1}) \\
\mathbf{f}_t &= \sigma(W_{wf}\mathbf{w}_t + W_{hf}\mathbf{h}_{t-1}) \\
\mathbf{o}_t &= \sigma(W_{wo}\mathbf{w}_t + W_{ho}\mathbf{h}_{t-1})
\end{aligned}
$$

- Proposed cell value

$$
\hat{C}_t = tanh(W_{wc}\mathbf{w}_t + W_{hc}\mathbf{h}_{t-1})
$$

- Update cell and hidden layer

$$
\begin{aligned}
C_t &= \mathbf{i}_t \odot \hat{C}_t + \mathbf{f}_t \odot C_{t-1} \\
\mathbf{h}_t &= \mathbf{o}_t \odot tanh(C_t)
\end{aligned}
$$

LSTM

How it prevents vanishing gradient problem?

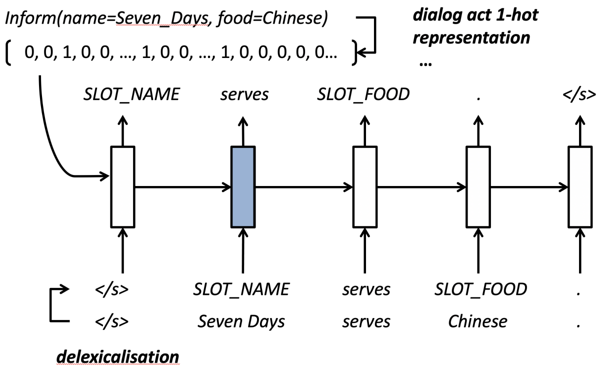
- Consider memory cell, where recurrence actually happens

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

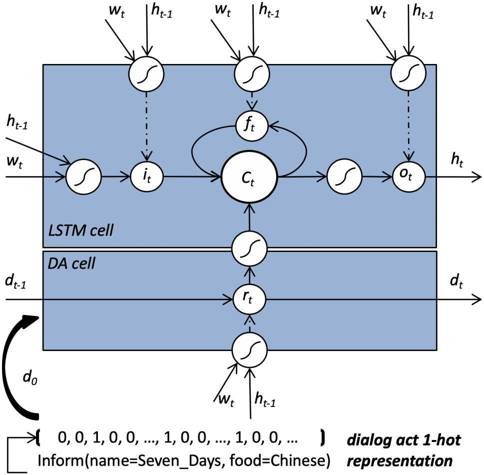
- We can back-propagate the gradient by chain rule

$$\frac{\partial E_t}{\partial C_{t-1}} = \frac{\partial E_t}{\partial C_t}\frac{\partial C_t}{\partial C_{t-1}} = \frac{\partial E_t}{\partial C_t}\mathbf{f}_t$$

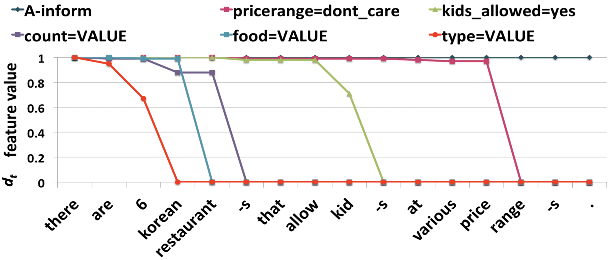
RNN for NLG [Wen et al., 2015a]



Semantically conditioned LSTM [Wen et al., 2015b]



Learned alignemnts



Human evaluation

Method	Informativeness	Naturalness
SC-LSTM	**2.59**	2.50
Class LM	2.46	2.45

Exampes

inform_no_match(area=tenderloin)
there are no restaurants in the tenderloin area.
there are 0 restaurants in the tenderloin area.
unfortunately there are 0 restaurants in the tenderloin area.
i could not find any restaurants in tenderloin.

Properties

Pros
: more accurate, does not require intermediate alignments

Cons
: does not utilise pre-trained word-vector embeddings

Generative pre-training (GPT)

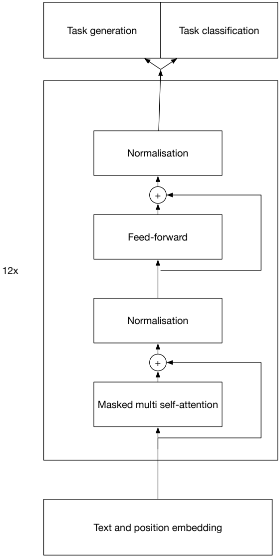
- Autoregressive language model that utilises transformer architecture
- Pretrained on large amounts of crawled text
- Predicts the next token u given context $U = [u_{-k}, \cdots, u_{-1}]$

$$\begin{aligned} \mathbf{h}_0 &= UW_e + W_p \\ \mathbf{h}_l &= transformer(\mathbf{h}_{l-1}), l \in [1, n] \\ p(u) &= softmax(\mathbf{h}_n W_e^T) \end{aligned}$$

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

	GPT	GPT-2	GPT-3
Parameters	117M	1.5B	175B
Data	12GB	40GB	570GB

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.

Human evaluation

Models trained on FewshotWOZ with only 50 dialogue acts in the training set and 500K in test set.

Method	Informativeness	Naturalness
SC-GPT	2.64	2.47
SC-LSTM	2.29	2.13
Human	2.92	2.72

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 I

Hochreiter, S. and Schmidhuber, J. (1997).
Long short-term memory.
Neural Computation, 9(8):1735–1780.

Oh, A. H. and Rudnicky, A. I. (2000).
Stochastic language generation for spoken dialogue systems.
In *ANLP-NAACL 2000 Workshop: Conversational Systems*.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002).
Bleu: a method for automatic evaluation of machine translation.
In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

References II

Pascanu, R., Mikolov, T., and Bengio, Y. (2013).
On the difficulty of training recurrent neural networks.
In Dasgupta, S. and McAllester, D., editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1310–1318, Atlanta, Georgia, USA. PMLR.

Stent, A., Marge, M., and Singhai, M. (2005).
Evaluating evaluation methods for generation in the presence of variation.
In Gelbukh, A., editor, *Computational Linguistics and Intelligent Text Processing*, pages 341–351, Berlin, Heidelberg. Springer Berlin Heidelberg.

References III

Walker, M., Rambow, O., and Rogati, M. (2002).
Training a sentence planner for spoken dialogue using boosting.
Computer Speech Language, 16:409–433.

Wen, T.-H., Gašić, M., Kim, D., Mrkšić, N., Su, P.-H., Vandyke, D., and Young, S. (2015a).
Stochastic Language Generation in Dialogue using Recurrent Neural Networks with Convolutional Sentence Reranking.
In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*.
Association for Computational Linguistics.

References IV

Wen, T.-H., Gašić, M., Mrkšić, N., Su, P.-H., Vandyke, D., and Young, S. (2015b).
Semantically conditioned LSTM-based natural language generation for spoken dialogue systems.
In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal. Association for Computational Linguistics.

Yu, K., Zen, H., Mairesse, F., and Young, S. (2010).
Context adaptive training with factorized decision trees for hmm-based speech synthesis.
In *Proceedings of Interspeech*.

Credits

We thank Tsung-Hsien Wen for sharing his slides on Stistical Natual Language Generation.