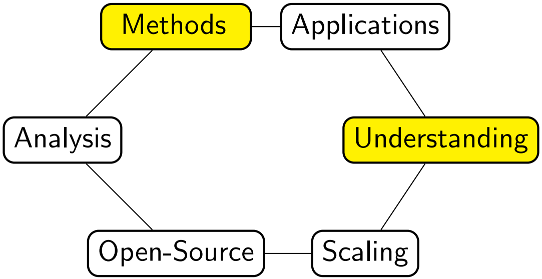
Learning How to Say It: Language Generation post Deep Learning

Alexander M Rush

Part 4: Deep Latent-Variable Mdoels



Deep Latent-Variable Models

Goal: Extent text generation to Expose specific choices as *discrete* latent variables.

$$p(y, z | x; \theta).$$

Deep Latent-Variable Models

Goal: Extent text generation to Expose specific choices as *discrete* latent variables.

$$p(y, z | x; \theta).$$

- y is our text output sequence
- z is a collection of latent variables
- θ are the neural network parameters.

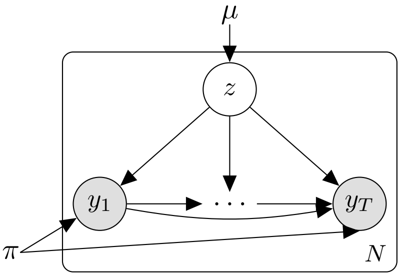
Example Model: Mixture of RNNs

Generative process:

1. Draw cluster $z \in \{1, \dots, K\}$ from a Categorical.
2. Draw words $y_{1:T}$ from RNNLM with parameters π_z.

$$p(y, z | x; \theta) = \mu_z \times \mathrm{RNNLM}(y_{1:T}; \pi_z)$$

j



Posterior Inference

We'll be interested in the *posterior* over latent variables z:

$$p(z \,|\, y, x;\, \theta) = \frac{p(y, z|x;\, \theta)}{p(y|x;\, \theta)} = \frac{p(y \,|\, x, z;\, \theta)p(z|x;\, \theta)}{\displaystyle\sum_{z'} p(y \,|\, x, z';\, \theta)p(z'|x;\, \theta)}.$$

Posterior Inference

We'll be interested in the *posterior* over latent variables z:

$$p(z \mid y, x;\, \theta) = \frac{p(y, z|x;\, \theta)}{p(y|x;\, \theta)} = \frac{p(y \mid x, z;\, \theta)p(z|x;\, \theta)}{\sum_{z'} p(y \mid x, z';\, \theta)p(z'|x;\, \theta)}.$$

How?

- Sum out over all discrete choices (e.g. run K RNNs).
- Variational inference based methods.

Application: Summary with Copy-Attention

(Gu et al, 2016) (Gulcehre et al, 2016)

Let z be a binary latent variable.

- If $z = 1$, let the model generate a new word.
- If $z = 0$, let the model copy a word from the source.

Inference:

Pointer-generator model + coverage summary

francis *saili* has signed a two-year deal to join munster later this year .
the 24-year-old was part of the new zealand under-20 side that won the junior world championship in italy in 2011 .
saili 's signature is something of a coup for munster and head coach anthony foley .

(See et al, 2017)

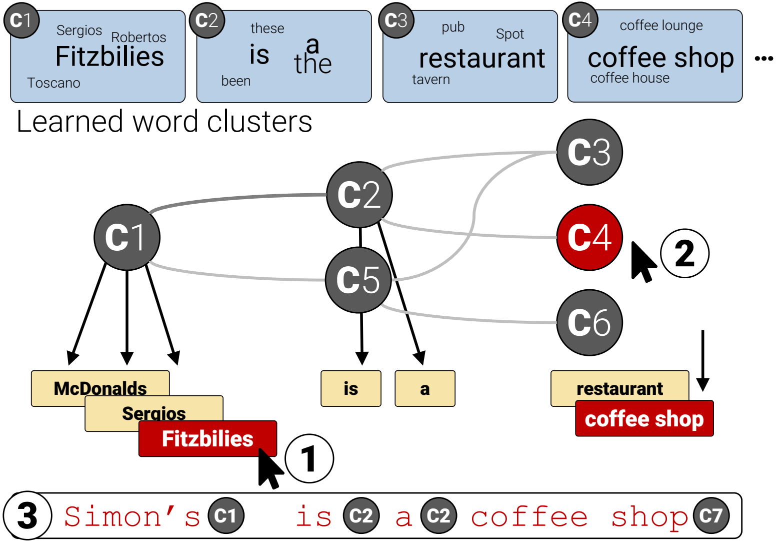
Latent Variable Models for Generation

Ongoing Work: Can we develop other discrete latent-variable models for generation?

Goals:

- Model Control
- Model Debugging
- Model Uncertainty

Example: Learning Neural Templates for Generation



MR	name[The Golden Palace], eatType[coffee shop], food[Fast food], priceRange[cheap], customer rating[5 out of 5], area[riverside]
Reference	A coffee shop located on the riverside called The Golden Palace, has a 5 out of 5 customer rating. Its price range are fairly cheap for its excellent Fast food.

Standard Approach

Step 1: Encode the Source

Fitzbillies,type[coffee shop],price[$<$ £20],food[Chinese],rate[3/5],area[city centre]

Step 2: Generate with RNN Decoder

Fitzbillies is a coffee shop providing Chinese food in the moderate price range . It is located in the city centre . Its customer rating is 3 out of 5.

Issues

1. Interpretable in its content selection?

   Decisions may come from anywhere in the source x.

2. Controllable in terms of style and form?

   Rely on a learned system to determine content.

Neural Template Generation Approach

Step 1: Encode the Source

Fitzbillies,ty[coffee shop],pr[$<$ £20],food[Chinese],cust[3/5],area[city centre]

Neural Template Generation Approach

Step 1: Encode the Source

Fitzbillies,ty[coffee shop],pr[< £20],food[Chinese],cust[3/5],area[city centre]

Step 2: Select a Template

The ___ ___ …	is a is an is an expensive …	___	providing serving offering …	___	food cuisine foods …	in the	high moderate less than average …	price price range …
.	It is	located in the located near near …	___	.	Its customer rating is Their customer rating is Customers have rated it …	___ out of ___	.	

Neural Template Generation Approach

Step 1: Encode the Source

Fitzbillies,ty[coffee shop],pr[$<$ £20],food[Chinese],cust[3/5],area[city centre]

Step 2: Select a Template

The ___ / ___ / ... | is a / is an / is an expensive / ... | ___ | providing / serving / offering / ... | ___ | food / cuisine / foods / ... | in the | high / moderate / less than average / ... | price / price range / ... |

. | It is | located in the / located near / near / ... | ___ | . | Its customer rating is / Their customer rating is / Customers have rated it / ... | ___ out of ___ | .

Step 3: Fill-in Each Segment

|| <u>Fitzbillies</u> ||

Neural Template Generation Approach

Step 1: Encode the Source

Fitzbillies,ty[coffee shop],pr[$<$ £20],food[Chinese],cust[3/5],area[city centre]

Step 2: Select a Template

| The ___
 ___
 ... | is a
 is an
 is an expensive
 ... | ___ | providing
 serving
 offering
 ... | ___ | food
 cuisine
 foods
 ... | in the | high
 moderate
 less than average
 ... | price
 price range
 ... |

. | It is | located in the
 located near
 near
 ... | ___ | . | Its customer rating is
 Their customer rating is
 Customers have rated it
 ... | ___ out of ___ | .

Step 3: Fill-in Each Segment

|| Fitzbillies || is a ||

Neural Template Generation Approach

Step 1: Encode the Source

Fitzbillies,ty[coffee shop],pr[$<$ £20],food[Chinese],cust[3/5],area[city centre]

Step 2: Select a Template

The ___ ___ ...	is a is an is an expensive ...	___	providing serving offering ...	___	food cuisine foods ...	in the	high moderate less than average ...	price price range ...

.	It is	located in the located near near ...	___	.	Its customer rating is Their customer rating is Customers have rated it ...	___ out of ___	.

Step 3: Fill-in Each Segment

|| Fitzbillies || is a || coffee shop ||

Neural Template Generation Approach

Step 1: Encode the Source

Fitzbillies,ty[coffee shop],pr[$<$ £20],food[Chinese],cust[3/5],area[city centre]

Step 2: Select a Template

The ___	is a	___	providing	___	food	in the	high	price
___	is an		serving		cuisine		moderate	price range
...	is an expensive		offering		foods		less than average	...
	...		...		...		...	

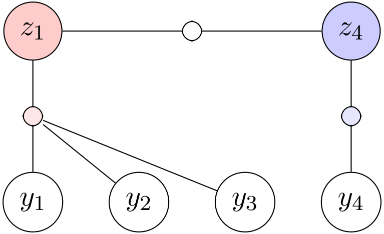
.	It is	located in the	___	.	Its customer rating is	___ out of ___	.
		located near			Their customer rating is		
		near			Customers have rated it		
		...			...		

Step 3: Fill-in Each Segment

|| Fitzbillies || is a || coffee shop || providing || Chinese || food || in the || moderate || price range || . || It is || located in the || city centre || . ||

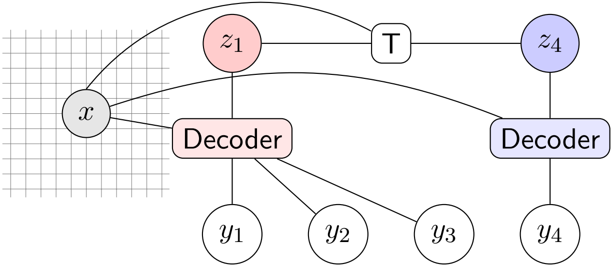
Technical Methodology: Hidden Semi-Markov Model

- HMM: discrete latent states with single emissions (e.g. words).
- HSMM: discrete latent states produce multiple emissions (e.g. phrases).
- Parameterized with *transition*, *emission*, and *length* distributions.



Technical Methodology: Neural Hidden Semi-Markov Model

- Employ HSMM as a conditional latent variable language model, $p(y_1, \ldots, y_T, z \mid x)$.
- Transition Distribution: NN between states.
- Emission Distribution: Seq2Seq+Attention, one per state k.



Technical Methodology: Learning Templates

- Fit model by maximizing log-marginal likelihood on training data.

$$\max_{\theta} \sum_{j} \log \sum_{z} p(y^{(j)}, z \mid x^{(j)}; \theta)$$

Details: Pre-score segmentations, HSMM forward algorithm for sum, backprop with autograd, all inference is exact.

Technical Methodology: Learning Templates

- Fit model by maximizing log-marginal likelihood on training data.

$$\max_{\theta} \sum_{j} \log \sum_{z} p(y^{(j)}, z \mid x^{(j)}; \theta)$$

Details: Pre-score segmentations, HSMM forward algorithm for sum, backprop with autograd, all inference is exact.

- Compute argmax segmentations to find common *templates*.

$$z^{(j)} = \arg\max_{z} p(y^{(j)}, z \mid x^{(j)}; \theta)$$

[The Wrestlers]$_{185}$ *[is a]*$_{29}$ *[coffee shop]*$_{164}$ *[that serves]*$_{188}$ *[English]*$_{139}$ *[food]*$_{18}$ *[in the]*$_{32}$ *[moderate]*$_{125}$ *[price range]*$_{180}$ *[.]*$_{90}$

Neural Template

The ___	is a	___	providing	___	food	in the	high	price
___	is an		serving		cuisine		moderate	price range
…	is an expensive		offering		foods		less than average	…
	…		…		…		…	

.	It is	located in the	___	.	Its customer rating is	___ out of ___	.
		located near			Their customer rating is		
		near			Customers have rated it		
		…			…		

E2E Challenge

	BLEU	NIST
Test		
Substitution	43.78	6.88
Neural Template	56.72	7.63
Full Neural Model	65.93	8.59

	BLEU	NIST	ROUGE-4
Conditional KN-LM	19.8	5.19	10.7
NNLM (field)	33.4	7.52	23.9
NNLM (field & word)	34.7	7.98	25.8
Neural Template	33.8	7.51	28.2

Issue 1: Interpretability

kenny warren

name: kenny warren, **birth date:** 1 april 1946,
birth name: kenneth warren deutscher, **birth place:** brooklyn, new york,
occupation: ventriloquist, comedian, author,
notable work: book - the revival of ventriloquism in america

1. kenny warren deutscher (april 1, 1946) is an american ventriloquist.
2. kenny warren deutscher (april 1, 1946 , brooklyn,) is an american ventriloquist.
3. kenny warren deutscher (april 1, 1946) is an american ventriloquist, best known for his the revival of ventriloquism.
4. “kenny” warren is an american ventriloquist.
5. kenneth warren “kenny” warren (born april 1, 1946) is an american ventriloquist, and author.

Issue 2: Controllability

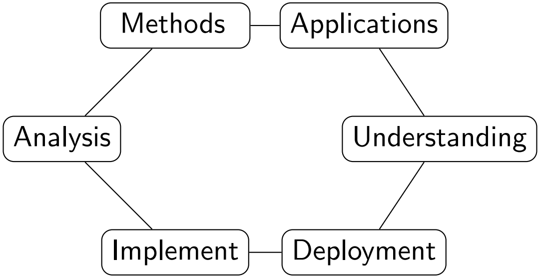
The Golden Palace

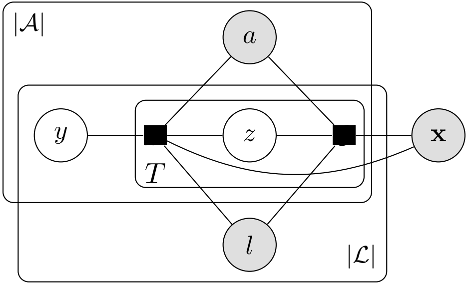
name[The Golden Palace], type[coffee shop], food[Chinese], priceRange[cheap] custRating[5 out of 5], area[city centre],

1. The Golden Palace is a coffee shop located in the city centre.
2. In the city centre is a cheap Chinese coffee shop called The Golden Palace.
3. The Golden Palace that serves Chinese food in the cheap price range. It is located in the city centre. Its customer rating is 5 out of 5.
4. The Golden Palace is a Chinese coffee shop.
5. The Golden Palace is a Chinese coffee shop with a customer rating of 5 out of 5.

Future Work

NLP post deep learning




$|\mathcal{A}|$
a
y
z
$\mathbf{x}$
T
l
$|\mathcal{L}|$

Reasoning-Based Models

Temporary

Long-Form Generation with Reasoning

Yuntian Deng, Anssi Kanervisto, and Alexander M. Rush. 2016. What You Get Is What You See: A Visual Markup Decompiler. In *Arxiv*.

Yuntian Deng, Yoon Kim, Justin Chiu, Demi Guo, and Alexander Rush. 2018. Latent alignment and variational attention. In *Advances in Neural Information Processing Systems*, pages 9735–9747.

Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. 2017. Structured attention networks. abs/1702.00887.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-Aware Neural Language Models. In *AAAI*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-Level Knowledge Distillation. In *EMNLP*.

Yoon Kim, Sam Wiseman, Andrew C. Miller, David Sontag, and Alexander M. Rush. 2018. Semi-amortized variational autoencoders.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017.

Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, System Demonstrations*, pages 67–72.

Brandon Reagen, Udit Gupta, Robert Adolf, Michael M Mitzenmacher, Alexander M Rush, Gu-Yeon Wei, and David Brooks. 2017. Weightless: Lossy weight encoding for deep neural network compression. *arXiv preprint arXiv:1711.04686*.

Alexander Rush. 2018. The annotated transformer. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 52–60.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *EMNLP*, September, pages 379–389.

Allen Schmaltz, Yoon Kim, Alexander M. Rush, and Stuart M. Shieber. 2016. Sentence-Level Grammatical Error Identification as Sequence-to-Sequence Correction. In *arxiv*.

Jean Senellart, Dakun Zhang, WANG Bo, Guillaume Klein, Jean-Pierre Ramatchandirin, Josep Crego, and Alexander Rush. 2018. Opennmt system description for wnmt 2018: 800 words/sec on a single-core cpu. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 122–128.

Hendrik Strobelt, Sebastian Gehrmann, Michael Behrisch, Adam Perer, Hanspeter Pfister, and Alexander M Rush. 2019. Seq2seq-v is: A visual debugging tool for sequence-to-sequence models. *IEEE transactions on visualization and computer graphics*, 25(1):353–363.

Hendrik Strobelt, Sebastian Gehrmann, Bernd Huber, Hanspeter Pfister, and Alexander M. Rush. 2016. Visual Analysis of Hidden State Dynamics in Recurrent Neural Networks. In *Arxiv*.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2017a. Challenges in Data-to-Document Generation. In *EMNLP*.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2017b. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical*

Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017, pages 2253–2263.

Sam Wiseman, Stuart M Shieber, and Alexander M Rush. 2018. Learning neural templates for text generation. *arXiv preprint arXiv:1808.10122*.