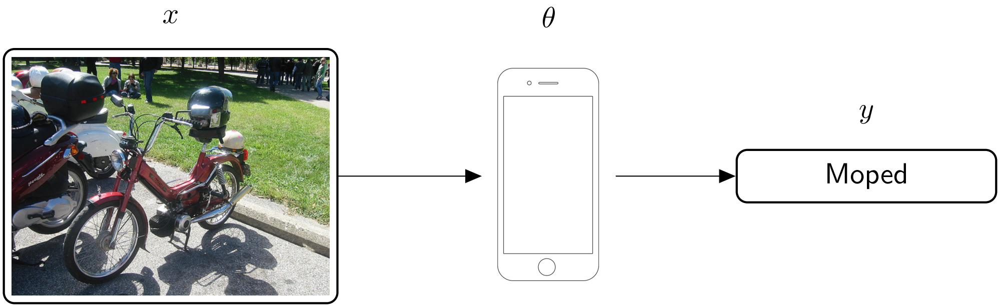
Learning How to Say It: Language Generation and Deep Learning

Alexander M Rush

Machine Learning for Multiclass Classification

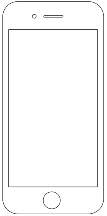


Machine Learning for Text Generation: Translation

x

Yalitza Aparicio acababa de graduarse de una escuela para maestros y aun no tenia empleo cuando el proceso de busqueda de actrices para la ultima pelicula de Alfonso Cuaron llego a su natal Tlaxiaco, Oaxaca.

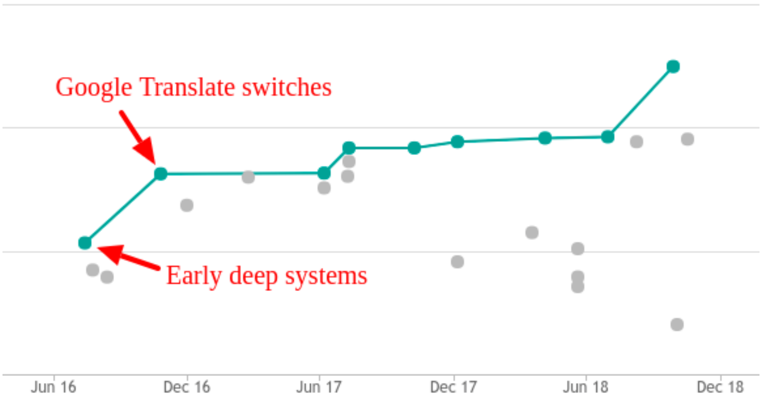
θ



$y_{1:T}$

Yalitza Aparicio had just finished her teaching degree and didn't yet have a job when the Mexican director Alfonso Cuaron held a casting call in her home of Tlaxiaco, Oaxaca, for the lead role in his semi-autobiographical drama, "Roma."

Translation Performance



Machine Learning for Text Generation

$$y^*_{1:T} = \arg\max_{y_{1:T}} f(y_{1:T}, x; \theta)$$

Machine Learning for Text Generation

$$y^*_{1:T} = \arg\max_{y_{1:T}} f(y_{1:T}, \textcolor{red}{x}; \theta)$$

- Input $\textcolor{red}{x}$, *what to talk about*

Machine Learning for Text Generation

$$y^*_{1:T} = \arg\max_{y_{1:T}} f(y_{1:T}, x; \theta)$$

- Input x, *what to talk about*
- Output text $y^*_{1:T}$, *how to say it*

Machine Learning for Text Generation

$$y^*_{1:T} = \arg\max_{y_{1:T}} {\color{red} f}(y_{1:T}, x; {\color{red}\theta})$$

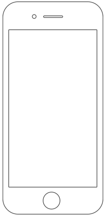
- Input x, *what to talk about*
- Output text $y^*_{1:T}$, *how to say it*
- Model ${\color{red} f(.;\theta)}$, learned from data

Sentence Summarization

x

Cambodian leader Hun Sen on friday rejected opposition parties' demands for talks outside the country, accusing them of trying to "internationalize" the political crisis.

θ

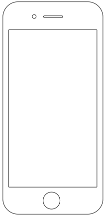


Sentence Summarization

x

Cambodian leader Hun Sen on friday rejected opposition parties' demands for talks outside the country, accusing them of trying to "internationalize" the political crisis.

θ



$y_{1:T}$

Cambodian government rejects opposition's call for talks abroad

GigaWord Dataset

(Rush et al. [2015] w/ Facebook)

Sep 13, 3:17 PM EDT

GERMANY IMPLEMENTS TEMPORARY BORDER CHECKS TO LIMIT MIGRANTS

BY GEIR MOULSON AND SHAWN POGATCHNIK
ASSOCIATED PRESS

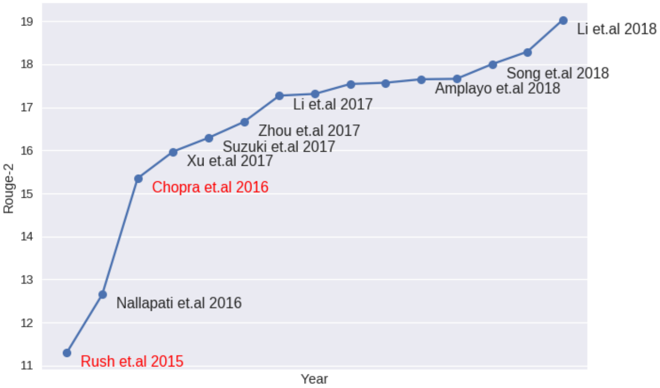
BERLIN (AP) -- Germany introduced temporary border controls Sunday to stem the tide of thousands of refugees streaming across its frontier, sending a clear message to its European partners that it needs more help with an influx that is straining its ability to cope.

AP Photo/Kay Nietfeld

Germany is a preferred destination for many people fleeing Syria's civil war and other troubled nations in the migration crisis that has bitterly divided Europe. They have braved dangerous sea crossings in flimsy

- Several million headlines paired with article leads.
- Model for abstractive summarization / compression.

Sentence Summarization



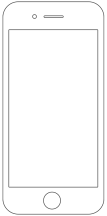
Target: `[Yalitza Aparicio had] just [finished her] teaching [degree] .`

Predict: `[Yalitza Aparicio had] recently [finished her] [degree].`

Talk about Data (Wiseman et al. [2017a])

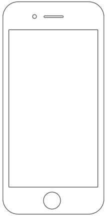
TEAM	WIN	LOSS	PTS	FG_PCT	RB	AS ...
Heat	11	12	103	49	47	27
Hawks	7	15	95	43	33	20

PLAYER	AS	RB	PT	FG	FGA	CITY ...
Tyler Johnson	5	2	27	8	16	Miami
Dwight Howard	11	17	23	9	11	Atlanta
Paul Millsap	2	9	21	8	12	Atlanta
Goran Dragic	4	2	21	8	17	Miami
Wayne Ellington	2	3	19	7	15	Miami
Dennis Schroder	7	4	17	8	15	Atlanta
Rodney McGruder	5	5	11	3	8	Miami
...						



The Atlanta Hawks defeated the Miami Heat, 103 - 95, at Philips Arena on Wednesday. Atlanta was in desperate need of a win and they were able to take care of a short-handed Miami team here. Defense was key for the Hawks, as they held the Heat to 42 percent shooting and forced them to commit 16 turnovers. Atlanta also dominated in the paint, winning the rebounding battle, 47 - 34, and outscoring them in the paint 58 - 26. The Hawks shot 49 percent from the field and assisted on 27 of their 43 made baskets. This was a near wire-to-wire win for the Hawks, as Miami held just one lead in the first five minutes. Miami (7 - 15) are as beat-up as anyone right now and it's taking a toll on the heavily used starters. Hassan Whiteside really struggled in this game, as he amassed eight points, 12 rebounds and one blocks on 4 - of - 12 shooting ...

$$\mathcal{K}^{L}(\sigma=2)=\left(\begin{array}{cc}-\frac{d^{2}}{dx^{2}}+4-\frac{3}{\cosh^{2}x} & \frac{3}{\cosh^{2}x}\\ \frac{3}{\cosh^{2}x} & -\frac{d^{2}}{dx^{2}}+4-\frac{3}{\cosh^{2}x}\end{array}\right)\qquad,$$



```
{ \cal K } ^ { L } ( \sigma = 2 ) = \left( \begin{array}
{ c c } { - \frac { d ^ { 2 } } { d x ^ { 2 } } + 4 - \frac
{ 3 } { \operatorname { c o s h } ^ { 2 } x } } \& { \frac
{ 3 } { d x ^ { 2 } } }  { \frac { 3 } { \operatorname
{ c o s h } ^ { 2 } x } } \& { - \frac { d ^ { 2 } }
{ d x ^ { 2 } } + 4 - \frac { 3 } { \operatorname { c o s h }
^ { 2 } x } }  \end{array} \right) \qquad
```

Temporary

Talk Outline

Goal

Learn How to Say It

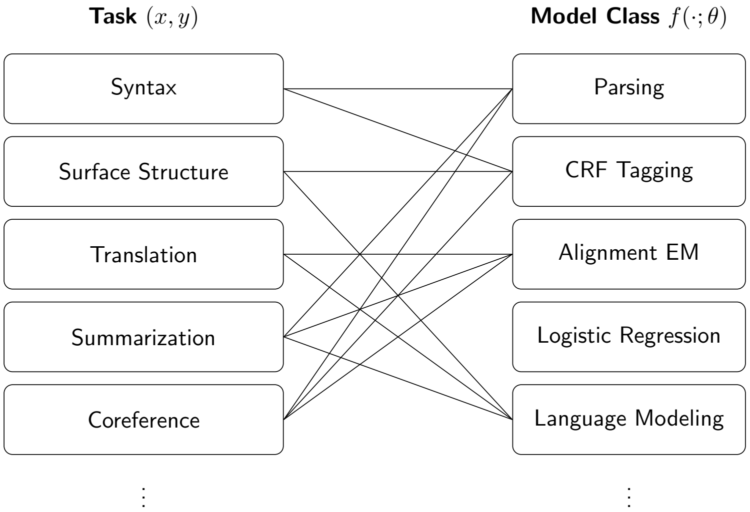
Talk Outline

Goal

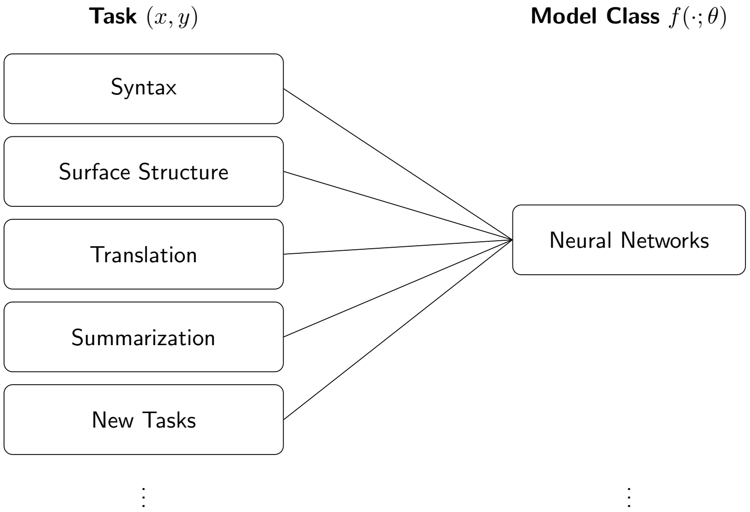
Learn How to Say It

- Background: Core Model and Implementation
- Work 1: Rethinking Model Training (*Beam Search Optimization*)
- Work 2: Rethinking Generation (*Learning Neural Templates*)
- Future Directions

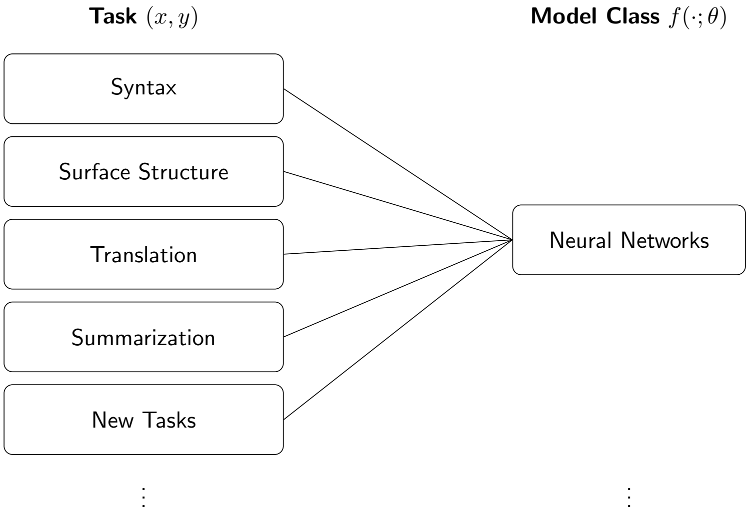
State-of-the-Art Natural Language Processing, circa 2009



State-of-the-Art Natural Language Processing, circa 2019



State-of-the-Art Natural Language Processing, circa 2019



State-of-the-Art Natural Language Processing, circa 2019

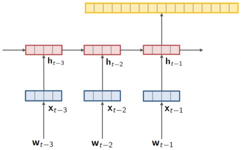
Task (x, y)

- Syntax
- Surface Structure
- Translation
- Summarization
- New Tasks

$\vdots$

Model Class $f(\cdot; \theta)$

Neural Networks



$\vdots$

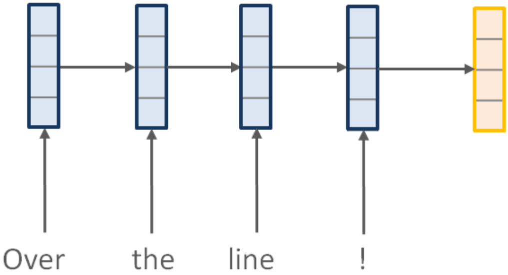
Recurrent Neural Network

$f(y_{1:T}, x_{1:S}; \theta)$

Over the line !

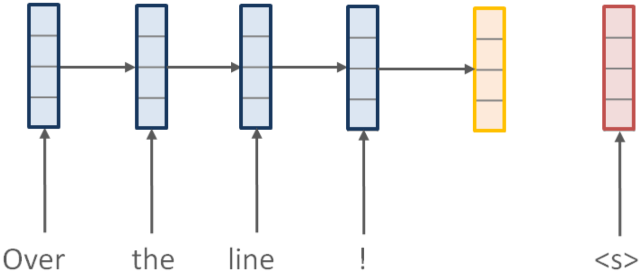
Recurrent Neural Network

$$f(y_{1:T}, x_{1:S}; \theta)$$



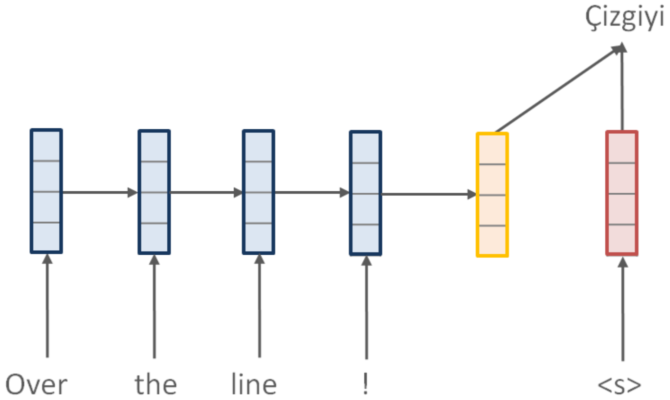
Recurrent Neural Network

$$f(y_{1:T}, x_{1:S}; \theta)$$



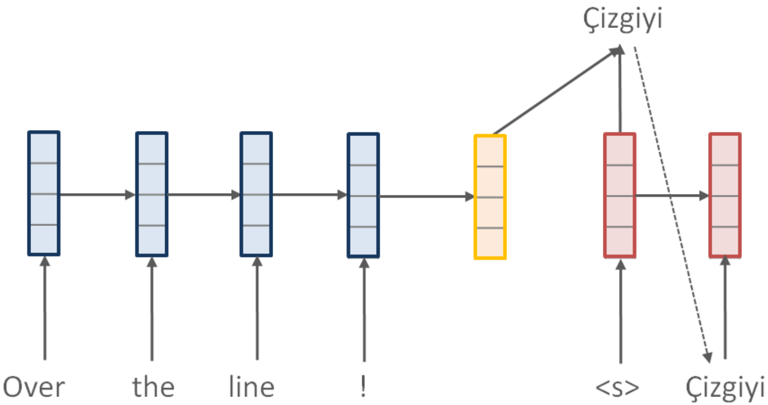
Recurrent Neural Network

$f(y_{1:T}, x_{1:S}; \theta)$



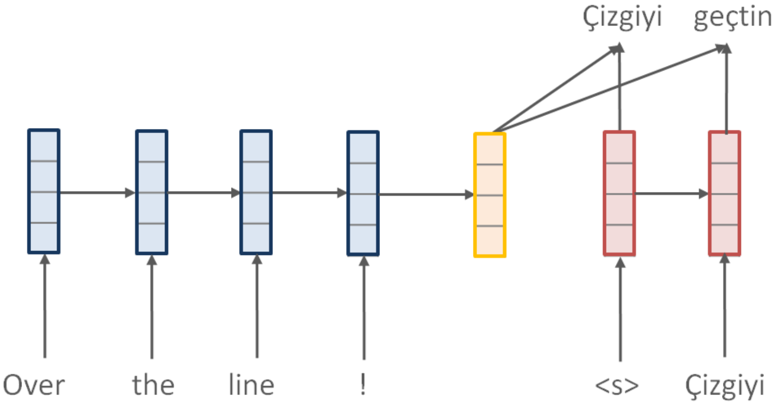
Recurrent Neural Network

$f(y_{1:T}, x_{1:S}; \theta)$



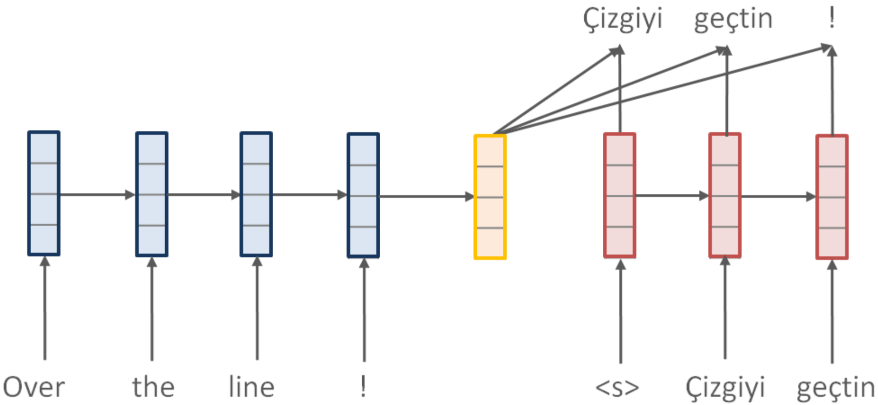
Recurrent Neural Network

$$f(y_{1:T}, x_{1:S}; \theta)$$



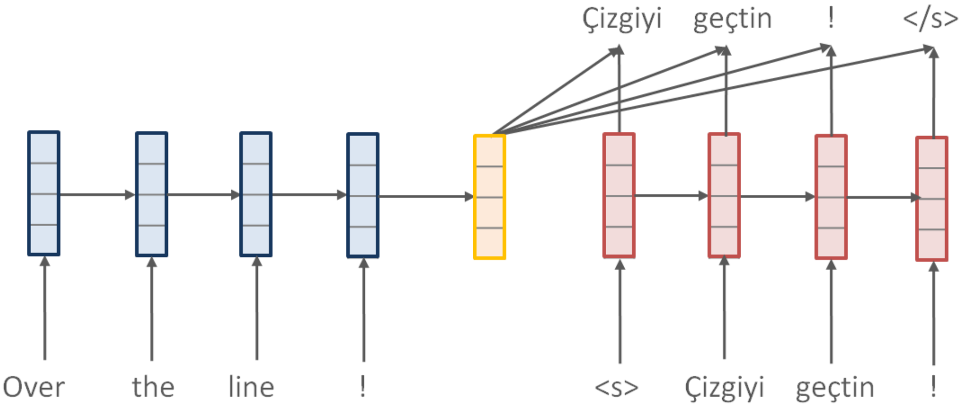
Recurrent Neural Network

$f(y_{1:T}, x_{1:S}; \theta)$



Recurrent Neural Network

$f(y_{1:T}, x_{1:S}; \theta)$



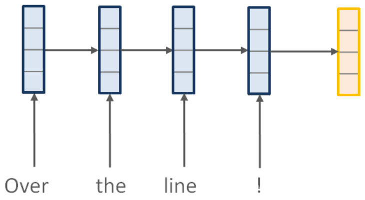
Recurrent Neural Network Math

Encoder:

$$\mathbf{h}_s^x \leftarrow \mathrm{RNN}(\mathbf{h}_{s-1}^x, x_s)$$

Context:

$$\mathbf{c} = \mathbf{h}_S^x$$



Recurrent Neural Network Math

Encoder:

$$\mathbf{h}_s^x \leftarrow \mathrm{RNN}(\mathbf{h}_{s-1}^x, x_s)$$

Context:

$$\mathbf{c} = \mathbf{h}_S^x$$

Decoder:

$$\mathbf{h}_t \leftarrow \mathrm{RNN}(\mathbf{h}_{t-1}, y_t)$$

Prediction:

$$p(y_t \mid y_{1:t-1}, x) = \mathrm{softmax}(\mathbf{W}[\mathbf{h}_{t-1}; \mathbf{c}])$$

Recurrent Neural Network Math

Encoder:

$$\mathbf{h}_s^x \leftarrow \mathrm{RNN}(\mathbf{h}_{s-1}^x, x_s)$$

Context:

$$\mathbf{c} = \mathbf{h}_S^x$$

Decoder:

$$\mathbf{h}_t \leftarrow \mathrm{RNN}(\mathbf{h}_{t-1}, y_t)$$

Prediction:

$$p(y_t \mid y_{1:t-1}, x) = \mathrm{softmax}(\mathbf{W}[\mathbf{h}_{t-1}; \mathbf{c}])$$

Generation Score:

$$f(y_{1:T}, x; \theta) = \log \sum_{t=1}^{T} p(y_t \mid y_{1:t-1}, x)$$

Decoder Example

Decoder:

$$\mathbf{h}_t \leftarrow \mathrm{RNN}(\mathbf{h}_{t-1}, y_t)$$

Example (Dyck-1 Language):

Well-balanced parenthesis language with random nesting-level indicators,

- Vocabulary: ( ) 0 1 2 3 4
- Example String: 0 ((2) (((4 4 4) 3) . . .

LSTMVis - Parenthesis Language (Strobelt et al. [2016] w/ IBM)

Temporary

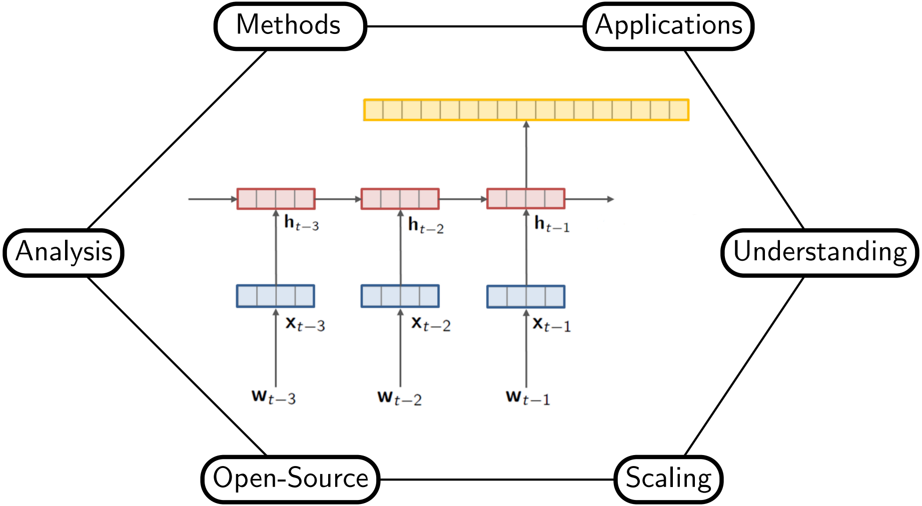
LSTMVis - Parenthesis Language (Strobelt et al. [2016] w/ IBM)

Temporary

LSTMVis - Natural Language (Strobelt et al. [2016] w/ IBM)

Temporary

Harvard NLP Deep Learning Research



Selected Harvard NLP Deep Learning Research

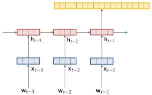
Kim et al. [2016]
Kim et al. [2017]
Wiseman et al. [2017a]

Rush et al. [2015]
Deng et al. [2016]
Schmaltz et al. [2016]

Methods — Applications

Strobelt et al. [2016]
Strobelt et al. [2019]
Wiseman et al. [2017b]

Analysis



Understanding

Wiseman et al. [2018]
Deng et al. [2018]
Kim et al. [2018]

Open-Source — Scaling

Klein et al. [2017]
Senellart et al. [2018]
Rush [2018]

Kim and Rush [2016]
Senellart et al. [2018]
Reagen et al. [2017]

Natural Lang. Processing
Machine Learning
Visualization

Talk Outline

1. Background: Core Model and Implementation
2. Work 1: Rethinking Model Training (*Beam Search Optimization*)
3. Work 2: Rethinking Generation (*Learning Neural Templates*)
4. Future Directions

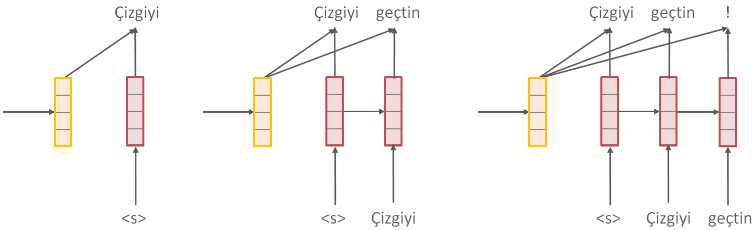
Machine Learning for Text Generation

$$y^*_{1:T} = \arg\max_{y_{1:T}} f(y_{1:T}; x, \theta)$$

- Input $x_{1:S}$, *what to talk about*
- Output text $y^*_{1:T}$, *how to say it*
- Model $f(.;\theta)$, learned from data

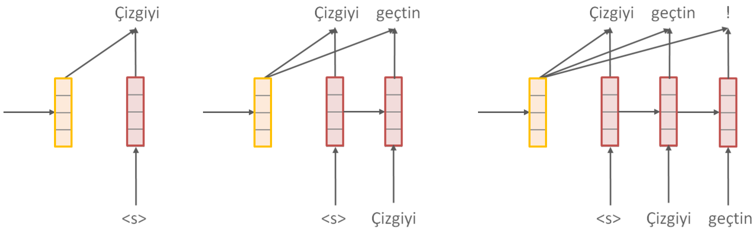
Training RNN

Parameters θ are trained to predict the next word *given the true history.*



Training RNN

Parameters θ are trained to predict the next word *given the true history.*

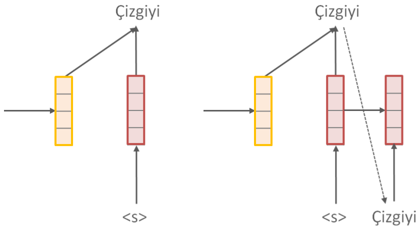


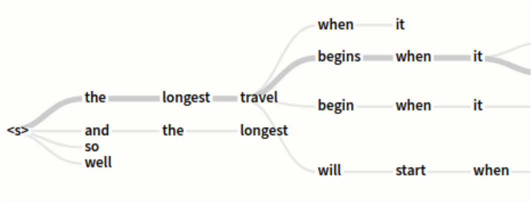
Objective is identical to multiclass classification.

$$\mathcal{L}(\theta) = -\sum_t \log p(y_t | y_{1:t-1}, x; \theta)$$

Generating with RNN

Parameters θ are deployed to predict the next word *given the predicted history*.





Requires predicting best sequence

$$y^*_{1:T} = \arg\max_{y_{1:T}} f(y_{1:T};\theta) = \arg\max_{y_{1:T}} \sum_t \log p(y_t|y_{1:t-1}, \mathbf{c};\theta)$$

But, intractable to solve exactly $O(\#\text{vocab}^T)$

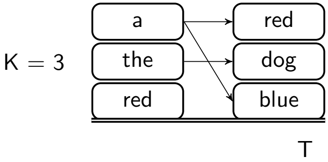
Standard Heuristic Method: Beam Search

$K = 3$

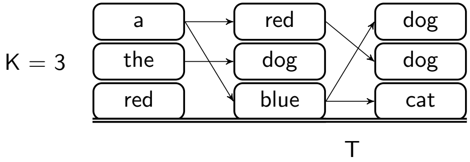
a
the
red

T

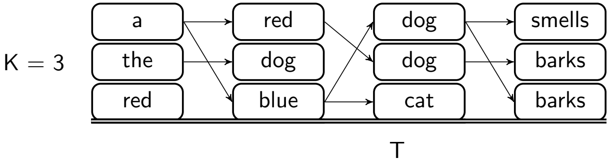
Standard Heuristic Method: Beam Search



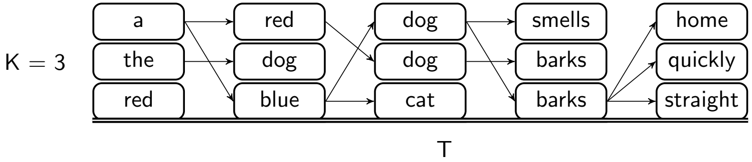
Standard Heuristic Method: Beam Search



Standard Heuristic Method: Beam Search

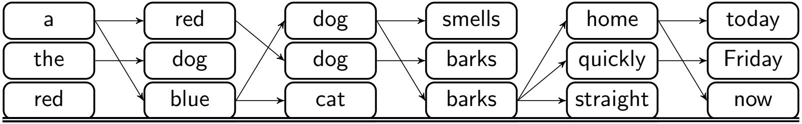


Standard Heuristic Method: Beam Search



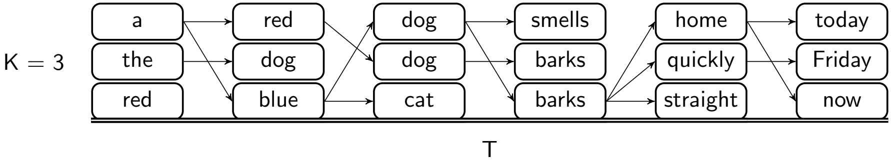
Standard Heuristic Method: Beam Search

$K = 3$



T

Standard Heuristic Method: Beam Search



1. Compute the score of every possible next word.

$$f(y_t, y_{1:t-1}^{(k)}) \leftarrow \log p(y_t \mid y_{1:t-1}^{(k)}, x) + \log p(y_{1:t-1}^{(k)} \mid x)$$

2. Prune to only the K highest-scoring,

$$y_{1:t}^{(1:K)} \leftarrow K \operatorname*{arg\,max}_{y_{1:t}} f(y_t, y_{1:t-1}^{(k)})$$

Theoretical Issues with Multiclass Training

1. Label Bias
   - Training is locally discriminative, but prediction is over entire sequences.

Theoretical Issues with Multiclass Training

1. Label Bias
   - Training is locally discriminative, but prediction is over entire sequences.
2. Exposure Bias
   - Training conditions on true history ($y_{1:t-1}$) but generates with predicted history.

Theoretical Issues with Multiclass Training

1. Label Bias
   - Training is locally discriminative, but prediction is over entire sequences.
2. Exposure Bias
   - Training conditions on true history ($y_{1:t-1}$) but generates with predicted history.
3. Metric Bias
   - Training uses multiclass classification, but evaluation uses n-gram match.

Sequence-to-Sequence Learning as Beam Search Optimization

Can we better model discrete sequences for text generation?

1. Label Bias
2. Exposure Bias
3. Metric Bias

Applications:

- (1) Improvements in training with less supervision.
- (2) Effective methods for downscaling translation models.

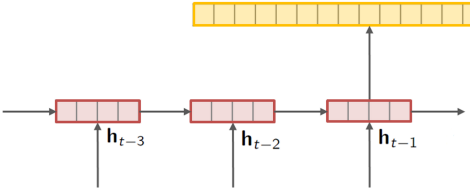
Modification 1: Global Scoring Function

Issue: Label Bias

- Training is locally discriminative, but prediction is over entire sequences.

Proposed Fix:

- Replace $\log p(y_t | y_{1:t-1}^{(k)}, \mathbf{c}; \theta)$ with a directly learned function $f(y_t, y_{1:t-1}^{(k)}, x; \theta)$



Modification 2: Beam Search at Training

Issue: Exposure Bias

- Training conditions on true history $(y_{1:t-1})$ but generates with predicted history.

Proposed Fix: During training:

1. Compute the score of every possible next word.

$$f(y_t, y_{1:t-1}^{(k)}) \leftarrow \log p(y_t \mid y_{1:t-1}^{(k)}, x) + \log p(y_{1:t-1}^{(k)} \mid x)$$

2. Prune to only the K highest-scoring,

$$y_{1:t}^{(1:K)} \leftarrow K \operatorname*{arg\,max}_{y_{1:t}} f(y_t, y_{1:t-1}^{(k)})$$

Modification 2: Beam Search at Training

Issue: Exposure Bias

- Training conditions on true history ($y_{1:t-1}$) but generates with predicted history.

Proposed Fix: During training:

1. Compute the score of every possible next word.

$$f(y_t, y_{1:t-1}^{(k)}) \leftarrow f(y_t, y_{1:t-1}^{(k)}, x; \theta)$$

2. Prune to only the K highest-scoring,

$$y_{1:t}^{(1:K)} \leftarrow K \arg\max_{y_{1:t}} f(y_t, y_{1:t-1}^{(k)})$$

Modification 3: Train with Margin

Issue: Metric Bias

- Training uses multiclass classification, but evaluation uses n-gram match.

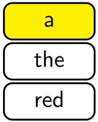
Proposed Fix: Use a structured SVM-style training loss:

- Margin between ground truth sequence $\hat{y}$ and worst predicted sequence $y^{(K)}$

$$\mathcal{L}(\theta) = \sum_t \Delta(\hat{y}_{1:t}, y^K_{1:t}) \left[1 - f(\hat{y}_t, y^{(g)}_{1:t-1}, x) + f(y^{(K)}_t, y^{(K)}_{1:t-1}, x) \right]$$

- Slack-rescaled, margin-based sequence criterion, at each time step.
- Δ is a task specific sequence cost, i.e. ngram-mismatch

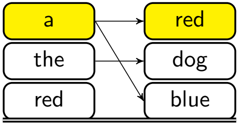
Beam Search Optimization Example



- True: ground-truth sequence $\hat{y}$
- Predicted: lowest-scoring prefix $y^{(K)}$

Strategy: if true falls off beam, restart from ground truth (learning as search optimization **?**)

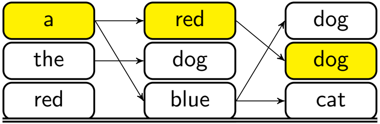
Beam Search Optimization Example



- True: ground-truth sequence $\hat{y}$
- Predicted: lowest-scoring prefix $y^{(K)}$

Strategy: if true falls off beam, restart from ground truth (learning as search optimization **?**)

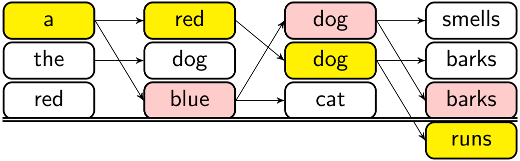
Beam Search Optimization Example



- True: ground-truth sequence $\hat{y}$
- Predicted: lowest-scoring prefix $y^{(K)}$

Strategy: if true falls off beam, restart from ground truth (learning as search optimization **?**)

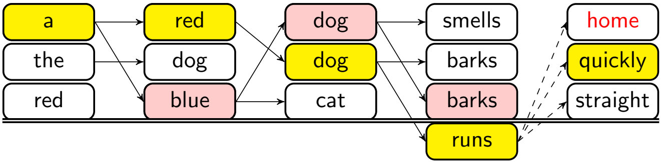
Beam Search Optimization Example



- True: ground-truth sequence $\hat{y}$
- Predicted: lowest-scoring prefix $y^{(K)}$

Strategy: if true falls off beam, restart from ground truth (learning as search optimization **?**)

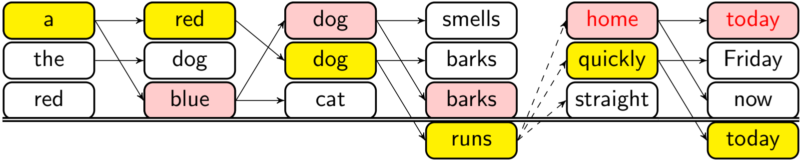
Beam Search Optimization Example



- True: ground-truth sequence $\hat{y}$
- Predicted: lowest-scoring prefix $y^{(K)}$

Strategy: if true falls off beam, restart from ground truth (learning as search optimization **?**)

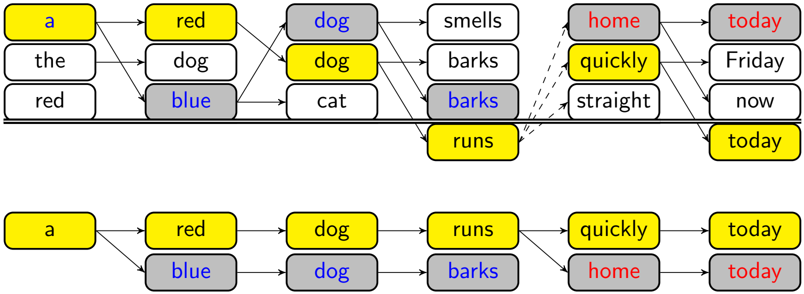
Beam Search Optimization Example



- True: ground-truth sequence $\hat{y}$
- Predicted: lowest-scoring prefix $y^{(K)}$

Strategy: if true falls off beam, restart from ground truth (learning as search optimization **?**)

Parameter Updates: Structured Backpropagation



- Margin gradients are sparse, only violating sequences get updates.
- Backprop as efficient as standard models, avoid exponential sum.

Results

Train Beam	$K = 1$	$K = 5$	$K = 10$
	Word Ordering (BLEU)		
seq2seq	25.2	29.8	31.0
BSO	28.0	33.2	34.3
BSO-Con	**28.6**	**34.3**	**34.5**

Results

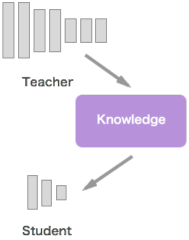
Train Beam	$K = 1$	$K = 5$	$K = 10$
	Word Ordering (BLEU)		
seq2seq	25.2	29.8	31.0
BSO	28.0	33.2	34.3
BSO-Con	**28.6**	**34.3**	**34.5**

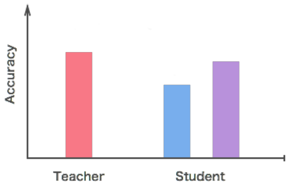
Results

Train Beam	$K = 1$	$K = 5$	$K = 10$
	Word Ordering (BLEU)		
seq2seq	25.2	29.8	31.0
BSO	28.0	33.2	34.3
BSO-Con	**28.6**	**34.3**	**34.5**
	Machine Translation (BLEU)		
seq2seq	22.53	24.03	23.87
BSO, SB-Δ	**23.83**	**26.36**	**25.48**
XENT	17.74	$\leq$ 20.5	$\leq$ 20.5
DAD	20.12	$\leq$ 22.5	$\leq$ 23.0
MIXER	20.73	-	$\leq$ 22.0

Goal: Shrink the size of text generation models.

- **Knowledge Distillation**: Train a *student* model to learn from a *teacher* model **???**.



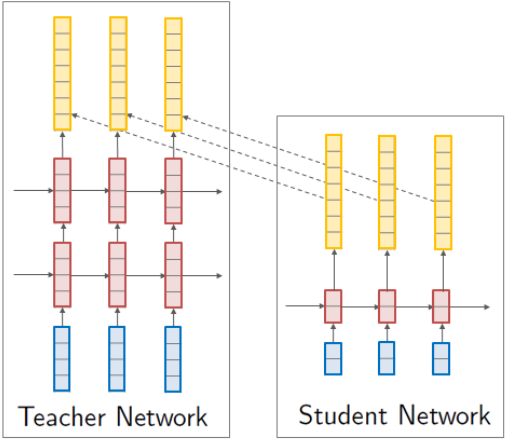


Multiclass Style: Word-Level Knowledge Distillation

Teacher model: $q(y_t | y_{1:t-1}, x; \theta_T)$

Cross-entropy between teacher and student

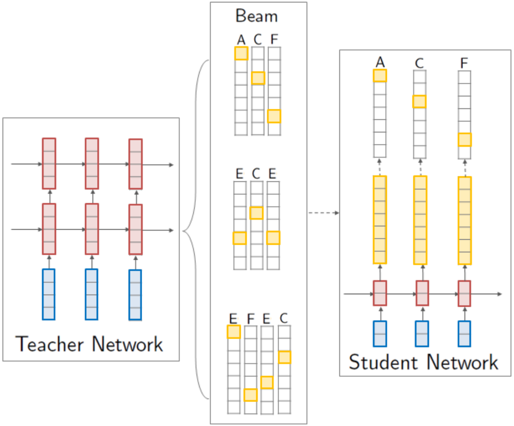
$$\mathcal{L}_{\text{WORD-KD}}(\theta) = -\sum_t \sum_v q(y_t = v \mid \hat{y}_{1:t-1}, x; \theta_T) \times$$
$$\log p(y_t = v \mid \hat{y}_{1:t-1}, x; \theta)$$



Sequence-Level Knowledge Distillation

$$\mathcal{L}_{\text{SEQ-KD}}(\theta) = -\log p(y^*_{1:T} \mid x; \theta)$$
$$\approx -\sum_{v_{1:T}} q(y_{1:T} = v_{1:t} | x; \theta_T) \log p(y_{1:T} | x; \theta)$$

Extension: $\mathcal{L}_{\text{SEQ-INTER}}(\theta)$ select sample based on ground truth $\hat{y}$ as well.



Results: WMT English $\rightarrow$ German Translation

Model	BLEU$_{K=1}$	$\Delta_{K=1}$	BLEU$_{K=5}$	$\Delta_{K=5}$
4×1000				
Teacher	17.7	$-$	19.5	$-$

Results: WMT English $\rightarrow$ German Translation

Model	BLEU$_{K=1}$	$\Delta_{K=1}$	BLEU$_{K=5}$	$\Delta_{K=5}$
4×1000				
Teacher	17.7	$-$	19.5	$-$
2×500				
Student	14.7	$-$	17.6	$-$
Word-KD	15.4	$+0.7$	17.7	$+0.1$

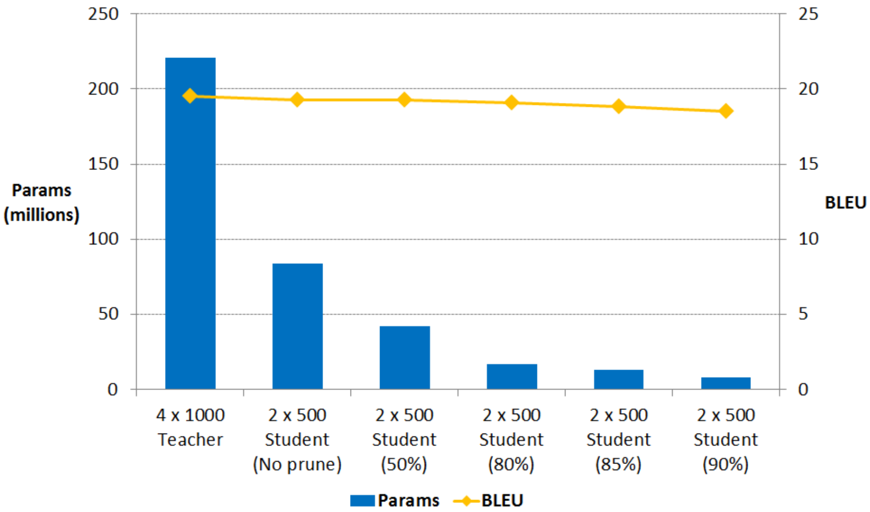
Results: WMT English $\rightarrow$ German Translation

Model	BLEU$_{K=1}$	$\Delta_{K=1}$	BLEU$_{K=5}$	$\Delta_{K=5}$
4×1000				
Teacher	17.7	–	19.5	–
2×500				
Student	14.7	–	17.6	–
Word-KD	15.4	+0.7	17.7	+0.1
Seq-KD	18.9	**+4.2**	19.0	+1.4
Seq-Inter	18.9	**+4.2**	19.3	**+1.7**

Results: WMT English $\rightarrow$ German Translation

Model	$\text{BLEU}_{K=1}$	$\Delta_{K=1}$	$\text{BLEU}_{K=5}$	$\Delta_{K=5}$
4×1000				
Teacher	17.7	–	19.5	–
2×500				
Student	14.7	–	17.6	–
Word-KD	15.4	+0.7	17.7	+0.1
Seq-KD	18.9	**+4.2**	19.0	+1.4
Seq-Inter	18.9	**+4.2**	19.3	**+1.7**
4×1000				
Seq-Inter	19.6	+1.9	19.8	+0.3

Combining Knowledge Distillation and Pruning



Application

Temporary

Talk Outline

1. Background: Core Model and Implementation
2. Work 1: Rethinking Model Training (*Beam Search Optimization*)
3. Work 2: Rethinking Generation (*Learning Neural Templates*)
4. Future Directions

Deep Latent-Variable Models

Goal: Expose specific choices as *discrete* latent variables.

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

Deep Latent-Variable Models

Goal: 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.

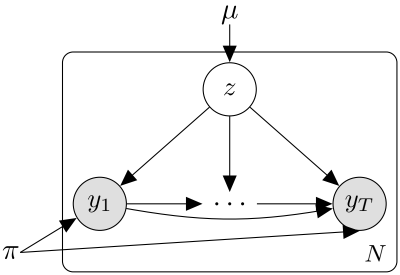
Example Model: Mixture of RNNs

Generative process:

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

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

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

Let z be a binary latent variable.

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

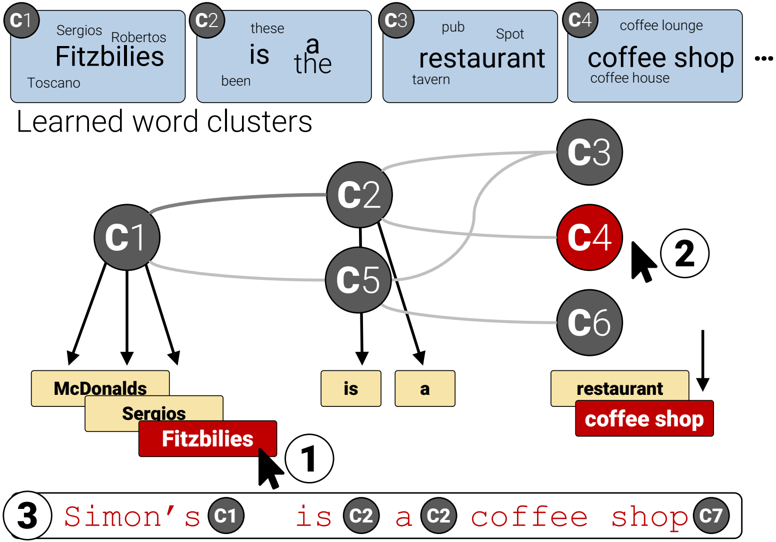
Latent Variable Models for Generation

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



Meaning Representation	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

|| <u>Fitzbillies</u> || 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 is an is an expensive ...	___	providing serving offering ...	___	food cuisine foods ...	in the	high moderate less than average ...	price price range ...

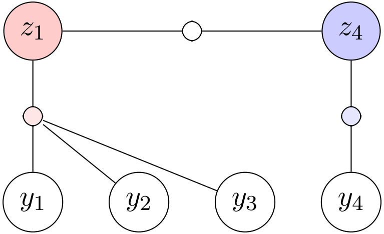
.	It is	located in the located near near ...	___	.	Its customer rating is Their customer rating is Customers have rated it ...	___ out of ___	.

Step 3: Fill-in Each Segment

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

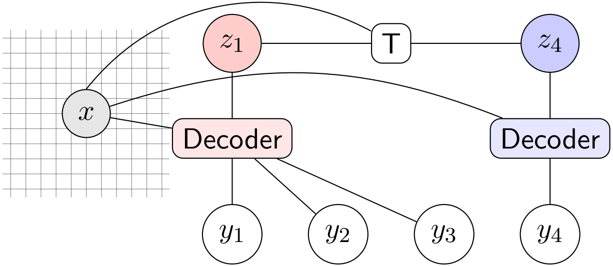
Technical Methodology: Hidden Semi-Markov Model

- 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 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 ___	.

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

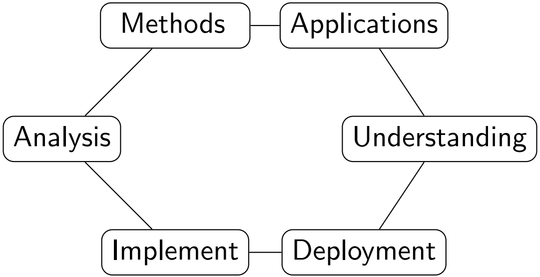
1. kenny warren deutscher (april 1, 1946) is an american ventriloquist.
2. kenny warren deutscher (april 1, 1946 , brooklyn,) is an american ventriloquist.
3. kenny warren deutscher (april 1, 1946) is an american ventriloquist, best known for his the revival of ventriloquism.
4. "kenny" warren is an 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



Long-Form Generation with Explicit Reasoning

(2)

TEAM	WIN	LOSS	PTS	FG_PCT	RB	AS ...
Hawks	11	12	103	49	47	27
Heat	7	15	95	43	34	20

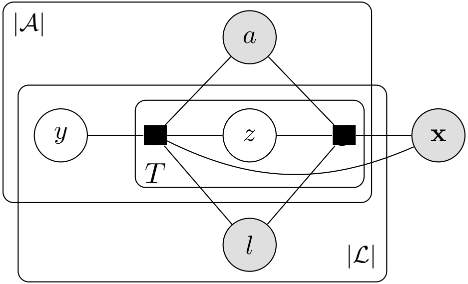
(3)

PLAYER	AS	RB	PT	FG	FGA	CITY ...
Tyler Johnson	5	2	27	8	16	Miami
Dwight Howard	11	17	23	9	11	Atlanta
Paul Millsap	2	9	21	8	12	Atlanta
Goran Dragic	4	2	21	8	17	Miami
Wayne Ellington	2	3	19	7	15	Miami
Dennis Schroder	7	4	17	8	15	Atlanta
Rodney McGruder	5	5	11	3	8	Miami
Hasan Whiteside	2	12	8	4	12	Miami
...						

(1)

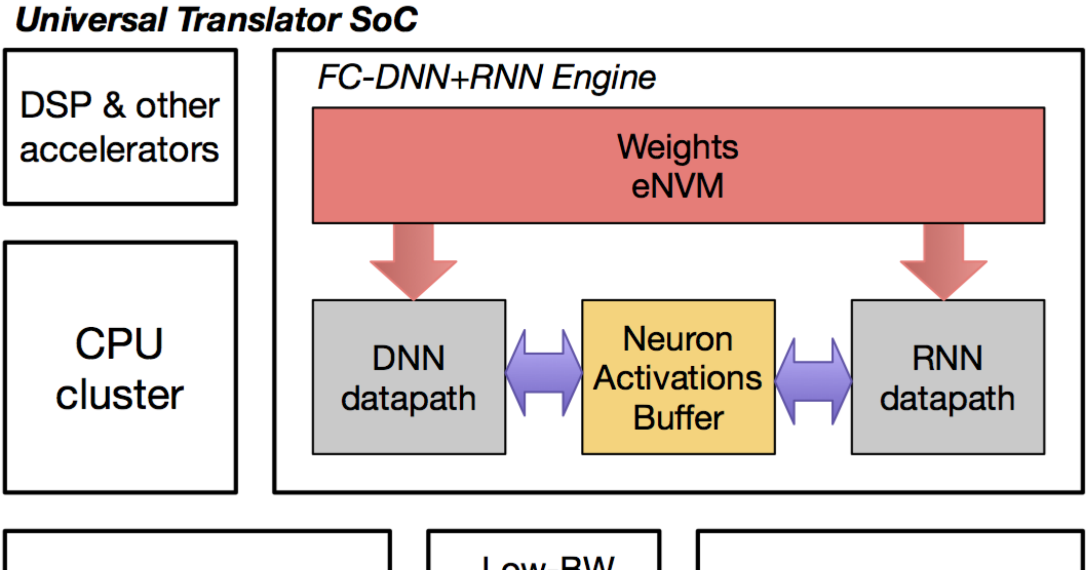
[The Atlanta Hawks defeated the Miami Heat, 103 - 95, at Philips Arena on Wednesday.] [Defense was key for the Hawks, as they held the Heat to 42 percent shooting and forced them to commit 16 turnovers. Atlanta also dominated in the paint, winning the rebounding battle, 47 - 34, and outscoring them in the paint 58 - 26. The Hawks shot 49 percent from the field and assisted on 27 of their 43 made baskets.] [Miami (7 - 15) are as beat-up as anyone right now. Hassan Whiteside really struggled in this game, as he amassed eight points, 12 rebounds and one blocks on 4 - of - 12 shooting ...

Probabilistic Programming (Preprint)



Learning Neural Reasoning-Based Models

Temporary


Universal Translator SoC
DSP & other accelerators
CPU cluster
FC-DNN+RNN Engine
Weights eNVM
DNN datapath
Neuron Activations Buffer
RNN datapath

An open-source neural machine translation system.

English Français 简体中文 한국어 日本語 Русский العربية

Home

Quickstart [Lua]

Quickstart [Python]

Advanced guide

Models and Recipes

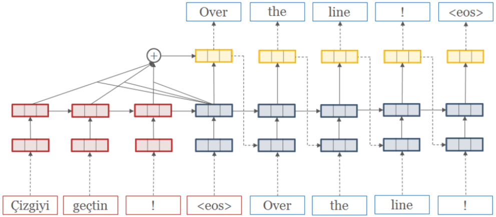
FAQ

About

Documentation

Home

OpenNMT is a industrial-strength, open-source (MIT) neural machine translation system utilizing the Torch/PyTorch mathematical toolkit.



OpenNMT is used as provided in production by major translation providers. The system is designed to be simple to use and easy to extend, while maintaining efficiency and state-of-the-art translation accuracy.

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 and Alexander M. Rush. 2016. Sequence-to-Sequence Learning as Beam-Search Optimization. In *EMNLP*.

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*.