Learning How to Say It: Language Generation post Deep Learning

Alexander M Rush

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

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

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

Talk about Text (Summary) (? w/ Facebook)

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.

f

Talk about Text (Summary) (? w/ Facebook)

f

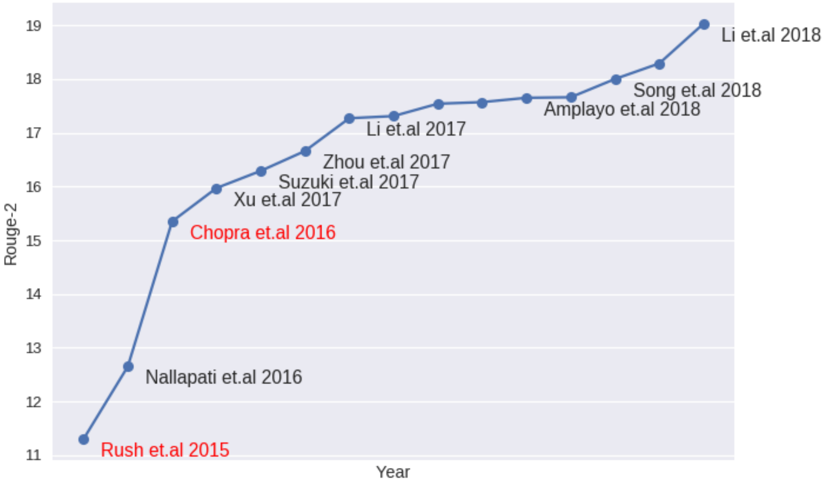
x

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

$y_{1:T}$

Cambodian government rejects opposition's call for talks abroad

Sentence Summarization



Talk about Text (Summary)

London, England (reuters) – Harry Potter star Daniel Radcliffe gains access to a reported $20 million fortune as he turns 18 on monday, but he insists the money won't cast a spell on him. Daniel Radcliffe as harry potter in "Harry Potter and the Order of the Phoenix" to the disappointment of gossip columnists around the world , the young actor says he has no plans to fritter his cash away on fast cars , drink and celebrity parties . " i do n't plan to be one of those people who , as soon as they turn 18 , suddenly buy themselves a massive sports car collection or something similar , " he told an australian interviewer earlier this month . " i do n't think i 'll be particularly extravagant " . " the things i like buying are things that cost about 10 pounds – books and cds and dvds . " at 18 , radcliffe will be able to gamble in a casino , buy a drink in a pub or see the horror film " hostel : part ii , " currently six places below his number one movie on the uk box office chart . details of how he 'll mark his landmark birthday are under wraps . his agent and publicist had no comment on his plans . " i 'll definitely have some sort of party , " he said in an interview . . .

Talk about Text (Summary)

London, England (reuters) – Harry Potter star Daniel Radcliffe gains access to a reported $20 million fortune as he turns 18 on monday, but he insists the money won't cast a spell on him. Daniel Radcliffe as harry potter in "Harry Potter and the Order of the Phoenix" to the disappointment of gossip columnists around the world , the young actor says he has no plans to fritter his cash away on fast cars , drink and celebrity parties . " i do n't plan to be one of those people who , as soon as they turn 18 , suddenly buy themselves a massive sports car collection or something similar , " he told an australian interviewer earlier this month . " i do n't think i 'll be particularly extravagant " . " the things i like buying are things that cost about 10 pounds – books and cds and dvds . " at 18 , radcliffe will be able to gamble in a casino , buy a drink in a pub or see the horror film " hostel : part ii , " currently six places below his number one movie on the uk box office chart . details of how he 'll mark his landmark birthday are under wraps . his agent and publicist had no comment on his plans . " i 'll definitely have some sort of party , " he said in an interview . . .

Harry Potter star Daniel Radcliffe gets $20m fortune as he turns 18 monday. Young actor says he has no plans to fritter his cash away. Radcliffe 's earnings from first five potter films have been held in trust fund.

Document Summarization

Talk about Data (?)

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
...						

Talk about Data (?)

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 ...

E2E Challenge 2018

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.

Submitter	Affiliation	System name	P?	BLEU	NIST	METEOR	ROUGE_L▲	CIDEr
HarvardNLP & Henry Elder	Harvard SEAS & Adapt	main_1_support_3		0.6737	8.6061	0.4523	0.7084	2.3056
Biao Zhang	Xiamen University	bzhang_submit	✓	0.6545	8.1840	0.4392	0.7083	2.1012
HarvardNLP & Henry Elder	Harvard SEAS & Adapt	main_1_support_2		0.6618	8.6025	0.4571	0.7038	2.3371
Shubham Agarwal	NLE	submission_third		0.6676	8.5416	0.4485	0.6991	2.2276
Shubham Agarwal	NLE	submission_second		0.6669	8.5388	0.4484	0.6991	2.2239
Thomson Reuters NLG	Thomson Reuters	NonPrimary_4_test_output_beam_5_model_13_post		0.6742	8.6590	0.4499	0.6983	2.3018
Thomson Reuters NLG	Thomson Reuters	NonPrimary_3_test_output_beam_5_model_11_post		0.6805	8.7777	0.4462	0.6928	2.3195
Chen Shuang	Harbin Institute of Technology	Abstract-greedy		0.6635	8.3977	0.4312	0.6909	2.0788

Talk about the Diagrams (? w/ Bloomberg)

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

Talk about the Diagrams (? w/ Bloomberg)

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

```
{ \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
```

Im2Latex

(? w/ Bloomberg)

$$A_0^3(\alpha'\rightarrow 0)=2g_d\,\varepsilon^{(1)}_\lambda\varepsilon^{(2)}_\mu\varepsilon^{(3)}_\nu\left\{\eta^{\lambda\mu}\left(p_1^\nu-p_2^\nu\right)+\eta^{\lambda\nu}\left(p_3^\mu-p_1^\mu\right)+\eta^{\mu\nu}\left(p_2^\lambda-p_3^\lambda\right)\right\}.$$

```
(A_{0}^{3}(\alpha^{\prime }\rightarrow 0)=2g_{d}\,\,\varepsilon^{(1)}_{\lambda}\varepsilon^{(2)}
_{\mu }\varepsilon^{(3)}_{\nu }\left\{ \eta ^{\lambda \mu}\left( p_{1}^{\nu }-p_{2}^{\nu }\right) +
\eta ^{\lambda \nu }\left(p_{3}^{\mu }-p_{1}^{\mu }\right)+\eta ^{\mu \nu }\left( p_{2}^{\lambda}
-p_{3}^{\lambda }\right) \right\} . \label{17}
```

$$\left\{\begin{array}{rcl}\delta_\epsilon B & \sim & \epsilon F\,, \\ \delta_\epsilon F & \sim & \partial\epsilon+\epsilon B\,,\end{array}\right.$$

```
\left\{\begin{array}{rcl}\delta_{\epsilon} B & \sim & \epsilon F
\, , \\\delta_{\epsilon} F & \sim & \partial\epsilon + \epsilon B
\, , \\\end{array}\right.
```

$$\int\limits_{\mathcal{L}^d_{d-1}} f(H)d\nu_{d-1}(H)=c_3\int\limits_{\mathcal{L}^A_2}\int\limits_{\mathcal{L}^L_{d-1}} f(H)[H,A]^2 d\nu^L_{d-1}(H)d\nu^A_2(L).$$

```
\int \limits_{{\cal L}^{d}_{d-1}}f(H)d\nu_{d-1}(H) =  c_{3} \int \limits_{{\cal L}^{A}_{2}}
\int \limits_{{\cal L}^{L}_{d-1}}f(H)[H,A]^{2}d\nu_{d-1}^{L}(H)d\nu_{2}^{A}(L).
```

$$J=\left(\begin{array}{cc}\alpha^t & \tilde{f}_2 \\ f_1 & \tilde{A}\end{array}\right)\left(\begin{array}{ll}0 & 0 \\ 0 & L\end{array}\right)\left(\begin{array}{cc}\alpha & \tilde{f}_1 \\ f_2 & A\end{array}\right)=\left(\begin{array}{ll}\tilde{f}_2 L f_2 & \tilde{f}_2 L A \\ \tilde{A} L f_2 & \tilde{A} L A\end{array}\right)$$

```
J=\left( \begin{array}{cc}\alpha ^{t} & \tilde{f}_{2} \\ f_{1} & \tilde{A}
\end{array}\right) \left( \begin{array}{ll}0 & 0 \\ 0 & L\end{array}\right)
\left( \begin{array}{cc}\alpha & \tilde{f}_{1} \\ f_{2} & A\end{array}\right) =
\left( \begin{array}{ll}\tilde{f}_{2}Lf_{2} & \tilde{f}_{2}LA \\ \tilde{A}Lf_{2} &
\tilde{A}LA\end{array}\right)
```

$$\lambda^{(2)}_{n,1}=\frac{\partial\overline{H}_0}{\partial q_{n,0}}\ ,lambda^{(2)}_{n,j_n}=\frac{\partial\overline{H}_0}{\partial q_{n,j_n-1}}-\mu_{n,j_n-1}\ ,\quad j_n=2,3,\cdots,m_n-1\ .$$

```
\lambda_{n,1}^{(2)}=\frac{\partial\overline{H}_0}{\partial q_{n,0}}\ ,\ \\lambda_{n,j_n}^{(2)}=\frac{
\partial\overline{H}_0}{\partial q_{n,j_n-1}}-\mu_{n,j_n-1}\ ,\       \ j_n=2,3,\cdots,m_n-1\ .
```

$$(P_{ll'}-K_{ll'})\phi'(z_q)|\chi>=0$$

```
(P_{ll'} - K_{ll'}) \phi '(z_{q})|\chi > = 0
```

Im2Latex

(? w/ Bloomberg)

Temporary

NLP

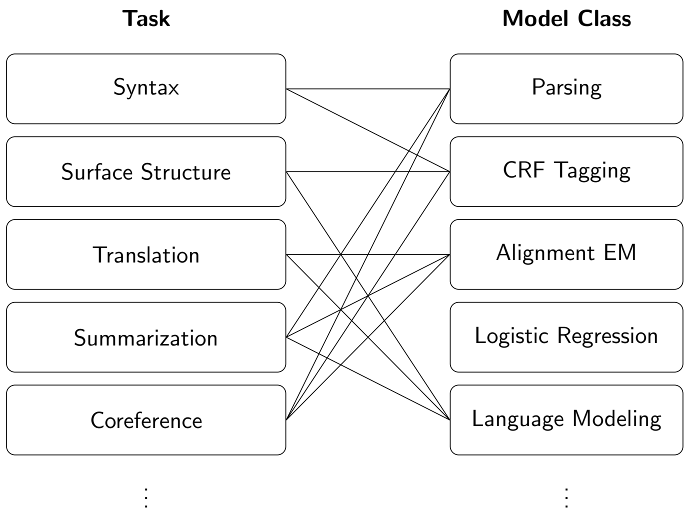
State-of-the-Art Natural Language Processing, circa 2009

Task

- Syntax
- Surface Structure
- Translation
- Summarization
- Coreference

⋮

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



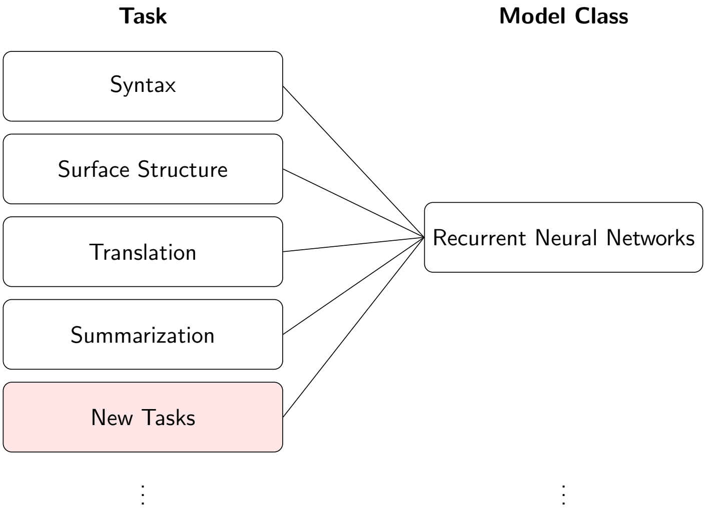
State-of-the-Art Natural Language Processing, circa 2019

Task

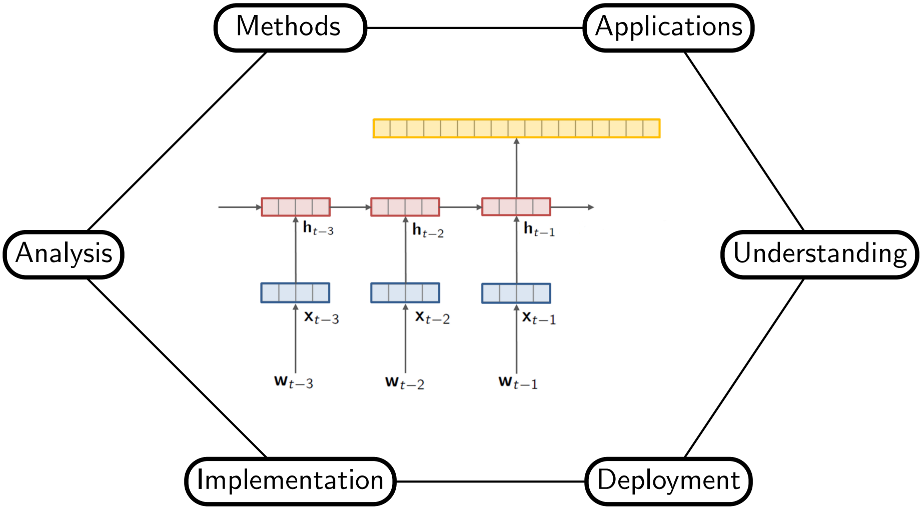
Syntax
Surface Structure
Translation
Summarization
New Tasks

⋮

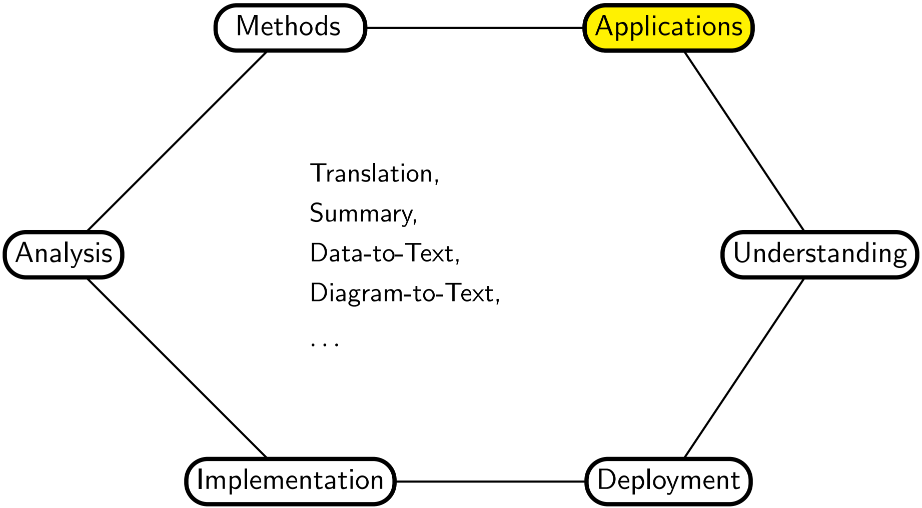
State-of-the-Art Natural Language Processing, circa 2019



Harvard NLP Deep Learning Research



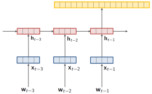
Harvard NLP Deep Learning Research



Selected Harvard NLP Deep Learning Research

Methods — Applications

Analysis



Understanding

Implement — Deployment

Natural Lang. Processing

Machine Learning

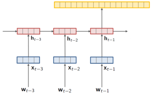
Visualization

Selected Harvard NLP Deep Learning Research

? ? ?

Methods — Applications

Analysis



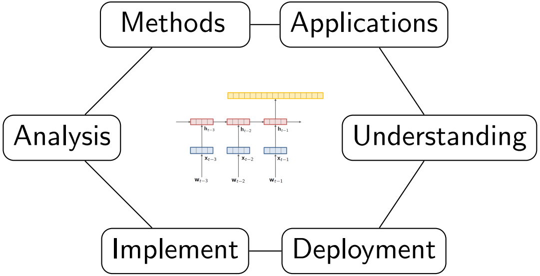
Understanding

Implement — Deployment

Natural Lang. Processing
Machine Learning
Visualization

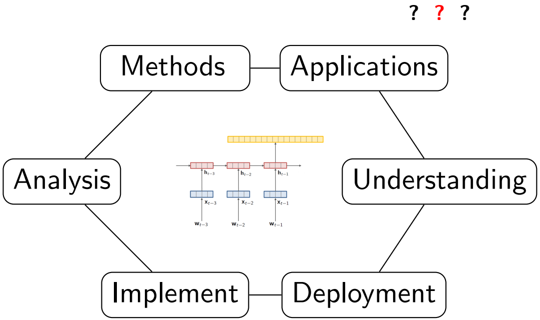
Selected Harvard NLP Deep Learning Research

? ? ?



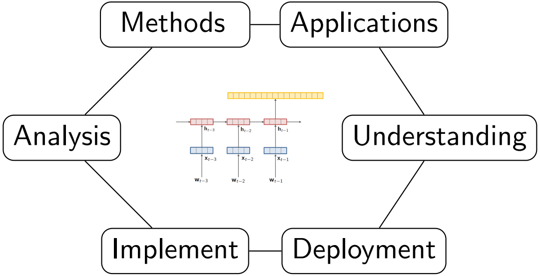
Natural Lang. Processing
Machine Learning
Visualization

Selected Harvard NLP Deep Learning Research



Natural Lang. Processing
Machine Learning
Visualization

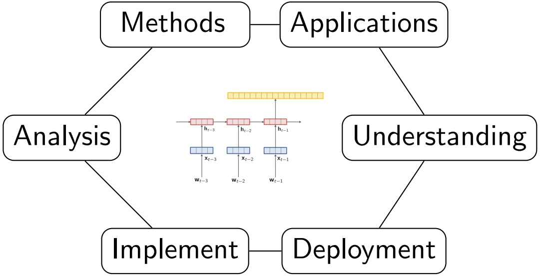
Selected Harvard NLP Deep Learning Research



? ? ?

Natural Lang. Processing
Machine Learning
Visualization

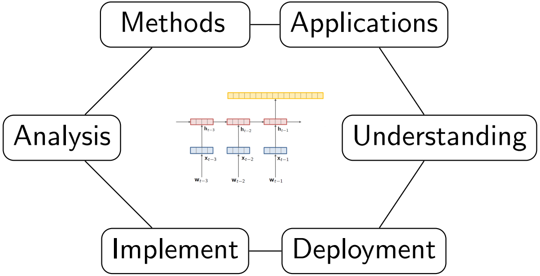
Selected Harvard NLP Deep Learning Research



? ? ?

Natural Lang. Processing
Machine Learning
Visualization

Selected Harvard NLP Deep Learning Research



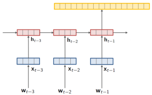
? ? ?

Natural Lang. Processing
Machine Learning
Visualization

Selected Harvard NLP Deep Learning Research

? ? ? ? ? ?

Methods — Applications

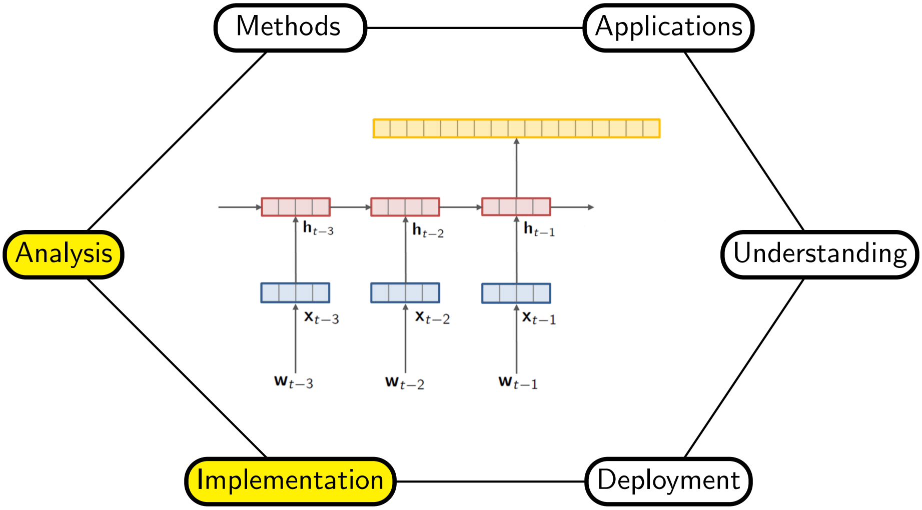
? ? ? Analysis  Understanding ? ? ?

Implement — Deployment

? ? ? ? ? ?

Natural Lang. Processing
Machine Learning
Visualization

Research Direction



Generation Setup (Reminder)

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

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

Generation Setup (Reminder)

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

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

Generation Setup (Reminder)

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

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

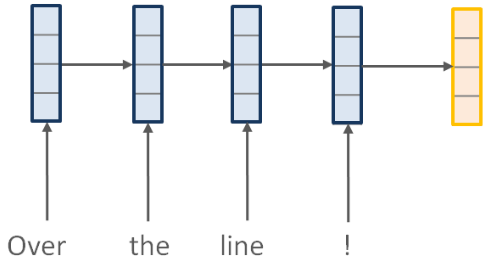
Generation Setup (Reminder)

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

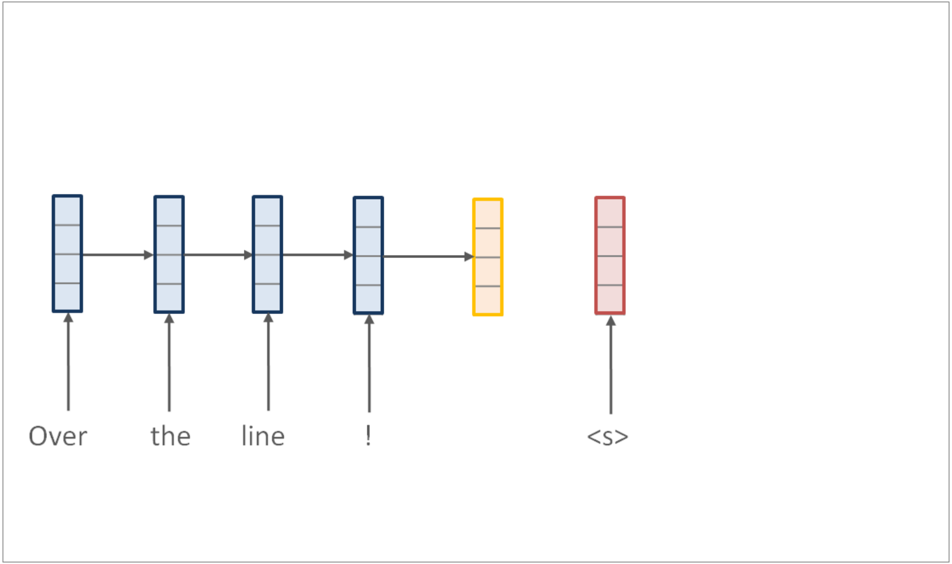
- Input $x_{1:S}$, *what to talk about*
- Output text $y^*_{1:T}$, *how to say it*
- Model $f(.;\theta)$, learned from data

Over the line !

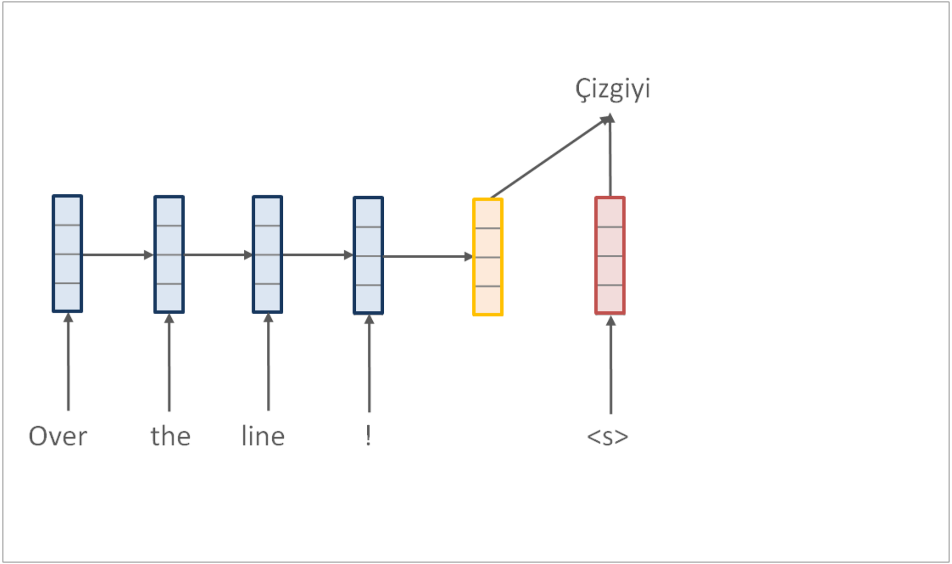
Recurrent Neural Networks



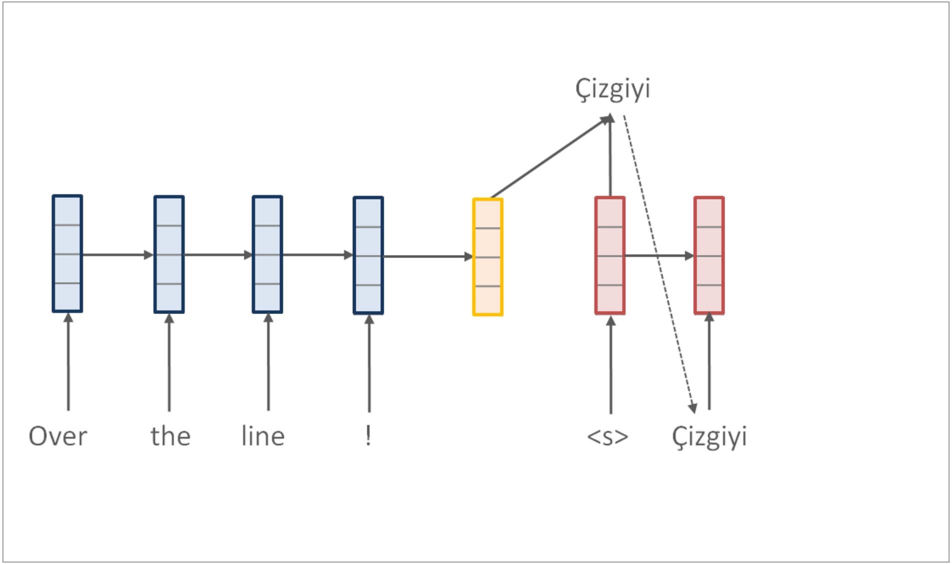
Recurrent Neural Networks



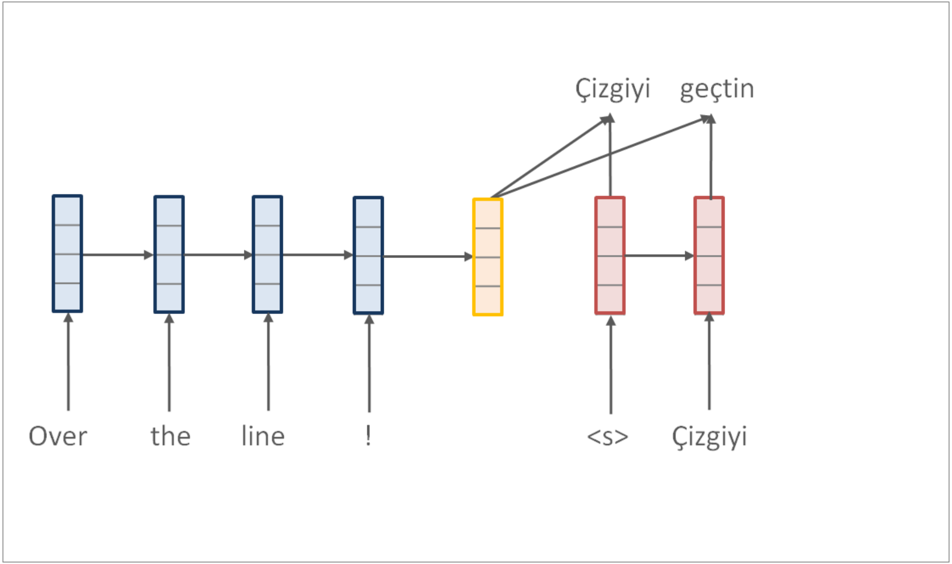
Recurrent Neural Networks



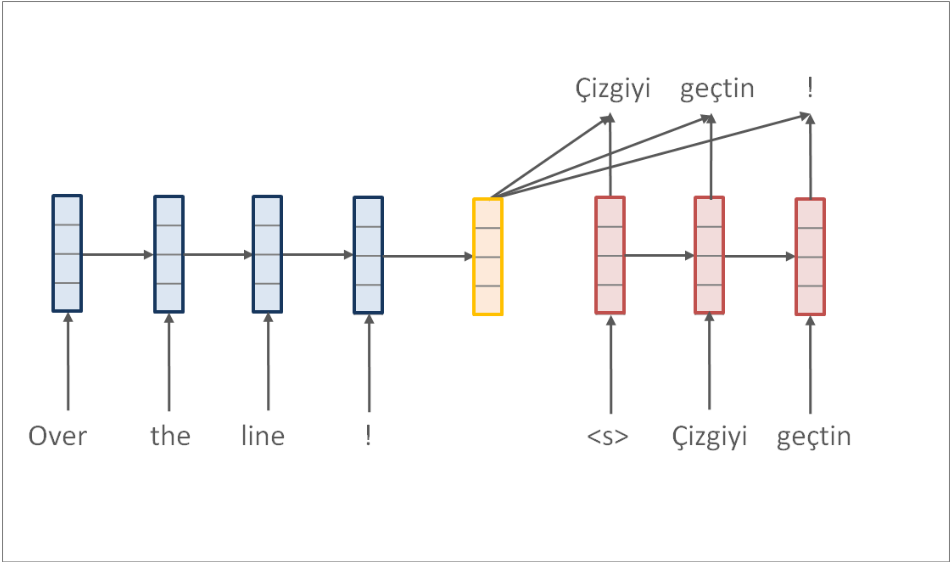
Recurrent Neural Networks



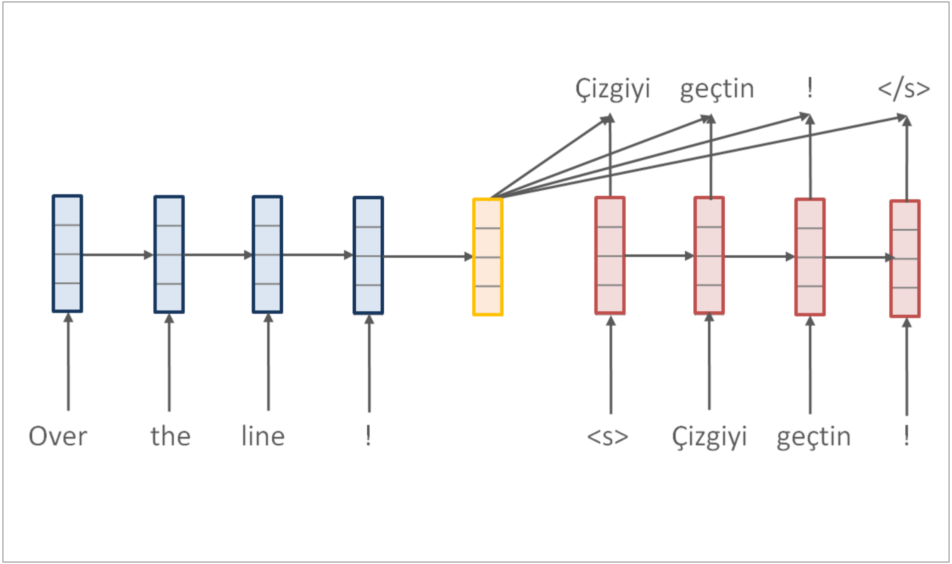
Recurrent Neural Networks



Recurrent Neural Networks



Recurrent Neural Networks



RNN 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:

$$p(y_{t+1} \mid y_{1:t}, x) = \mathrm{softmax}(\mathbf{W}[\mathbf{h}_t, \mathbf{c}])$$

RNN 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:

$$p(y_{t+1} \mid y_{1:t}, x) = \mathrm{softmax}(\mathbf{W}[\mathbf{h}_t, \mathbf{c}])$$

Generation:

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

Toy Example: Parenthesis Language

alphabet: `( ) 0 1 2 3 4`

corpus: `( 1 ( 2 ) () ) 0 ( ( ( 3 ) ) 1 )`

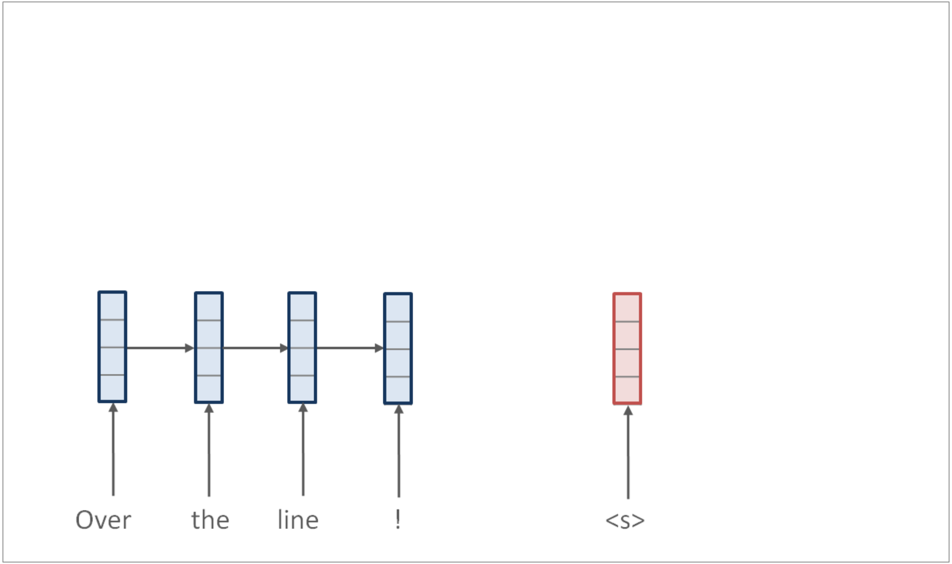
LSTMVis - Parenthesis Language (? w/ IBM)

Temporary

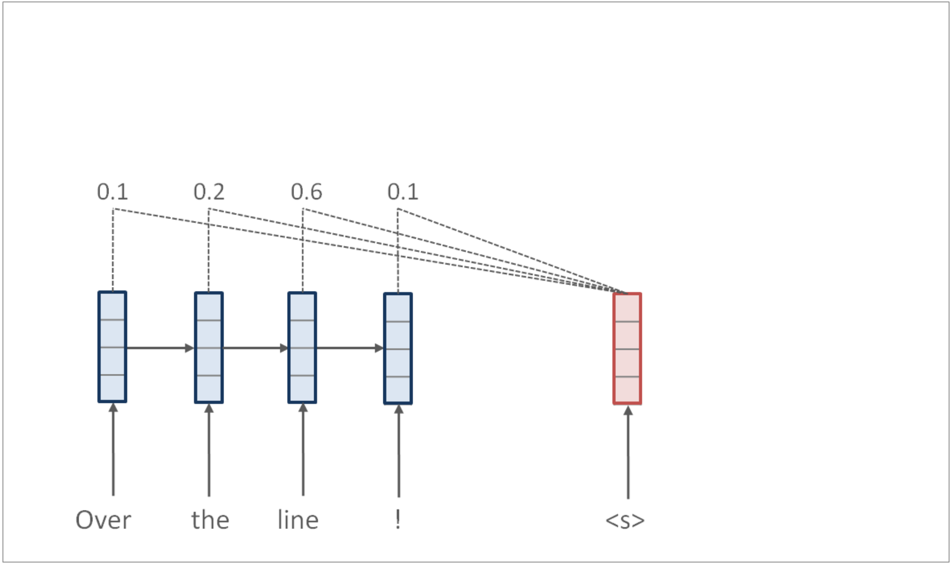
LSTMVis - Natural Language (? w/ IBM)

Temporary

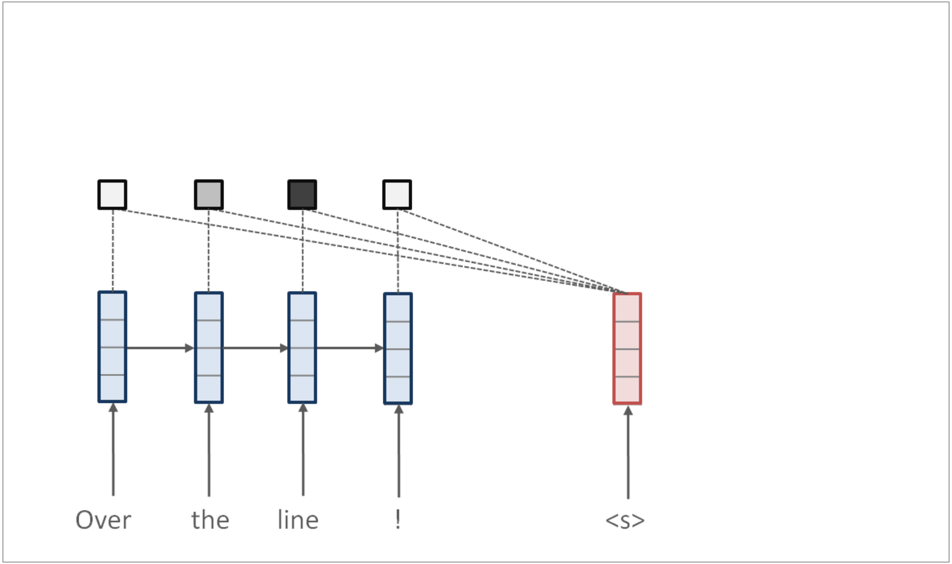
Seq2Seq + Attention Model



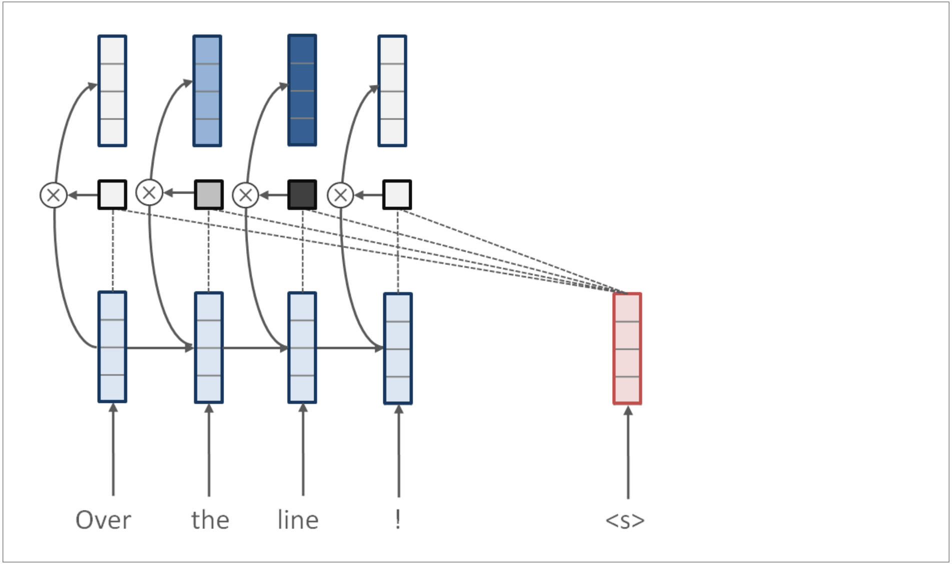
Seq2Seq + Attention Model



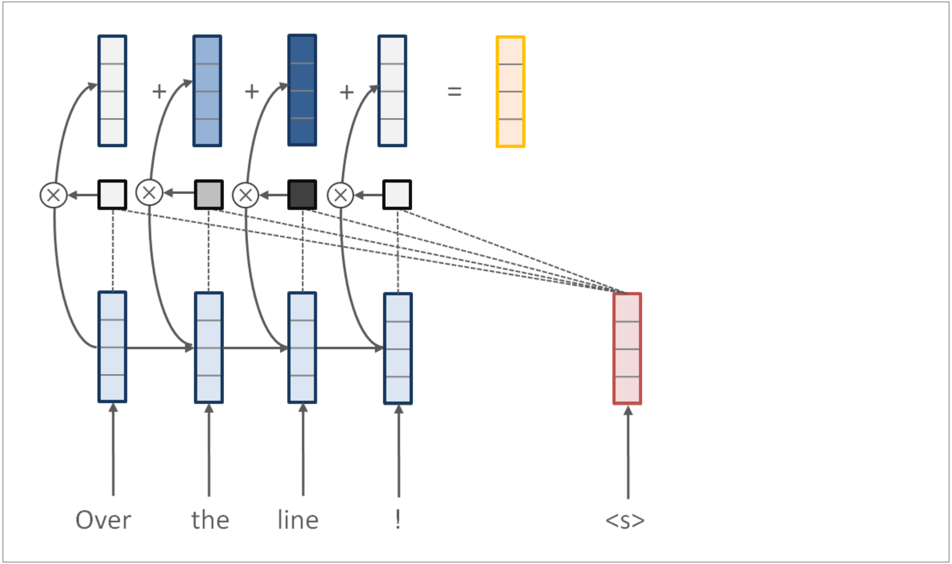
Seq2Seq + Attention Model



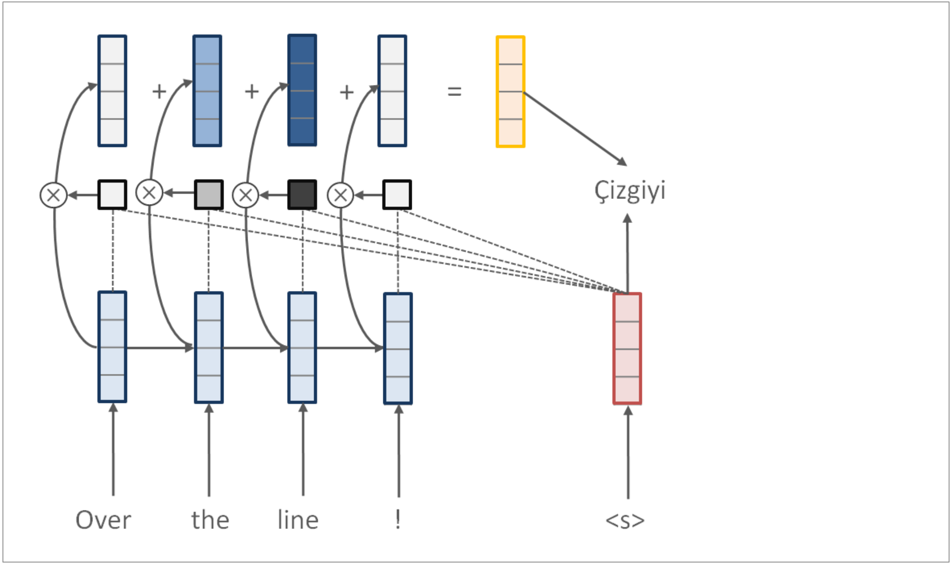
Seq2Seq + Attention Model



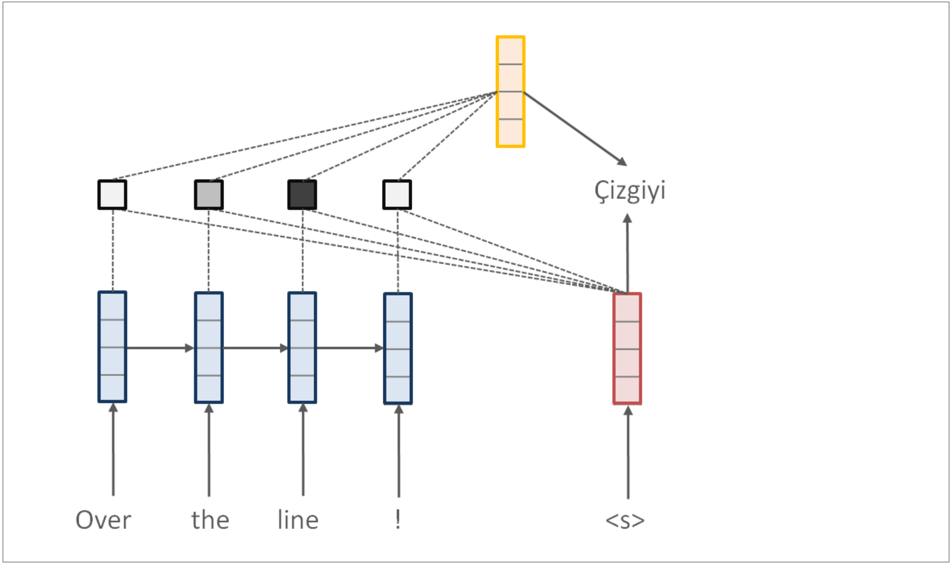
Seq2Seq + Attention Model



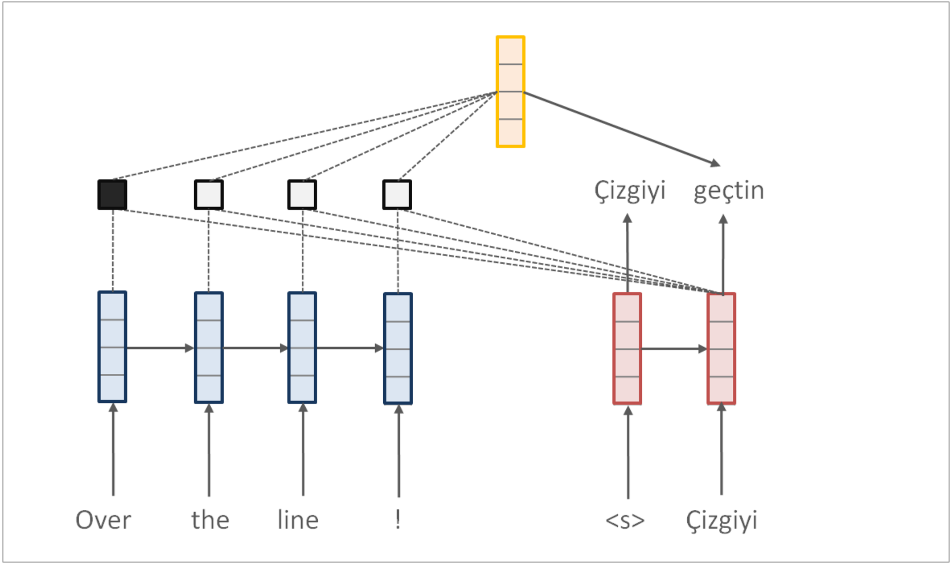
Seq2Seq + Attention Model



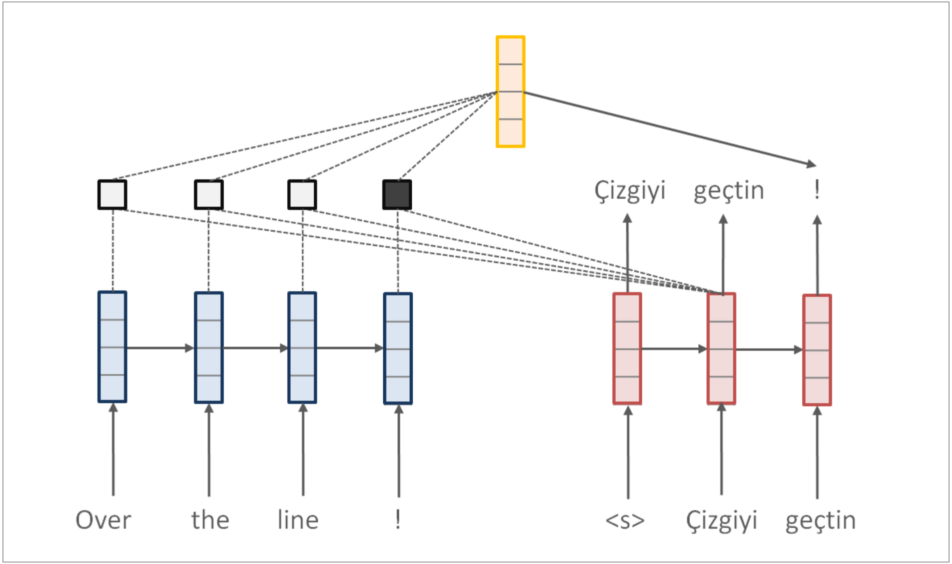
Seq2Seq + Attention Model



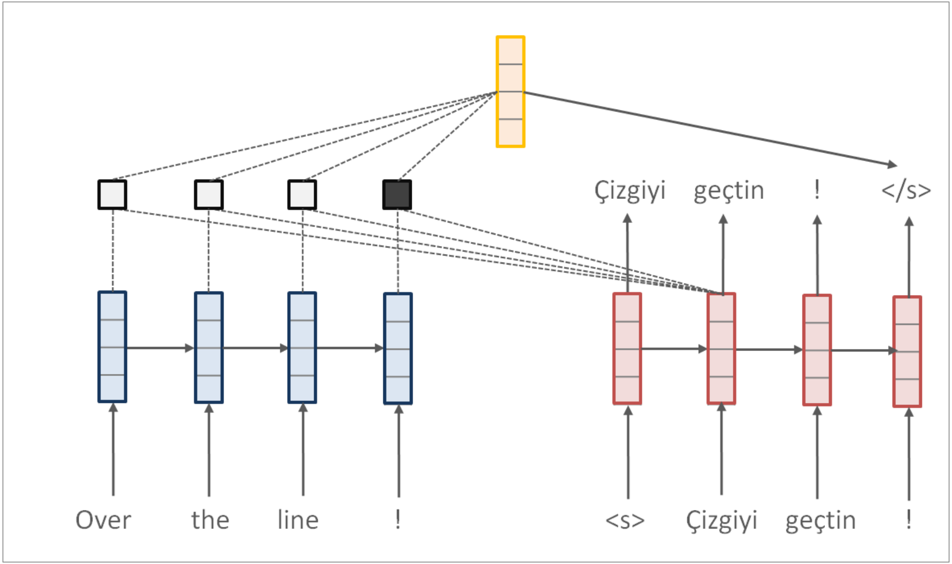
Seq2Seq + Attention Model



Seq2Seq + Attention Model



Seq2Seq + Attention Model



Attention Math

Encoder:

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

Attention

$$\alpha \leftarrow \mathrm{softmax}([\mathbf{h}_1^x; \dots; \mathbf{h}_S^x]^\top \mathbf{h}_t) \qquad \mathbf{c} \leftarrow \sum_{s=1}^{S} \alpha_s \mathbf{h}_s^x$$

Decoder:

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

Prediction:

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

Seq2SeqVis (? w/ IBM)

Temporary

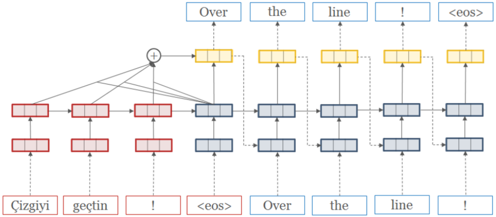
An open-source neural machine translation system.

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

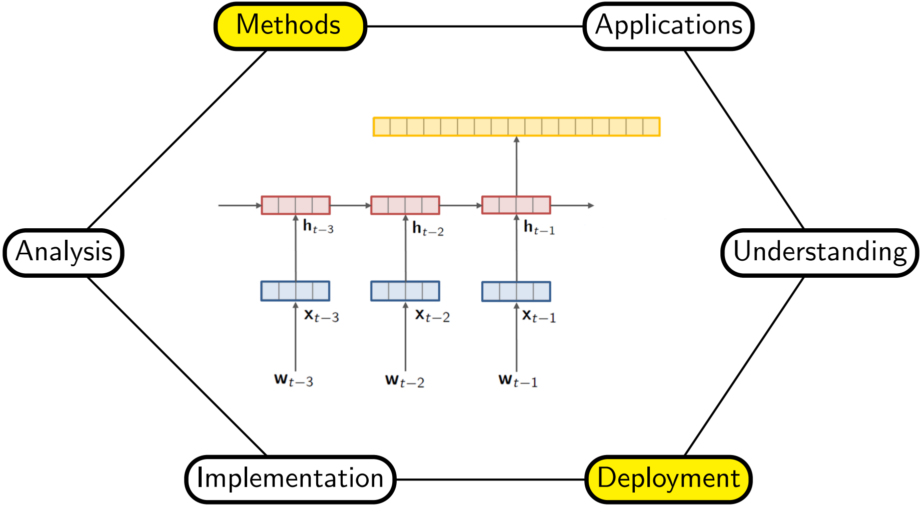
OpenNMT is a industrial-strength, open-source (MIT) neural machine translation system utilizing the Torch/PyTorch mathematical toolkit.



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.

- Collaborative open-source project started at Harvard, now self-sustaining.
- Used in production by Systran, Ubiqus, Booking.com, and others.
- Over 100 developers in France, China, Japan, Portugal, and the US.
- Designed to be research extensible to latest machine translation techniques.
- Pretrained models for translation as well as everything in this talk.

Research Direction



Structured Modeling

Generation Setup (Reminder)

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

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

Generation Setup (Reminder)

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

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

Generation Setup (Reminder)

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

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

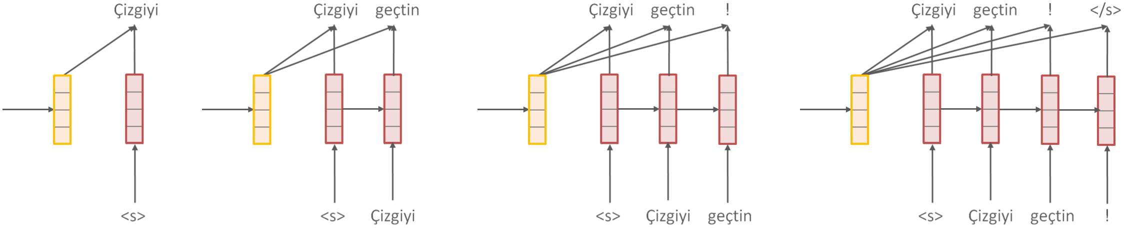
Generation Setup (Reminder)

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

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

Training Seq2Seq

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

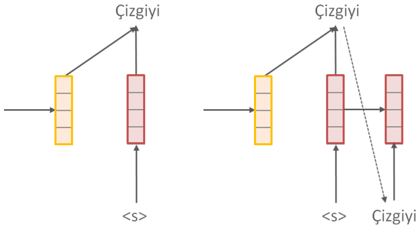


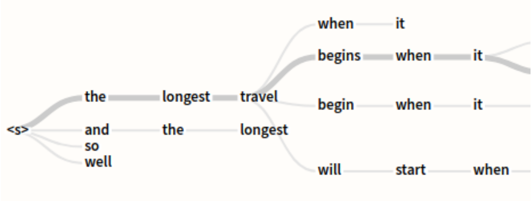
Pretend it is Multiclass classification:

$$\mathrm{NLL}(\theta) = -\sum_t \log p(y_t | y_{1:t-1}, \mathbf{c}; \theta)$$

Deploying Seq2Seq

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





Deploy as a structured model:

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

Best Sequence Heuristic: Beam Search

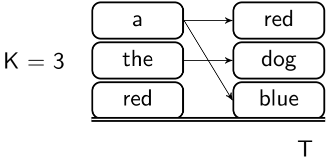
a

$K = 3$ the

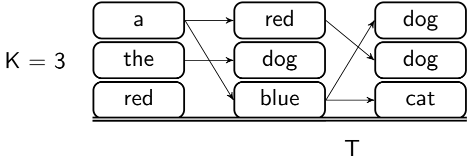
red

T

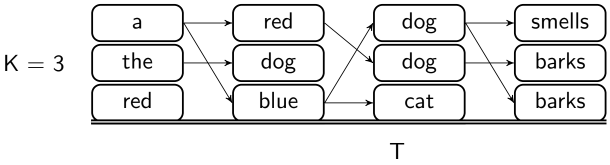
Best Sequence Heuristic: Beam Search



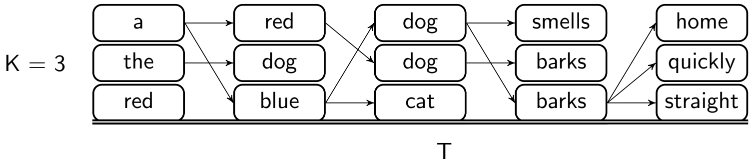
Best Sequence Heuristic: Beam Search



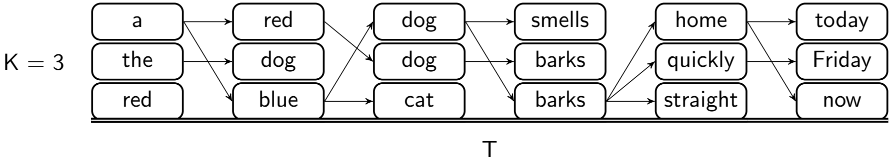
Best Sequence Heuristic: Beam Search



Best Sequence Heuristic: Beam Search



Best Sequence Heuristic: Beam Search



1. Compute the score of every possible next word.

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

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

1. Exposure Bias
   - Training by conditioning on true $y_{1:t-1}$.
2. Metric Bias
   - Training with local NLL, evaluate with hamming-style losses.
3. Label Bias
   - Locally normalized models have known pathological issues.

Work

Can we exploit discrete sequences to improve models for text generation?

Applications:

- (1) Sequence-to-Sequence as Beam Search Optimization for training.
- (2) Sequence Knowledge Distillation for deployment.

Beam Search Optimization (?)

Motivation: Can we fix target theoretical issues by unifying training and test objective?

Change 1: Modify Scoring Function

Same model, but replace $\log p(y_t | y^{(k)}_{1:t-1}, \mathbf{c}; \theta)$ with globally normalized $f(y_t, y^{(k)}_{1:t-1}, \mathbf{c}; \theta)$

Change 2: Run Beam Search During Training

1. Compute the score of every possible next word.

$$f(y_t, y^{(k)}_{1:t-1}) \leftarrow \log p(y_t \mid y^{(k)}_{1:t-1}, \mathbf{c}) + \log p(y^{(k)}_{1:t-1} \mid \mathbf{c})$$

2. Prune to only the K highest-scoring,

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

Change 2: Run Beam Search During Training

1. Compute the score of every possible next word.

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

2. Prune to only the K highest-scoring,

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

Change 3: Replace train to enforce beam-search margin

New Global Objective:

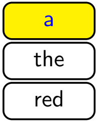
- Margin between gold seq $y^{(g)}$ and last seq on beam $y^{(K)}$

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

- Slack-rescaled, margin-based sequence criterion, at each time step.

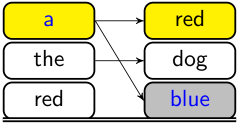
- When violation occurs, target replaces current beam (learning as search optimization **?**)

Beam Search Optimization Training Example



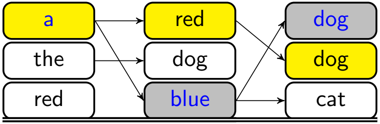
- Color Gold: target sequence $y^{(g)}$
- Color Gray: violating sequence $y^{(K)}$

Beam Search Optimization Training Example



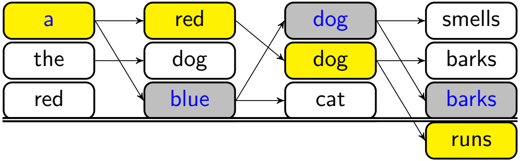
- Color Gold: target sequence $y^{(g)}$
- Color Gray: violating sequence $y^{(K)}$

Beam Search Optimization Training Example



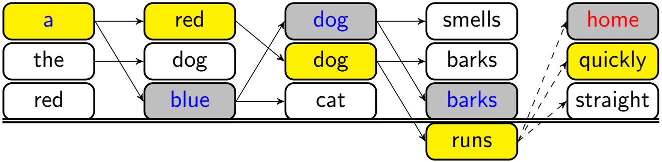
- Color Gold: target sequence $y^{(g)}$
- Color Gray: violating sequence $y^{(K)}$

Beam Search Optimization Training Example



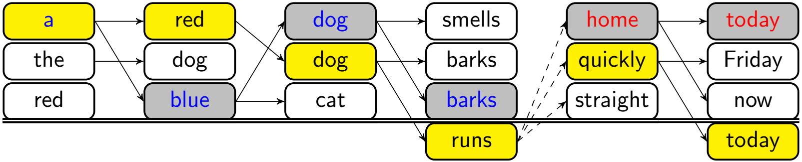
- Color Gold: target sequence $y^{(g)}$
- Color Gray: violating sequence $y^{(K)}$

Beam Search Optimization Training Example



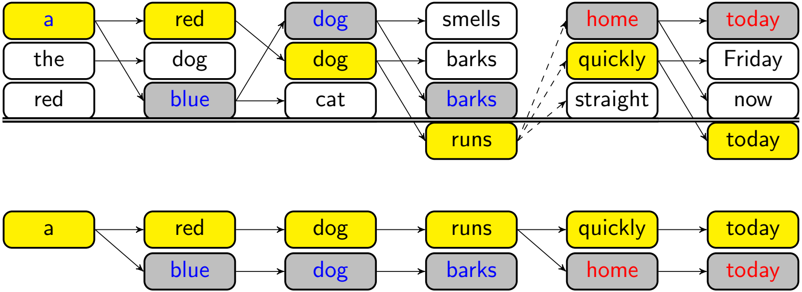
- Color Gold: target sequence $y^{(g)}$
- Color Gray: violating sequence $y^{(K)}$

Beam Search Optimization Training Example



- Color Gold: target sequence $y^{(g)}$
- Color Gray: violating sequence $y^{(K)}$

Structured Backpropagation



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

Theoretical Issues Revisited

- Exposure Bias
  - Beam search at training
- Train/Test Loss Mismatch
  - Slack-rescaled margin can capture correct loss.
- Label Bias **?**
  - Sequence regression is not locally normalized

	$K_e = 1$	$K_e = 5$	$K_e = 10$
	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**

	$K_e = 1$	$K_e = 5$	$K_e = 10$
	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**
	Dependency Parsing (UAS/LAS)		
seq2seq	**87.33/82.26**	88.53/84.16	88.66/84.33
BSO	86.91/82.11	91.00/**87.18**	91.17/**87.41**
BSO-Con	85.11/79.32	**91.25**/86.92	**91.57**/87.26

	$K_e = 1$	$K_e = 5$	$K_e = 10$
	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**
	Dependency Parsing (UAS/LAS)		
seq2seq	**87.33/82.26**	88.53/84.16	88.66/84.33
BSO	86.91/82.11	91.00/**87.18**	91.17/**87.41**
BSO-Con	85.11/79.32	**91.25**/86.92	**91.57**/87.26
	Machine Translation (BLEU)		
seq2seq	22.53	24.03	23.87
BSO, SB-Δ, K_t=6	**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

Sequence Knowledge Distillation (?)

Goal: Compress text generation models.

- **Pruning**: Prune weights based on importance criterion **??**
- **Knowledge Distillation**: Train a *student* model to learn from a *teacher* model **???**.

Other methods:

- low-rank matrix factorization of weight matrices **?**
- weight binarization **?**
- weight sharing **?**

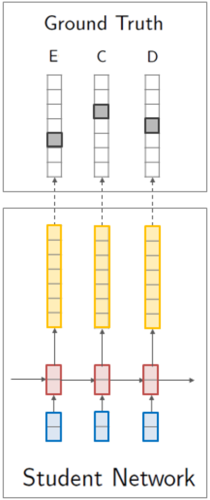
Baseline Model

Standard model minimize NLL(θ):

$$-\sum_t \log p(y_t = y_t^{(g)} \mid y_{1:t-1}^{(g)}, \mathbf{c}; \theta)$$

where $y_t^{(g)}$ is the ground truth word at time t.

Cross-entropy with ground truth.

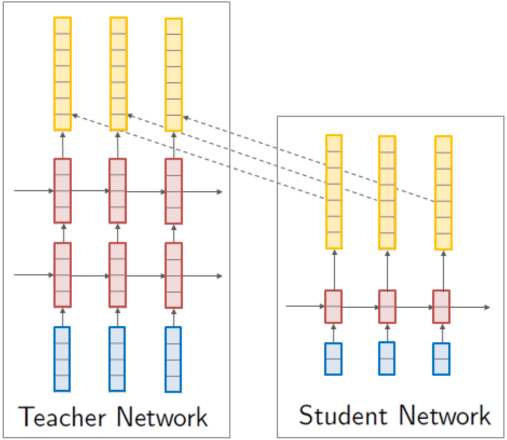


Standard Method: Word-Level Knowledge Distillation

Teacher network: $q(y_t | y_{1:t-1}, \mathbf{c}; \theta_T)$

Minimize cross-entropy between teacher and student distribution $\mathcal{L}_{\text{WORD-KD}}(\theta)$

$$-\sum_{t}\sum_{v} q(y_t = v \mid y^{(g)}_{1:t-1}, \mathbf{c}; \theta_T) \times$$
$$\log p(y_t = v \mid y^{(g)}_{1:t-1}, \mathbf{c}; \theta)$$



Sequence-Level Knowledge Distillation

Proposal: Replace multi-class with sequence distribution. Instead of word NLL,

$$-\sum_{t}\sum_{v} q(y_t = v \mid y^{(g)}_{1:t-1}, \mathbf{c}; \theta_T) \times \log p(y_t = v \mid y^{(g)}_{1:t-1}, \mathbf{c}; \theta)$$

Minimize cross-entropy between q and p implied sequence-distribution

$$-\sum_{v_1} \dots \sum_{v_T} q(y_{1:T} = v_{1:T} | \mathbf{c}; \theta_T) \times \log p(y_{1:T} = v_{1:T} | \mathbf{c}; \theta)$$

However, as before this term is intractable.

A Simple Approximation

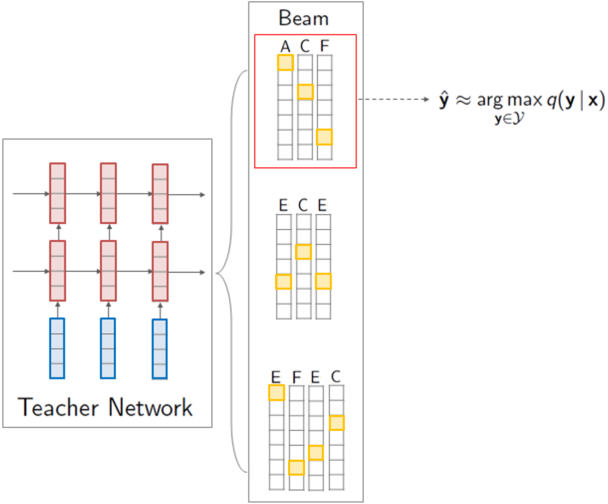
Approximate $q(y_{1:T} \mid \mathbf{c})$ with mode

$$q(y_{1:T} \mid \mathbf{c}) \approx \mathbf{1}\{\arg\max_{y} q(y_{1:T} \mid \mathbf{c})\}$$

Roughly obtained with beam search

$$y^*_{1:T} \approx \arg\max_{y_{1:T}} q(y_{1:T} \mid \mathbf{c})$$

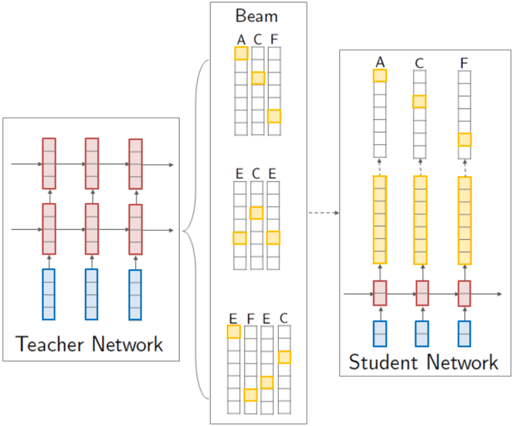
Empirically, point estimate captures significant mass



Sequence-Level Knowledge Distillation

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

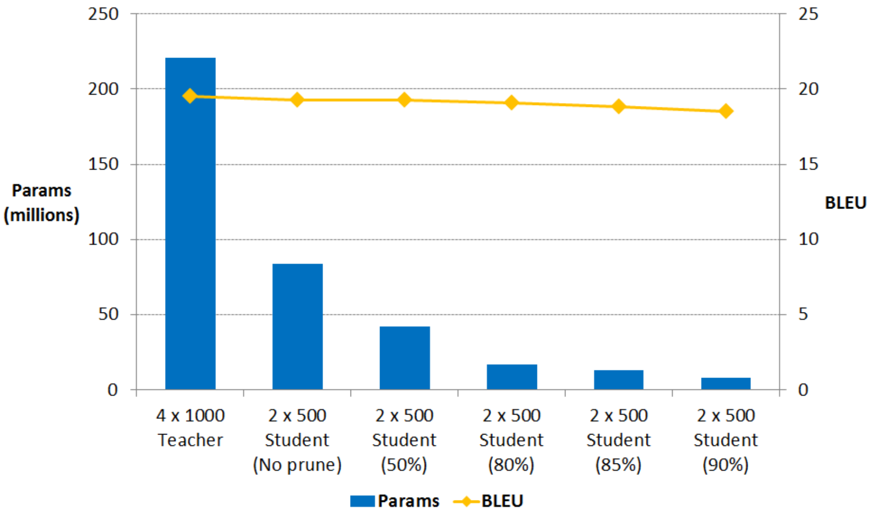
Simplest model: train the student model on y^* with NLL



Results: English $\rightarrow$ German

Model	$\text{BLEU}_{K=1}$	$\Delta_{K=1}$	$\text{BLEU}_{K=5}$	$\Delta_{K=5}$	PPL	$p(y^*)$
4×1000						
Teacher	17.7	–	19.5	–	6.7	1.3%
Seq-Inter	19.6	+1.9	19.8	+0.3	10.4	8.2%
2×500						
Student	14.7	–	17.6	–	8.2	0.9%
Word-KD	15.4	+0.7	17.7	+0.1	8.0	1.0%
Seq-KD	18.9	**+4.2**	19.0	+1.4	22.7	16.9%
Seq-Inter	18.9	**+4.2**	19.3	**+1.7**	15.8	7.6%

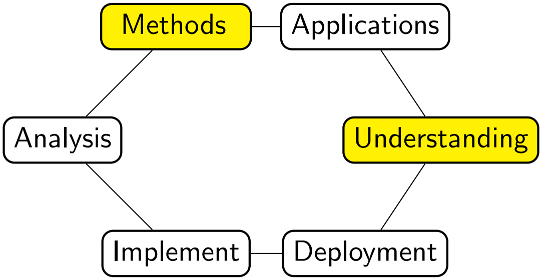
Combining Knowledge Distillation and Pruning



Application

Temporary

Research Direction



Method: Deep Latent-Variable Models

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

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

Method: Deep Latent-Variable Models

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

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

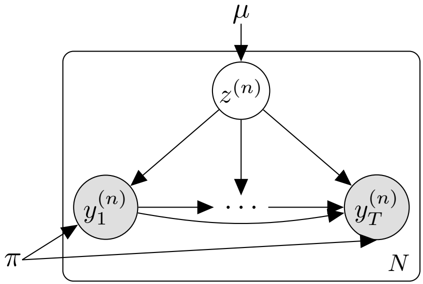
- y is our observed data
- z is a collection of problem-specific latent variables
- θ are the deterministic, neural network parameters.

Example Model: Mixture of RNNs

Generative process:

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

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



Main Requirement: Posterior Inference

For models $p(y, z;\, \theta)$, we'll be interested in the *posterior* over latent variables z:

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

Main Requirement: Posterior Inference

For models $p(y, z;\, \theta)$, we'll be interested in the *posterior* over latent variables z:

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

Why?

- Required for training
- Latent z gives separation of data.

How?

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

In Applications: 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

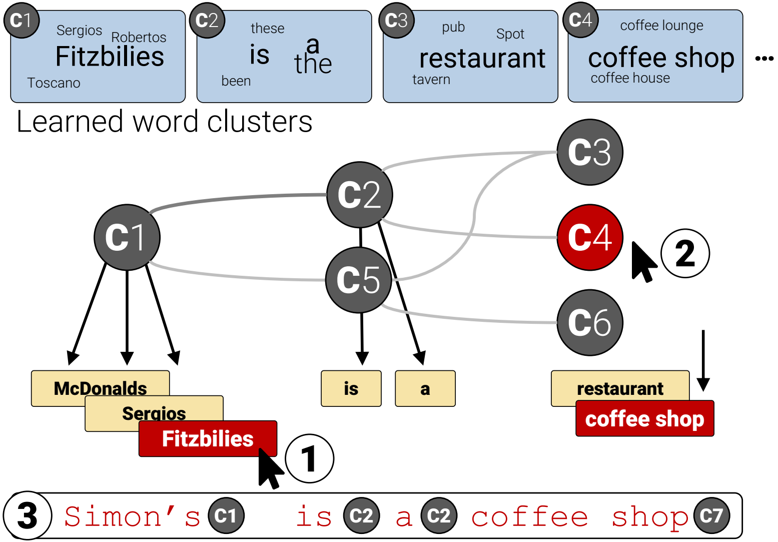
francis *saili* has signed a two-year deal to join munster later this year .
the 24-year-old was part of the new zealand under-20 side that won the junior world championship in italy in 2011 .
saili 's signature is something of a coup for munster and head coach anthony foley .

(See et al, 2017)

Latent Variable Models for Generation

- Can we develop other discrete latent-variable models for generation?
- Perhaps each important aspect of generation can be built-in directly.
- Goals:
  - Model Control
  - Model Debugging
  - Model Uncertainty

Approach 1: Learning Neural Templates



Standard Copy Generation Approach

Step 1: Encode the Source

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

Step 2: Generate with Copy 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.

(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

|| Fitzbillies ||

(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 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 || <u>coffee shop</u> || providing || <u>Chinese</u> || food || in the || moderate || price range || . || It is || located in the || <u>city centre</u> || . ||

Criteria

1. Interpretable in its content selection.

   Decisions are localized to a segment of the template.

2. Easily controllable in terms of style and form.

   Alternative realizations through different templates.

Criteria

1. Interpretable in its content selection.

   Decisions are localized to a segment of the template.

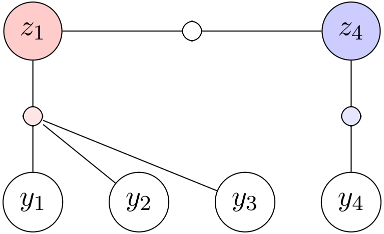
2. Easily controllable in terms of style and form.

   Alternative realizations through different templates.

However: templates feel much less "end-to-end". How can we learn them from data?

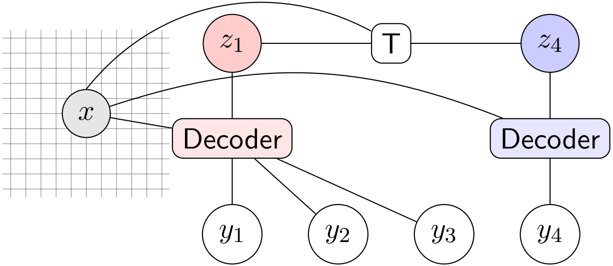
Technical Methodology: Hidden Semi-Markov Model

- HMM: discrete latent states with single emissions (e.g. words).
- HSMM: discrete 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+Copy-Attention, one per state k.



Technical Methodology: Learning Templates

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

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

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.

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

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 ___	.

Experimental Setup

- Two datasets, E2E challenge and WikiBio
- Training with 35 and 65 state models, each 1x300 LSTMs.
- Extract 100 most common templates for each.
- Vocabulary limited to non-copy-able words.
- Generation with beam search with a pre-selected template.

E2E Challenge

	BLEU	NIST
Val		
Substitution	43.71	6.72
Neural Template	66.06	7.93
Full Neural Model	69.25	8.48
Test		
Substitution	43.78	6.88
Neural Template	56.72	7.63
Full Neural Model	65.93	8.59

WikiBio

	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

- Custom KN and NNLM Baselines from LeBret et al (2016)

k

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

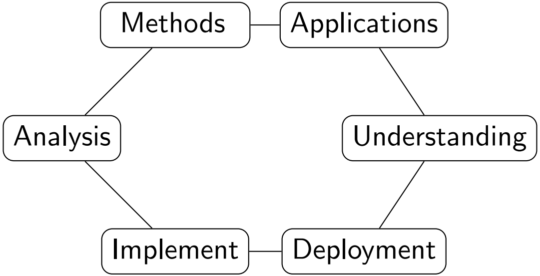
1. kenny warren deutscher (april 1, 1946) is an american ventriloquist.
2. kenny warren deutscher (april 1, 1946 , brooklyn,) is an american ventriloquist.
3. kenny warren deutscher (april 1, 1946) is an american ventriloquist, best known for his the revival of ventriloquism.
4. "kenny" warren is an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

Controllability

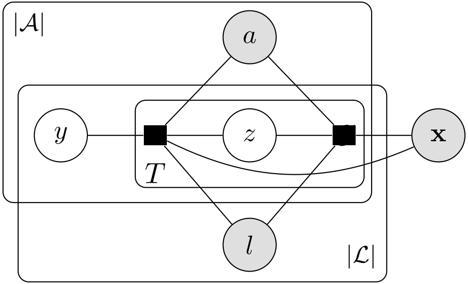
The Golden Palace
name[The Golden Palace], type[coffee shop], food[Chinese], priceRange[cheap] custRating[5 out of 5], area[city centre],
1. The Golden Palace is a coffee shop located in the city centre. 2. In the city centre is a cheap Chinese coffee shop called The Golden Palace. 3. The Golden Palace 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



Probabilistic Programming (Preprint)



Reasoning-Based Models

Temporary

Hardware for NLP (Preprint)

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 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*.