

# Interpreting, Training, and Distilling Seq2Seq Models

Alexander Rush (@harvardnlp)

(with Yoon Kim, Sam Wiseman, Allen Schmaltz, Sebastian Gehrmann, Hendrik Strobelt)



at



What's ML aspects have defined NLP problems?

① Large, discrete input state spaces.

- Vocabulary sizes in 10,000 – 100,000

② Long-term dependencies

- *Sasha is giving a talk today at twitter, . . . , he is excited.*

③ Variable-length output spaces

- e.g. sentences, documents, conversations

*Although current deep learning research tends to claim to encompass NLP, I'm (1) much less convinced about the strength of the results, compared to the results in, say, vision*

...

- Michael Jordan (2014) (quoted in Chris Manning, "Computational Linguistics and Deep Learning" )

Sequence-to-Sequence is pretty convincing

- Machine Translation (?????)
- Question Answering (?)
- Sentence Compression (?)
- Parsing (?)
- *Summarization* (?)
- Conversation (?)
- Argument Generation (?)
- *Grammar Correction* (?)
- Speech (?)
- Caption Generation (?)
- Video-to-Text (?)

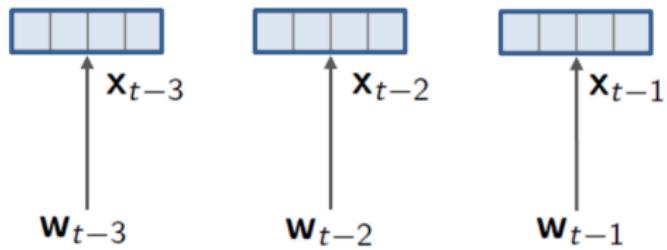
## Seq2Seq Neural Network Toolbox

Embeddings      sparse features       $\Rightarrow$       dense features

RNNs      feature sequences       $\Rightarrow$       dense features

Softmax      dense features       $\Rightarrow$       discrete predictions

Embeddings      sparse features     $\Rightarrow$     dense features



police

March

expected

group

state

made

network

city

group

first

create

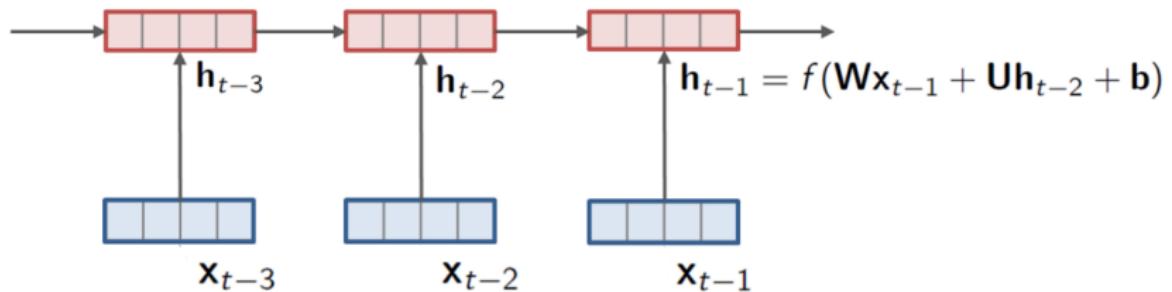
March

funding

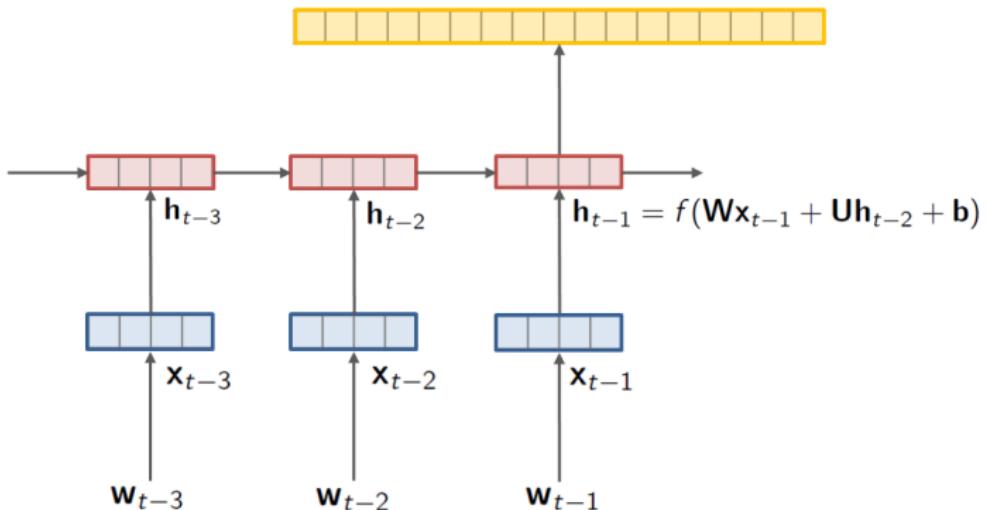
percent

author

RNNs/LSTMs      feature sequences     $\Rightarrow$     dense features



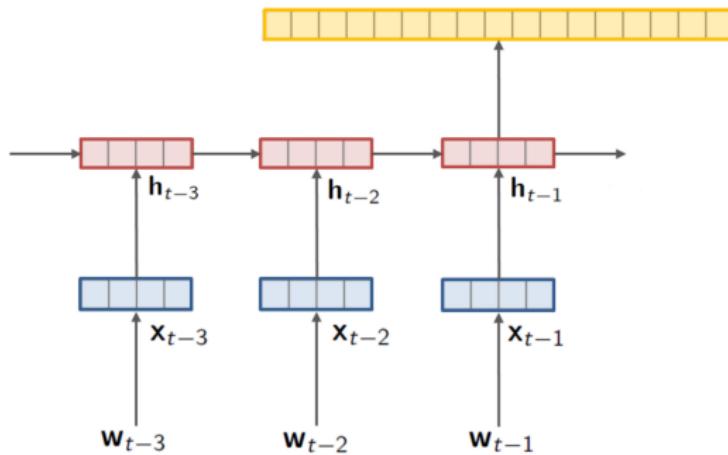
Softmax      dense features     $\Rightarrow$     discrete predictions



$$p(\mathbf{w}_t | \mathbf{w}_1, \dots, \mathbf{w}_{t-1}; \theta) = \text{softmax}(\mathbf{W}_{out} \mathbf{h}_{t-1} + \mathbf{b}_{out})$$

$$p(\mathbf{w}_{1:T}) = \prod_t p(\mathbf{w}_t | \mathbf{w}_1, \dots, \mathbf{w}_{t-1})$$

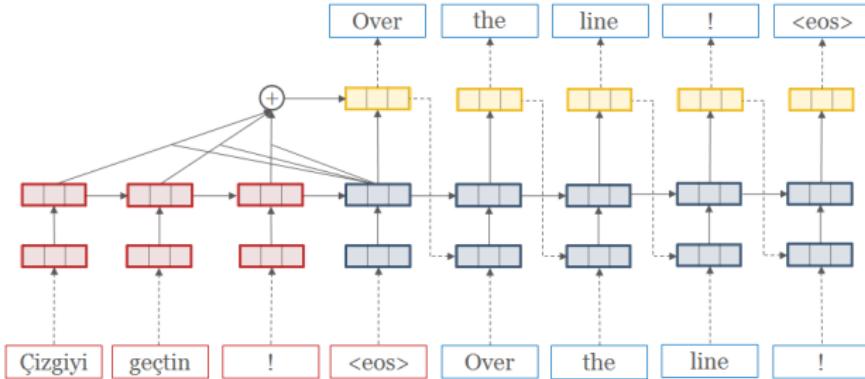
## Contextual Language Model / "seq2seq"



- Key idea, contextual language model based on encoder  $\mathbf{c}$ :

$$p(\mathbf{w}_{1:T} | \mathbf{c}) = \prod_t p(\mathbf{w}_t | \mathbf{w}_1, \dots, \mathbf{w}_{t-1}, \mathbf{c})$$

## Actual Seq2Seq / Encoder-Decoder / Attention-Based Models



- Different encoders, attention mechanisms, input feeding, ...
- Almost all models use LSTMs or other gated RNNs
- Large multi-layer networks necessary for good performance.
  - 4 layer, 1000 hidden dims is common for MT

## Seq2Seq Applications: Sentence Summarization (?)

### Source

*Russian Defense Minister Ivanov called Sunday for the creation of a joint front for combating global terrorism.*

### Target

*Russia calls for joint front against terrorism.*

- Used by The Washington Post to suggest headlines (?)

## Seq2Seq Applications: Sentence Summarization (?)

### Source

*Russian Defense Minister Ivanov called Sunday for the creation of a joint front **for combating** global terrorism.*

### Target

*Russia calls for joint front **against** terrorism.*

- Used by The Washington Post to suggest headlines (?)

## **Seq2Seq Applications: Grammar Correction (?)**

### **Source**

*There is no **a doubt**, tracking **systems has** brought many benefits in this information age .*

### **Target**

*There is no doubt, tracking systems have brought many benefits in this information age .*

- First-place on BEA 11 grammar correction shared task (?)

## Seq2Seq Applications: Grammar Correction (?)

### Source

*There is no **a doubt**, tracking **systems has** brought many benefits in this information age .*

### Target

*There is no doubt, tracking systems have brought many benefits in this information age .*

- First-place on BEA 11 grammar correction shared task (?)

## This Talk

- How can we **interpret** these learned hidden representations?
- How should we **train** these style of models?
- How can we **shrink** these models for practical applications?

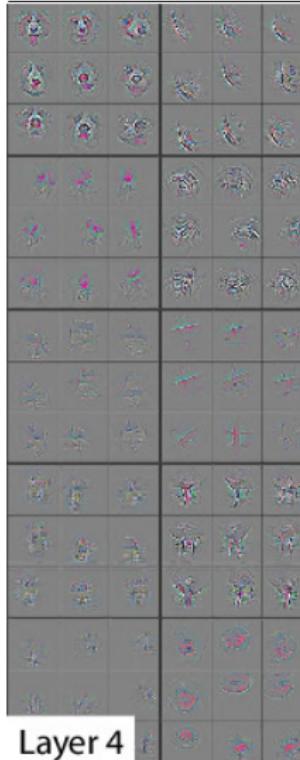
## This Talk

- How can we **interpret** these learned hidden representations?

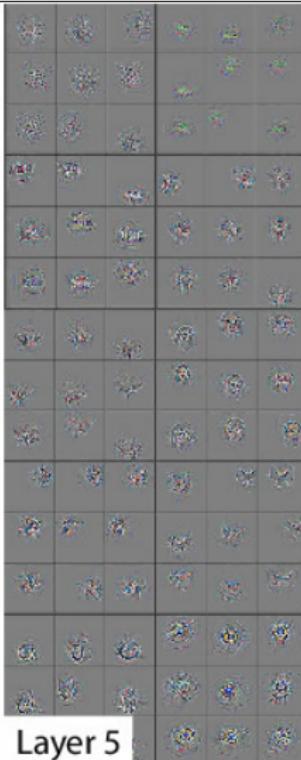
LSTMVis

(?)

- How should we **train** these style of models? (?)
- How can we **shrink** these models for practical applications? (?)



Layer 4

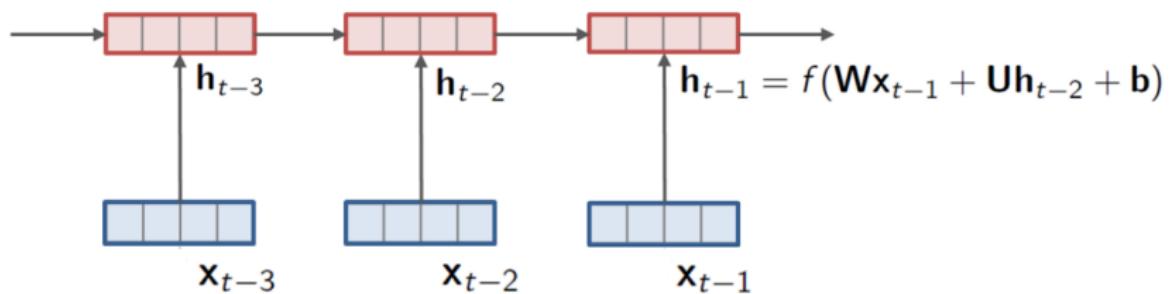
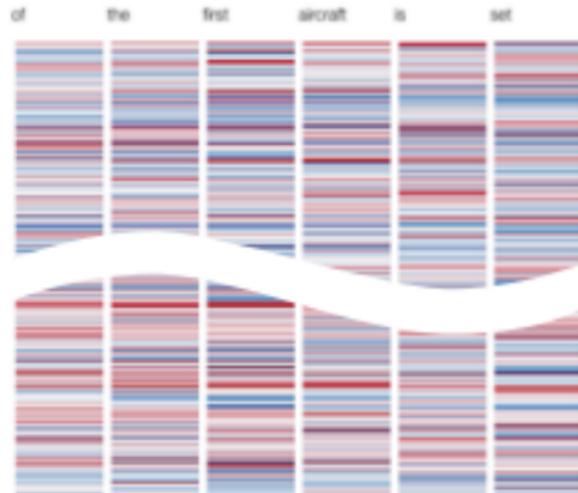


Layer 5



(?)

# Vector-Space RNN Representation



ttp://www.ynetnews.com/ ] English-language website of Israel's largest English-language site of Israeli sing  
 tp://www.bacahets.com/ -english languages airsite of tsraelis sing  
 d : xne.waea..awatoa. s &ntiaca-sardeelh oanbtisanfanreif' aatd  
 mw-2 ♦pilisoessis./ern.c](deen epesaaki eledh,irthraonse, cose  
 dr <:ahb-nptwt.xi gh/ma) Tvdryzi couedisu:tha-oo tu,stui lveper  
 stp.tcoa2drulwoclensr] p. Ilvaod., eytc-n dm-oibuv[s] bblimsultalybn

gest newspaper ' [[ Yedioth Ahronoth ]] ' ' Hebrew-language period  
 et aawsaperso [ Tel i(feanemt) ' ' ' [errewsleanguage: arosodi  
 irscoe ena iTThAoainnSrmuw] ey s [ 'ineia'siwdde'hsolrifr:  
 us.setlgor s.asatCareeg' aClrisz] ie': :, #: Taaaat Baseeiloiianfvl  
 - tuaevrtid,tBAmSusyut] Asaoigs]], . . . sMBolous: Toua-n: dwoapnu  
 ad, iiuiticp. ] (SvHvtusuiDnoegano . . . ]:{ CCuiboheCybksls:r-epcnts

icals: ' ' ' [[ Globes ]] ' ' [ http://www.globes.co.il/ ] business da  
 cal: ' ' ' ' [ Taaba ] ' ' ([ http://www.buobal.comun/sA-ytinessaet  
 stl' [ hAeovelt sahad :xge.woirrtoael.iT&ai leg eoy  
 tt' &[ &&mCoerone': .i'odw.,: niiisaue.eni/omcC.(eftgir iiu  
 a'n:, C:&:#\*: afDrusu]!, .omel p<, dha; deuoot/ihncsifS, urhos t, tun  
 nk i <]: &11s TGuiltrsi, :bacmr-xtpob-gresislerlnafad] losptad, ifrm

illy ' ' [[ Haaretz | Ha'Aratz ] ] ' ' [ http://www.haaretz.co.il/ ] Relativ  
 ly \* [ [ Terrdn Ferantah ] ] ' ' ([ http://www.bonmdst.comun/s-eateoi  
 re ' ' hAilnntteHalsrcnol' sahad :xne.waamrt dheoh. ol.c &opinive  
 ki : \* CO Sanlt hitim' lie : , imcdw-2 ♦phi iserdit. ina/cmfi. (aflcana  
 ds-! [ tBTCommgd ] Won aae, : baerr. <taib-dulcnnc/arnesi ] liceysto  
 nds#&: GI Duvccsaosucltel] zl, : o'omt], : eo a2nivfsrooeiunala) uvvro

(?)

## Example 1: Synthetic (Finite-State) Language

alphabet: ( ) 0 1 2 3 4

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

- Numbers are randomly generated, must match nesting level.
  - Train a predict-next-word language model (decoder-only).

$$p(\mathbf{w}_t | \mathbf{w}_1, \dots, \mathbf{w}_{t-1})$$

## [Parens Example]

## Example 2: Real Language

**alphabet:** all english words

**corpus:** Project Gutenberg Children's books

- Train a predict-next-word language model (decoder-only).

$$p(\mathbf{w}_t | \mathbf{w}_1, \dots, \mathbf{w}_{t-1})$$

[LM Example]

### Example 3: Seq2Seq Encoder

**alphabet:** all english words

**corpus:** Summarization

- Train a full seq2seq model, examine *encoder* LSTM.

[Summarization Example]

## This Talk

- How can we **interpret** these learned hidden representations? (?)
- How should we **train** these style of models?

Sequence-to-Sequence Learning as Beam-Search Optimization

(?)

- How can we **shrink** these models for practical applications (?)?

## Some More Seq2Seq Details

Training Objective: Multiclass NLL (for training targets  $y_{1:T}$ )

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

Test Objective: Structured output space

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

- Note: Completely intractable  $O(\#\text{vocab}^T)$

## Some More Seq2Seq Details

Training Objective: Multiclass NLL (for training targets  $y_{1:T}$ )

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

Test Objective: Structured output space

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

- Note: Completely intractable  $O(\#\text{vocab}^T)$

## Standard Approach: Beam Search

- ① Start with  $K$  partial starting hypotheses  $\mathbf{w}^{(1:K)}$
- ② For timesteps  $t$  from 1 to  $T$ :

- ① Compute for all  $k, \mathbf{w}_t$

$$s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)}) \leftarrow \log p(\mathbf{w}_t | \mathbf{w}_{1:t-1}^{(k)}, \mathbf{c}) + \log p(\mathbf{w}_{1:t-1}^{(k)} | \mathbf{c})$$

- ② Replace the  $K$  highest scoring target sequences

$$\mathbf{w}_{1:t}^{(1:K)} \leftarrow K \arg \max_{\mathbf{w}_{1:t}} s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)})$$

## Standard Approach: Beam Search

- ① Start with  $K$  partial starting hypotheses  $\mathbf{w}^{(1:K)}$
- ② For timesteps  $t$  from 1 to  $T$ :
  - ① Compute for all  $k$ ,  $\mathbf{w}_t$

$$s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)}) \leftarrow \log p(\mathbf{w}_t | \mathbf{w}_{1:t-1}^{(k)}, \mathbf{c}) + \log p(\mathbf{w}_{1:t-1}^{(k)} | \mathbf{c})$$

- ② Replace the  $K$  highest scoring target sequences

$$\mathbf{w}_{1:t}^{(1:K)} \leftarrow K \arg \max_{\mathbf{w}_{1:t}} s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)})$$

## Standard Approach: Beam Search

- ① Start with  $K$  partial starting hypotheses  $\mathbf{w}^{(1:K)}$
- ② For timesteps  $t$  from 1 to  $T$ :
  - ① Compute for all  $k$ ,  $\mathbf{w}_t$

$$s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)}) \leftarrow \log p(\mathbf{w}_t | \mathbf{w}_{1:t-1}^{(k)}, \mathbf{c}) + \log p(\mathbf{w}_{1:t-1}^{(k)} | \mathbf{c})$$

- ② Replace the  $K$  highest scoring target sequences

$$\mathbf{w}_{1:t}^{(1:K)} \leftarrow K \arg \max_{\mathbf{w}_{1:t}} s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)})$$

## Standard Approach: Beam Search

- ① Start with  $K$  partial starting hypotheses  $\mathbf{w}^{(1:K)}$
- ② For timesteps  $t$  from 1 to  $T$ :
  - ① Compute for all  $k$ ,  $\mathbf{w}_t$

$$s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)}) \leftarrow \log p(\mathbf{w}_t | \mathbf{w}_{1:t-1}^{(k)}, \mathbf{c}) + \log p(\mathbf{w}_{1:t-1}^{(k)} | \mathbf{c})$$

- ② Replace the  $K$  highest scoring target sequences

$$\mathbf{w}_{1:t}^{(1:K)} \leftarrow K \arg \max_{\mathbf{w}_{1:t}} s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)})$$

## Beam Search Example ( $K = 3$ )



For timesteps  $t$  from 1 to  $T$ :

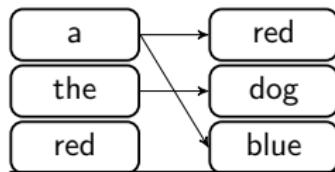
- ① Compute for all  $k, \mathbf{w}_t$

$$s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)}) \leftarrow \log p(\mathbf{w}_t | \mathbf{w}_{1:t-1}^{(k)}, \mathbf{c}) + \log p(\mathbf{w}_{1:t-1}^{(k)} | \mathbf{c})$$

- ② Replace the  $K$  highest scoring target sequences

$$\mathbf{w}_{1:t}^{(1:K)} \leftarrow K \arg \max_{\mathbf{w}_{1:t}} s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)})$$

## Beam Search Example ( $K = 3$ )



For timesteps  $t$  from 1 to  $T$ :

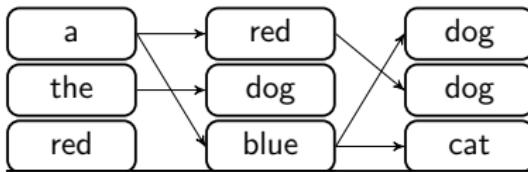
- ① Compute for all  $k, \mathbf{w}_t$

$$s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)}) \leftarrow \log p(\mathbf{w}_t | \mathbf{w}_{1:t-1}^{(k)}, \mathbf{c}) + \log p(\mathbf{w}_{1:t-1}^{(k)} | \mathbf{c})$$

- ② Replace the  $K$  highest scoring target sequences

$$\mathbf{w}_{1:t}^{(1:K)} \leftarrow K \arg \max_{\mathbf{w}_{1:t}} s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)})$$

## Beam Search Example ( $K = 3$ )



For timesteps  $t$  from 1 to  $T$ :

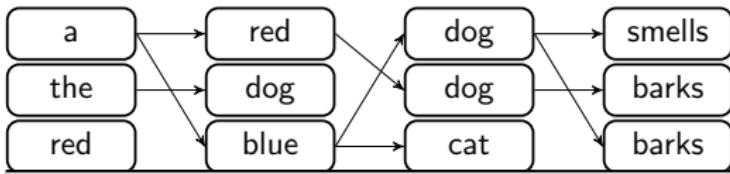
- ① Compute for all  $k, \mathbf{w}_t$

$$s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)}) \leftarrow \log p(\mathbf{w}_t | \mathbf{w}_{1:t-1}^{(k)}, \mathbf{c}) + \log p(\mathbf{w}_{1:t-1}^{(k)} | \mathbf{c})$$

- ② Replace the  $K$  highest scoring target sequences

$$\mathbf{w}_{1:t}^{(1:K)} \leftarrow K \arg \max_{\mathbf{w}_{1:t}} s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)})$$

## Beam Search Example ( $K = 3$ )



For timesteps  $t$  from 1 to  $T$ :

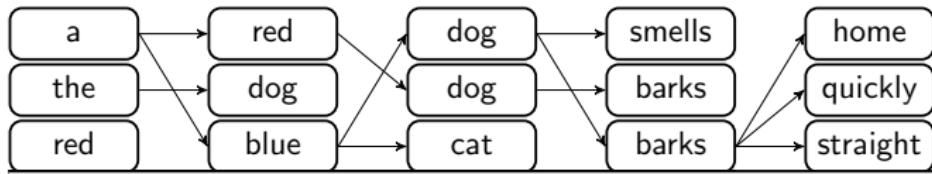
- ① Compute for all  $k, \mathbf{w}_t$

$$s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)}) \leftarrow \log p(\mathbf{w}_t | \mathbf{w}_{1:t-1}^{(k)}, \mathbf{c}) + \log p(\mathbf{w}_{1:t-1}^{(k)} | \mathbf{c})$$

- ② Replace the  $K$  highest scoring target sequences

$$\mathbf{w}_{1:t}^{(1:K)} \leftarrow K \arg \max_{\mathbf{w}_{1:t}} s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)})$$

## Beam Search Example ( $K = 3$ )



For timesteps  $t$  from 1 to  $T$ :

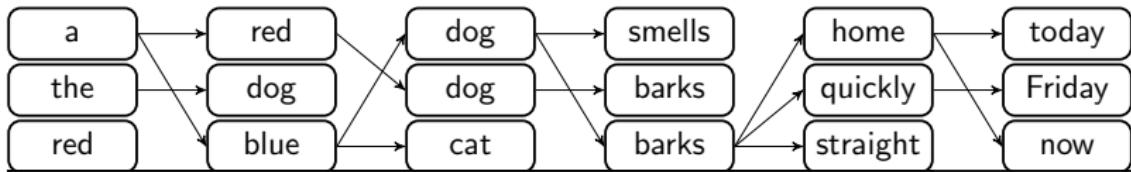
- ① Compute for all  $k, \mathbf{w}_t$

$$s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)}) \leftarrow \log p(\mathbf{w}_t | \mathbf{w}_{1:t-1}^{(k)}, \mathbf{c}) + \log p(\mathbf{w}_{1:t-1}^{(k)} | \mathbf{c})$$

- ② Replace the  $K$  highest scoring target sequences

$$\mathbf{w}_{1:t}^{(1:K)} \leftarrow K \arg \max_{\mathbf{w}_{1:t}} s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)})$$

## Beam Search Example ( $K = 3$ )



For timesteps  $t$  from 1 to  $T$ :

- ① Compute for all  $k, \mathbf{w}_t$

$$s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)}) \leftarrow \log p(\mathbf{w}_t | \mathbf{w}_{1:t-1}^{(k)}, \mathbf{c}) + \log p(\mathbf{w}_{1:t-1}^{(k)} | \mathbf{c})$$

- ② Replace the  $K$  highest scoring target sequences

$$\mathbf{w}_{1:t}^{(1:K)} \leftarrow K \arg \max_{\mathbf{w}_{1:t}} s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)})$$

## Theoretical **Issues** with Standard Setup

- Exposure Bias

- Training by conditioning on true  $y_{1:t-1}$ ,

$$p(\mathbf{w}_t = y_t | \mathbf{w}_{1:t-1} = y_{1:t-1}, \mathbf{c}; \theta)$$

- Train/Test Loss Mismatch

- Training with local NLL, evaluate with hamming-style losses (BLEU)

- Label Bias (?)

- Locally normalized models have known pathological issues

Related Work: Modify training data

- Data as Demonstrator (?), Scheduled Sampling (?)

Related Work: Use Reinforcement Learning

- MIXER (?)
- Actor-Critic (?)

Opinion:

- DAD methods only address exposure bias,
- RL is too strong a hammer .

Related Work: Modify training data

- Data as Demonstrator (?), Scheduled Sampling (?)

Related Work: Use Reinforcement Learning

- MIXER (?)
- Actor-Critic (?)

Opinion:

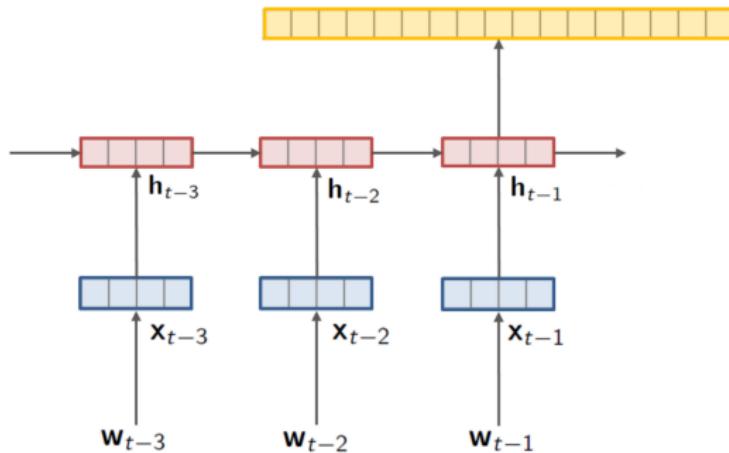
- DAD methods only address exposure bias,
- RL is too strong a hammer .

## Our Proposal: Seq2Seq as Beam Search Optimization

New Setup: Run beam search at training.

- (Idea 1) Replace local softmax with sequence scorer  $f$
- (Idea 2) Run beam search during training time
- (Idea 3) Replace local training objective with beam-search margin

(Idea 1) Replace local softmax with sequence scorer  $f$



Same model, but replace  $\log p(\mathbf{w}_t | \mathbf{w}_{1:t-1}^{(k)}, \mathbf{c}; \theta)$  with unnormalized  $f(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)}, \mathbf{c}; \theta)$

(Idea 2) Run beam search during training

- ① Start with  $K$  partial starting hypotheses  $\mathbf{w}^{(1:K)}$
- ② For timesteps  $t$  from 1 to  $T$ :
  - ① Compute for all  $k$ ,  $\mathbf{w}_t$

$$s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)}) \leftarrow \log p(\mathbf{w}_t | \mathbf{w}_{1:t-1}^{(k)}, \mathbf{c}; \theta) + \log p(\mathbf{w}_{1:t-1}^{(k)} | \mathbf{c}; \theta)$$

- ② Replace the  $K$  highest scoring target sequences

$$\mathbf{w}_{1:t}^{(1:K)} \leftarrow K \arg \max_{\mathbf{w}_{1:t}} s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)})$$

(Idea 2) Run beam search during training

① Start with  $K$  partial starting hypotheses  $\mathbf{w}^{(1:K)}$

② For timesteps  $t$  from 1 to  $T$ :

① Compute for all  $k$ ,  $\mathbf{w}_t$

$$s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)}) \leftarrow f(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)}, \mathbf{c}; \theta)$$

② Replace the  $K$  highest scoring target sequences

$$\mathbf{w}_{1:t}^{(1:K)} \leftarrow K \arg \max_{\mathbf{w}_{1:t}} s(\mathbf{w}_t, \mathbf{w}_{1:t-1}^{(k)})$$

(Idea 3) Replace local training objective with beam-search margin

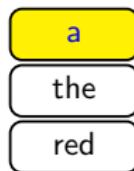
New Objective:

- Margin between target seq  $y$  and last seq on beam  $\mathbf{w}^{(K)}$

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

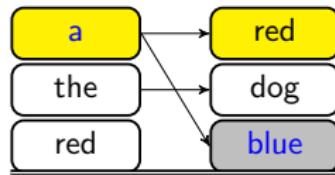
- Slack-rescaled, margin-based sequence criterion, at each time step.
- When violation occurs, target replaces current beam (learning as search optimization (?))

## Beam Search Optimization Example ( $K = 3$ )



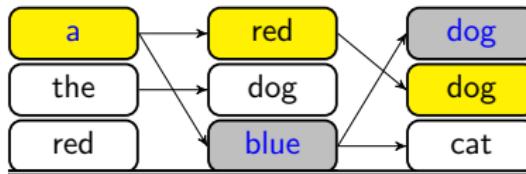
- Color Gold: target sequence  $y$
- Color Gray: violating sequence  $\mathbf{w}^{(K)}$

## Beam Search Optimization Example ( $K = 3$ )



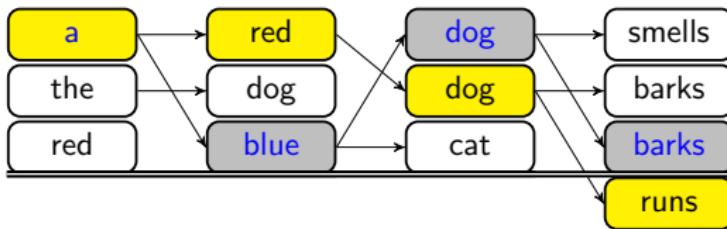
- Color Gold: target sequence  $y$
- Color Gray: violating sequence  $w^{(K)}$

## Beam Search Optimization Example ( $K = 3$ )



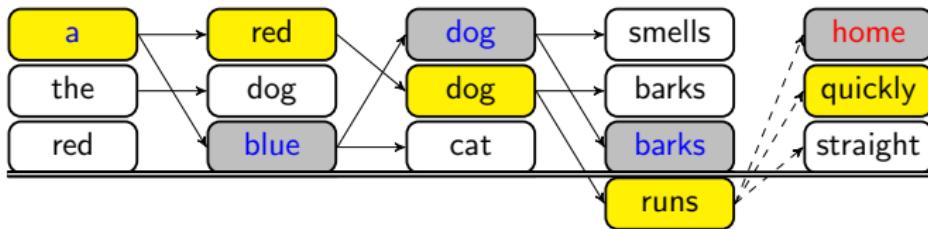
- Color **Gold**: target sequence  $y$
- Color **Gray**: violating sequence  $w^{(K)}$

## Beam Search Optimization Example ( $K = 3$ )



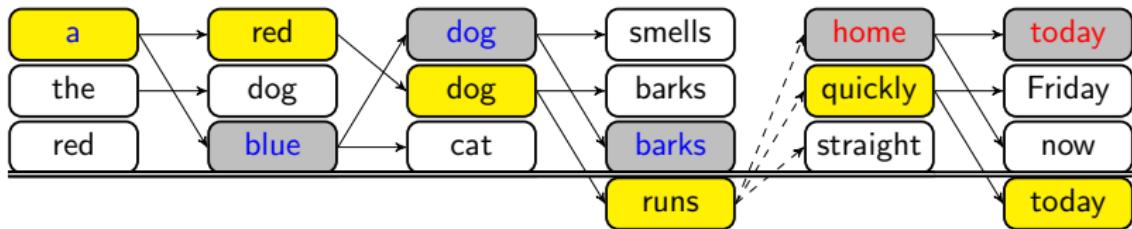
- Color **Gold**: target sequence  $y$
- Color **Gray**: violating sequence  $\mathbf{w}^{(K)}$

## Beam Search Optimization Example ( $K = 3$ )



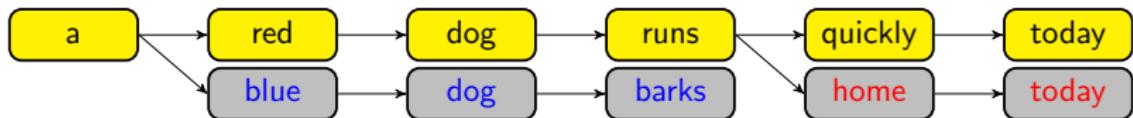
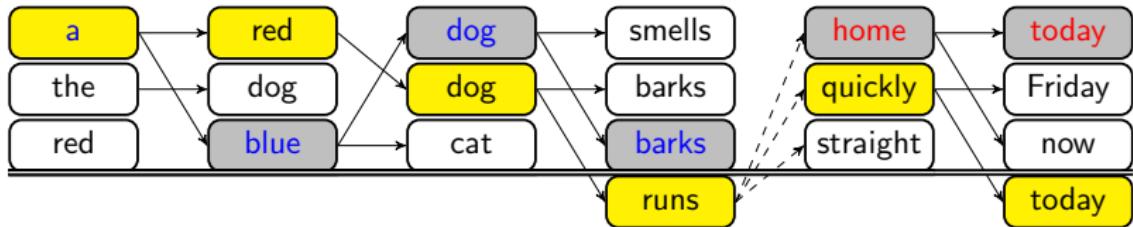
- Color **Gold**: target sequence  $y$
- Color **Gray**: violating sequence  $w^{(K)}$

## Beam Search Optimization Example ( $K = 3$ )



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

## Structured Backpropagation



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

## Theoretical **Issues** with Standard Setup

- 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

## Experiments

Experiments run on three different seq2seq baseline tasks

- Word Ordering
- Dependency Parsing
- Machine Translation

Details:

- Utilize our *seq2seq-attn* code, very strong attention-based system
- Pretrained with NLL.
- Trained with a curriculum to gradually increase beam size.

	$K_e = 1$	$K_e = 5$	$K_e = 10$
Word Ordering (BLEU)			
seq2seq	25.2	29.8	31.0
BSO	28.0	33.2	34.3
BSO-Con	<b>28.6</b>	<b>34.3</b>	<b>34.5</b>
Dependency Parsing (UAS/LAS)			
seq2seq	<b>87.33/82.26</b>	88.53/84.16	88.66/84.33
BSO	86.91/82.11	91.00/ <b>87.18</b>	91.17/ <b>87.41</b>
BSO-Con	85.11/79.32	<b>91.25</b> /86.92	<b>91.57</b> /87.26
Machine Translation (BLEU)			
seq2seq	22.53	24.03	23.87
BSO, SB- $\Delta$ , $K_t=6$	<b>23.83</b>	<b>26.36</b>	<b>25.48</b>
XENT	17.74	$\leq 20.5$	$\leq 20.5$
DAD	20.12	$\leq 22.5$	$\leq 23.0$
MIXER	20.73	-	$\leq 22.0$

## This Talk

- How can we **interpret** these learned hidden representations? (?)
- How should we **train** these style of models? (?)
- How can we **shrink** these models for practical applications?

Sequence-Level Knowledge Distillation

(?)

## Seq2Seq In Practice

### Benefits

- Very accurate
- General purpose
- Possibly interpretable

### Downsides

- Models are really big (MT model is 4 layers each of 1000 units)
- Beam search can be quite slow

## Related Work: Compressing Deep 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 (?)

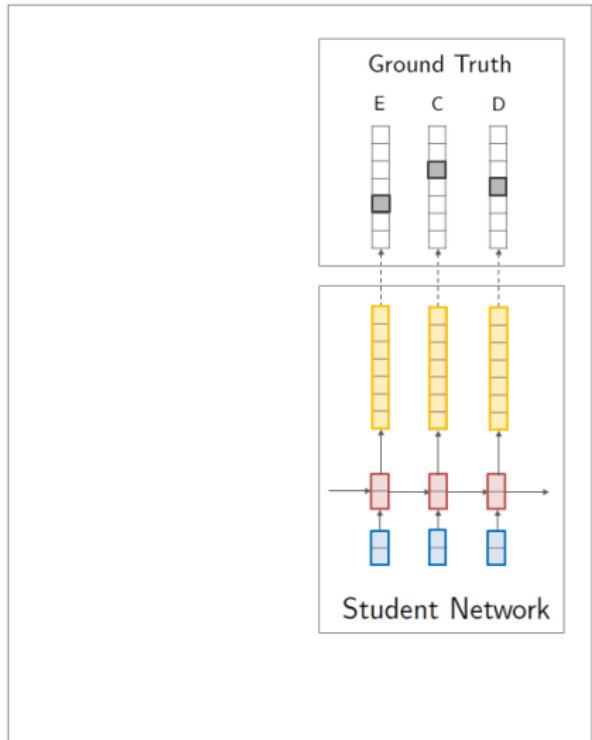
## Baseline Model

Standard model minimize  $\text{NLL}(\theta)$ :

$$-\sum_t \log p(\mathbf{w}_t = y_t | \mathbf{w}_{1:t-1}, \mathbf{c}; \theta)$$

where  $y_t$  is the ground truth word at time  $t$ .

Cross-entropy with ground truth.

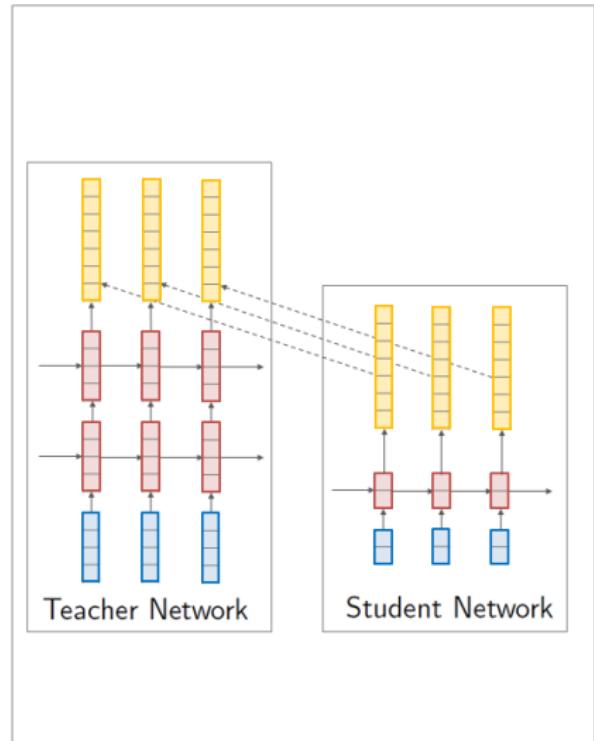


## Word-Level Knowledge Distillation

Teacher network:  $q(\mathbf{w}_t | \mathbf{w}_{1:t-1}, \mathbf{c}; \theta_T)$

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

$$-\sum_t \sum_v q(\mathbf{w}_t = v | \mathbf{w}_{1:t-1}, \mathbf{c}; \theta_T) \times \\ \log p(\mathbf{w}_t = v | \mathbf{w}_{1:t-1}, \mathbf{c}; \theta)$$



## This Work: Sequence-Level Knowledge Distillation

Instead of word NLL,

$$-\sum_t \sum_v q(\mathbf{w}_t = v \mid \mathbf{w}_{1:t-1}, \mathbf{c}; \theta_T) \times \log p(\mathbf{w}_t = v \mid \mathbf{w}_{1:t-1}, \mathbf{c}; \theta)$$

Minimize cross-entropy between  $q$  and  $p$  implied sequence-distributions

$$-\sum_{\mathbf{w}_{1:T}} q(\mathbf{w}_{1:T} \mid \mathbf{c}; \theta_T) \times \log p(\mathbf{w}_{1:T} \mid \mathbf{c}; \theta)$$

Note: Exponential sum over possible  $\mathbf{w}_{1:T}$ .

## A Simple Approximation

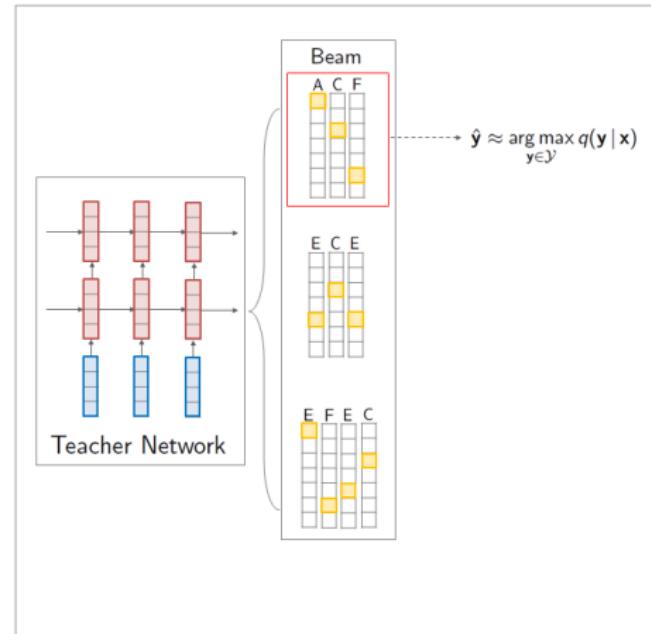
Approximate  $q(\mathbf{w}_{1:T} | \mathbf{c})$  with mode

$$q(\mathbf{w}_{1:T} | \mathbf{c}) \approx \mathbf{1}\{\arg \max_{\mathbf{w}} q(\mathbf{w}_{1:T} | \mathbf{c})\}$$

Roughly obtained with beam search

$$\mathbf{w}_{1:T}^* \approx \arg \max_{\mathbf{w}_{1:T}} q(\mathbf{w}_{1:T} | \mathbf{c})$$

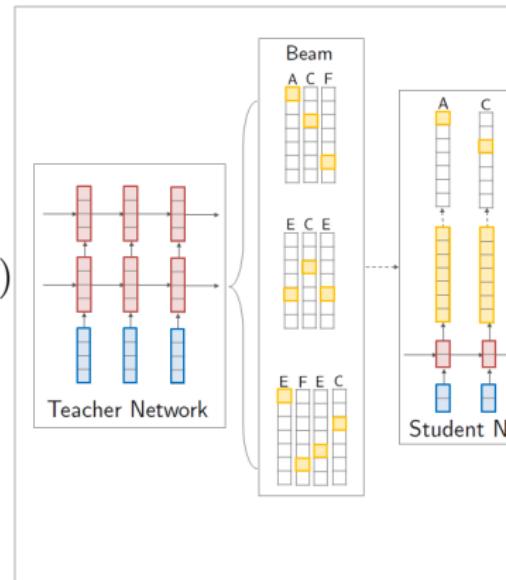
Empirically, point estimate captures significant mass



## Sequence-Level Knowledge Distillation

$$\begin{aligned}\mathcal{L}_{\text{SEQ-KD}}(\theta) &= -\log p(\mathbf{w}_{1:T}^* \mid \mathbf{c}; \theta) \\ &\approx - \sum_{\mathbf{w}_{1:T}} q(\mathbf{w}_{1:T} \mid \mathbf{c}; \theta_T) \log p(\mathbf{w}_{1:T} \mid \mathbf{c}; \theta)\end{aligned}$$

Simplest model: train the student model on  $\mathbf{w}^*$  with NLL



## Results: English → German

Model	$\text{BLEU}_{K=1}$	$\Delta_{K=1}$	$\text{BLEU}_{K=5}$	$\Delta_{K=5}$	PPL	$p(\mathbf{w}^*)$
$4 \times 1000$						
Teacher	17.7	—	19.5	—	6.7	1.3%
Seq-Inter	19.6	+1.9	19.8	+0.3	10.4	8.2%
$2 \times 500$						
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	<b>+4.2</b>	19.0	+1.4	22.7	16.9%
Seq-Inter	18.9	<b>+4.2</b>	19.3	<b>+1.7</b>	15.8	7.6%

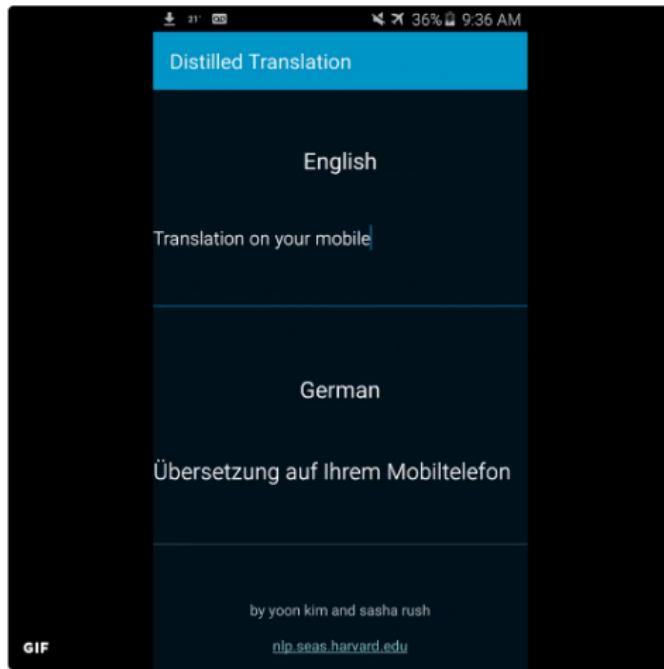
## Combining Knowledge Distillation and Pruning

Model	Prune %	Params	BLEU	Ratio
$4 \times 1000$	0%	221 m	19.5	1×
$2 \times 500$	0%	84 m	19.3	3×
$2 \times 500$	50%	42 m	19.3	5×
$2 \times 500$	80%	17 m	19.1	13×
$2 \times 500$	85%	13 m	18.8	18×
$2 \times 500$	90%	8 m	18.5	26×



harvardnlp  
@harvardnlp

Seq KD ([arxiv.org/abs/1606.07947](https://arxiv.org/abs/1606.07947)): learn small  
LSTMs for fast translation. Runs on a phone  
([nlp.seas.harvard.edu/translation.apk](http://nlp.seas.harvard.edu/translation.apk))



# Thank You



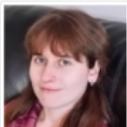
## Graduate Students



Sebastian  
Gehrmann



Yoon Kim



Victoria  
Krakovna



Allen  
Schmaltz



Sam Wiseman

## Undergraduate Researchers



Jeffrey Ling



Keyon Vafa



Alex Wang



Mike Zhai

# References I