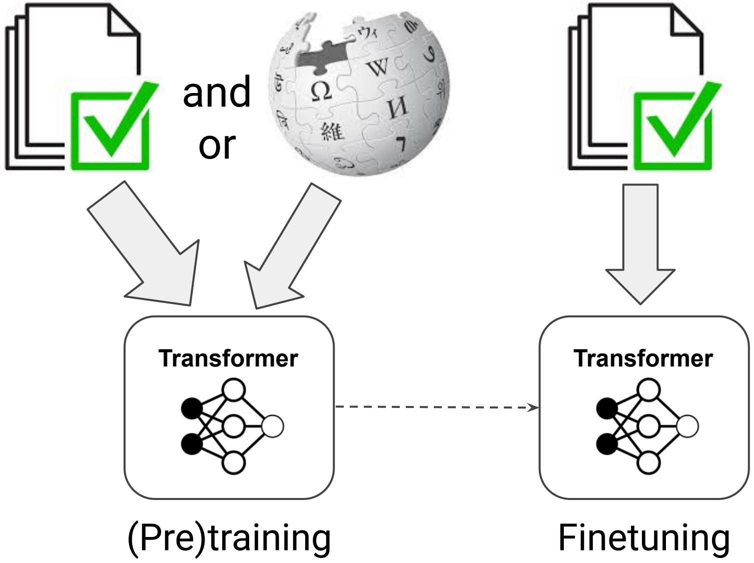
Large-context and efficient models of language

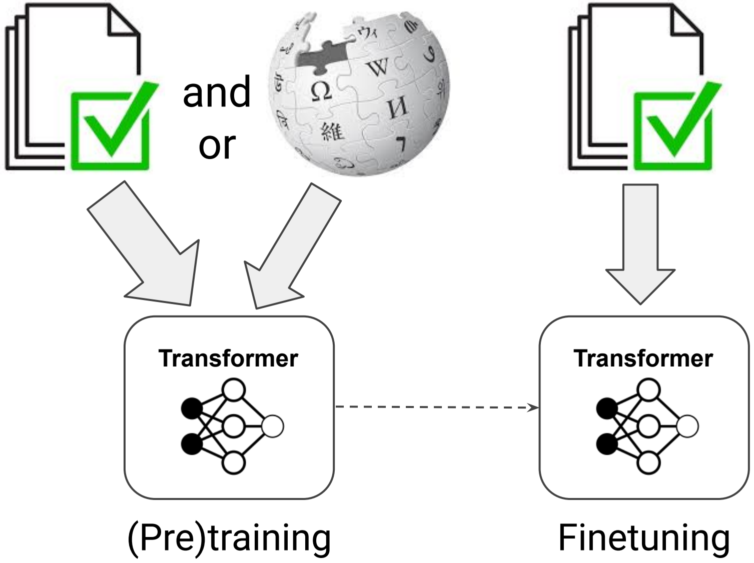
Nikolaos Pappas

Dominant paradigm



Huge impact in NLP tasks.

Dominant paradigm



- ❌ Big data requirements
- ❌ Poor on rare or new words
- ❌ Computationally expensive

Huge impact in NLP tasks.

My past research

Build models that learn from language efficiently

1

Modeling documents

Representation learning
[JAIR'17;EMNLP'18]

Multilingual transfer
[IJNLP'17,TACL'19]

Structural comparisons
[EMNLP'20]

2

Promoting data efficiency

Deep word sharing
[WMT'18;TACL'19;ICML'19]

Grounding to lexicons
[EMNLP'20]

3

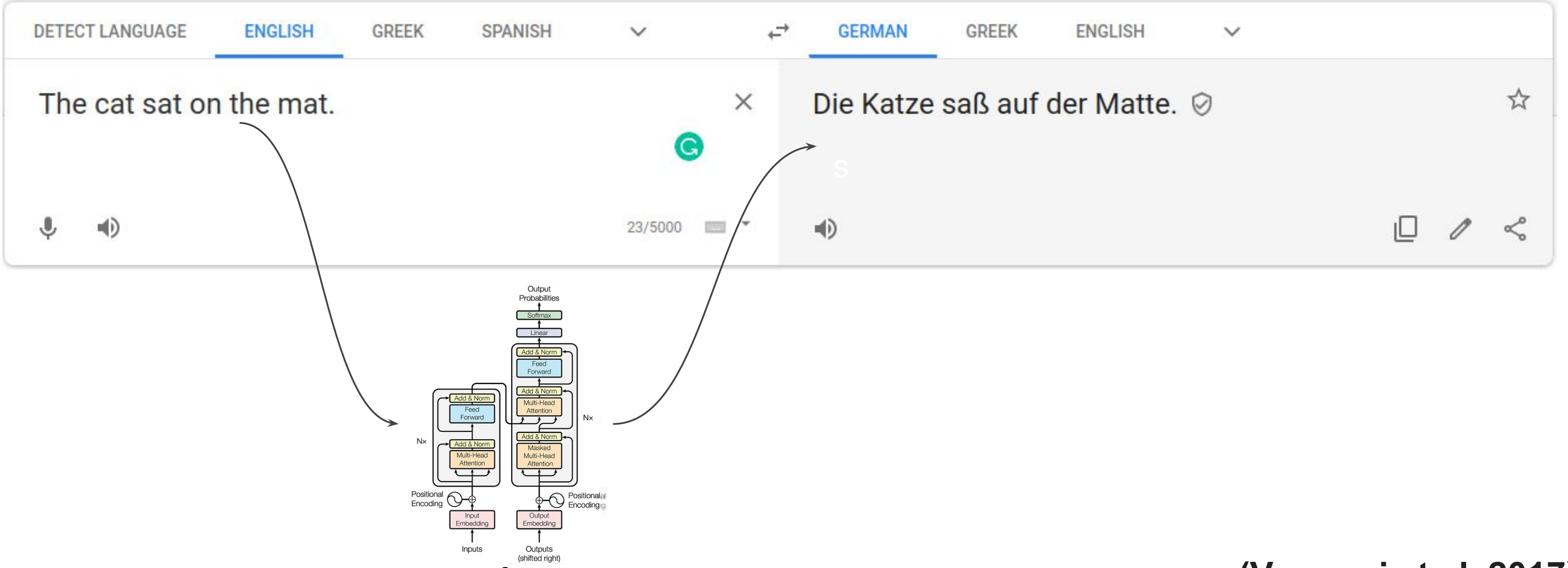
Reducing cost

Model design
[ICLR subm.]

Training objectives
[EMNLP'20]

Scalable components
[ICML'20, ICLR subm.]

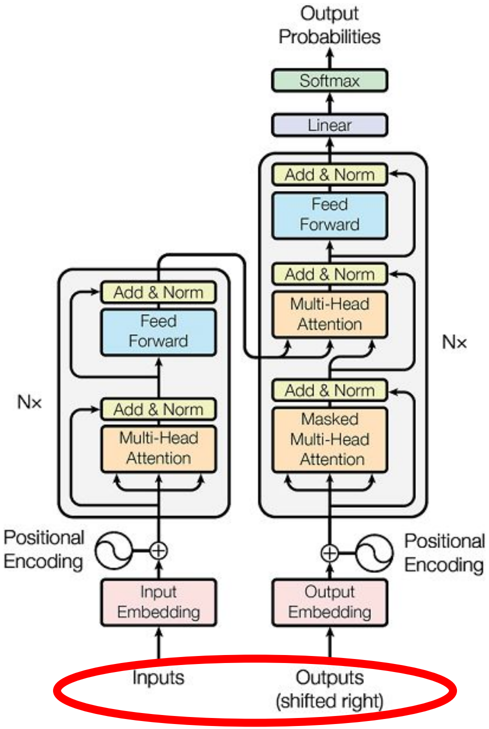
Transformer origins



Transformer

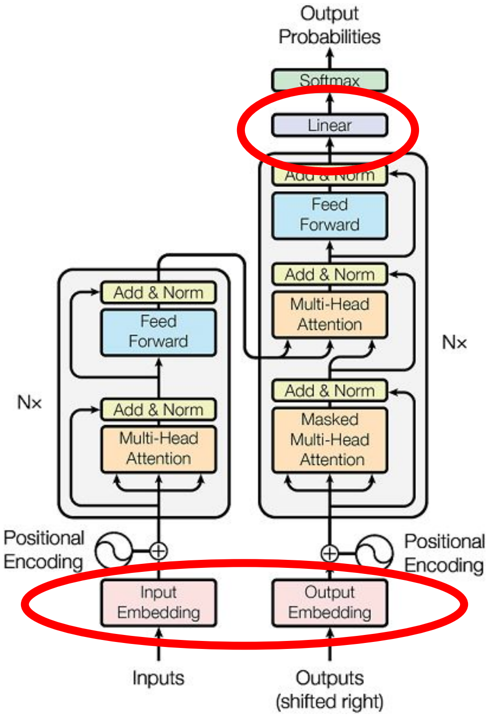
(Vaswani et al. 2017)

Limitations: Narrow context



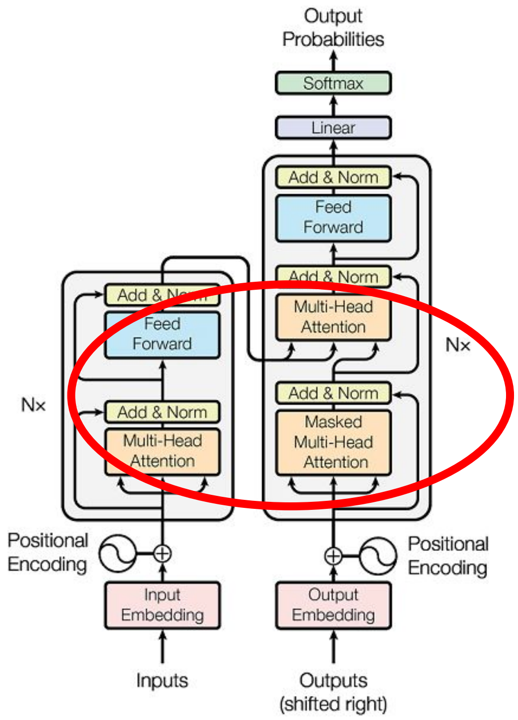
- Fixed and narrow context for prediction
- Suboptimal for document tasks

Limitations: Rigid parameterization



- Dominates model size
- Performs poorly on rare types (data hungry)
- Requires ungraceful changes for adaptation

Limitations: Quadratic complexity



- Does not scale to long text sequences
- Wastes memory for parallelization
- Slow for autoregressive inference

Overview

Objective: show ways to address these challenges in neural MT

1. Dynamic hierarchical attention [EMNLP 2018]
2. Deep word sharing and grounding [ICML 2019;EMNLP 2020]
3. Random feature attention [ICLR subm.]

Overview

0. Introduction / Motivation

You are here

1. Incorporating larger context
2. Promoting data efficiency
3. Reducing computational cost
4. Future work

Sentence-level translation

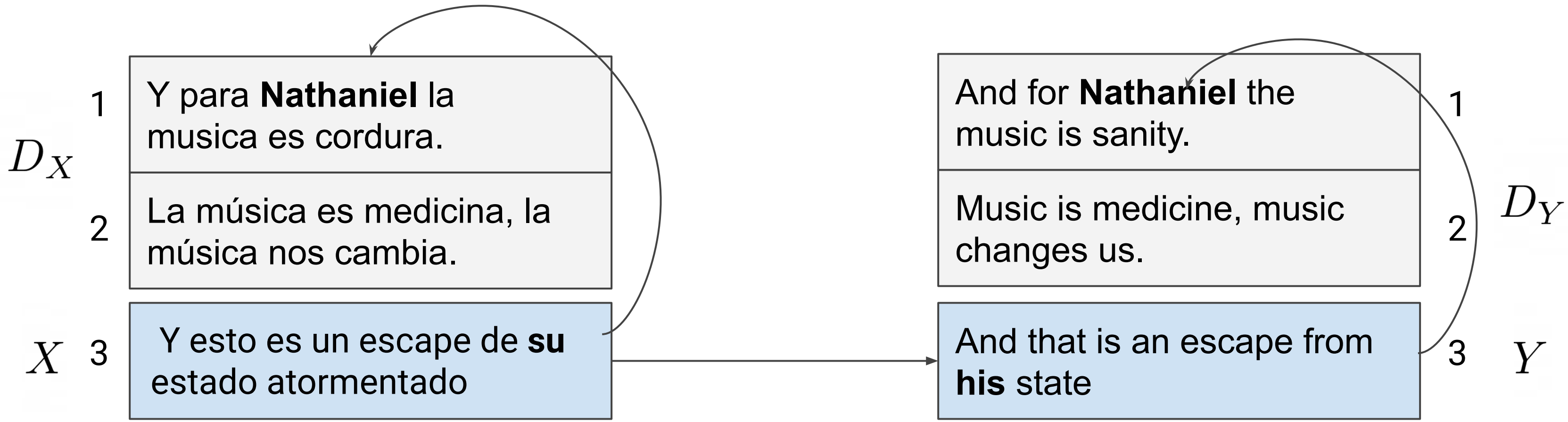
He, she or its ?

X Y esto es un escape de **su** estado atormentado

Y And that is an escape from **his** state

$$P(Y|X) = \prod_{t=1}^{T} p(y_t|y_{<t}, X)$$

Extra-sentential context to the rescue



$$P(Y|X) = \prod_{t=1}^{T} p(y_t|y_{<t}, X, D_X, D_Y)$$

Sentence context   Extra-sentential context

Previous efforts

1 Y para **Nathaniel** la musica es cordura.

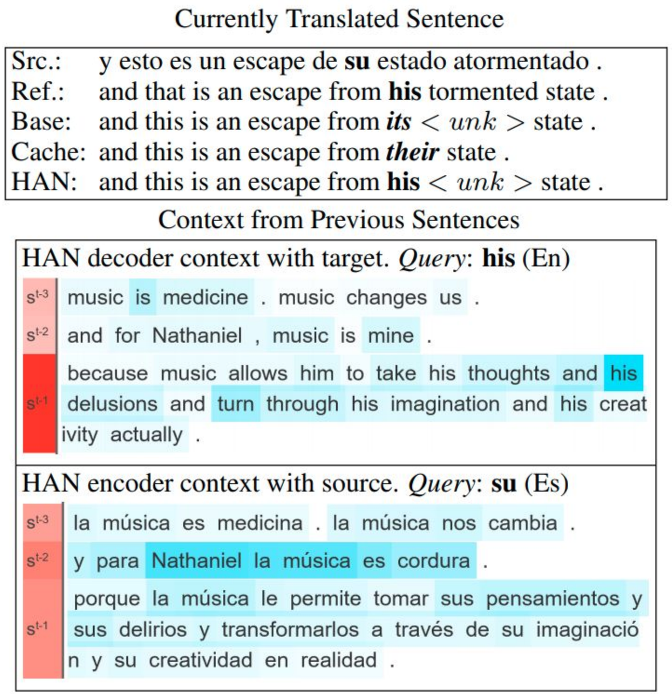
2 La música es medicina, la música nos cambia.

3 Y esto es un escape de **su** estado atormentado

- Concatenation (Tiedemann & Scherrer, 2017)
- Additional attention (Jean et al., 2017)
- Hierarchical context (Wang et al., 2017)
- Continuous cache (Tue et al., 2018)
- ... and many other recently (Voita et al., 2018; Lopes et al., 2020; Liu et al., 2020; Yu et al., 2020)

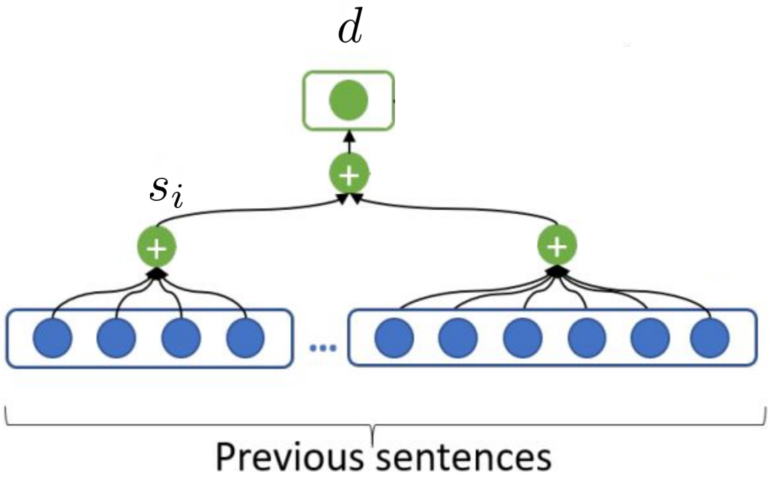
Dynamic hierarchical context [EMNLP 2018]

- Exploit source and target document context
- Compute a dynamic document context for each token
- Increased interpretability in the attention maps



Lesly Miculicich, Dhananjay Ram, **Nikolaos Pappas** and James Henderson, Document-Level Neural Machine Translation with Hierarchical Attention Networks, EMNLP 2018.

Hierarchical attention



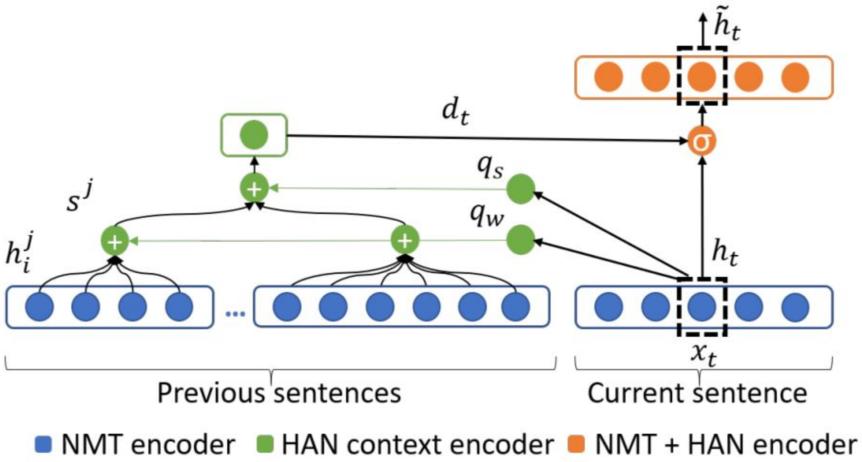
- Encoding with recurrent networks
- Pooling based on attention with a "learned context" per level

$$a_i = \frac{\texttt{exp}(s_i^\top u_s)}{\sum_j \texttt{exp}(s_j^\top u_s)}$$

$$d = \sum_i \alpha_i s_i$$

(Yang et al., 2017)

Dynamic hierarchical attention [EMNLP 2018]



- Encoding with transformer s
- Pooling based on multi-head attention conditioned on encoded tokens

$$q_s = f_s(h_t)$$

$$d_t = \mathrm{FFN}(\underset{j}{\mathrm{MultiHead}}(q_s, s^j))$$

- Context gating

$$\lambda_t = \sigma(W_h h_t + W_d d_t)$$

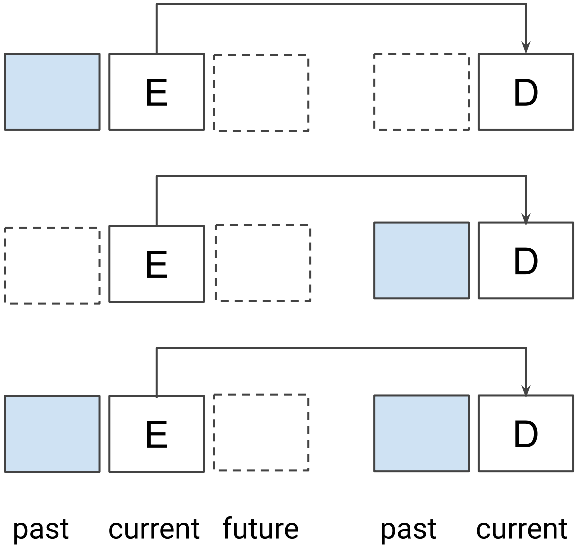
$$\widetilde{h}_t = \lambda_t h_t + (1 - \lambda_t) d_t$$

Document machine translation

- Document evaluation metrics
  - Noun/Pronoun accuracy
  - Lexical coherence: metric-based (LSA)
  - Lexical cohesion: repeated/total

- Datasets with document boundaries

TED Talks		Subtitles		News
Zh–En	Es–En	Zh–En	Es–En	Es–En
0.2M	0.2M	2.2M	4.0M	0.2M



Sentence-level results

	TED Talks		Subtitles		News
	Zh–En	Es–En	Zh–En	Es–En	Es–En
NMT transformer	16.87	35.44	28.60	35.20	21.36
+ cache	17.32 ***	36.46 ***	28.86	35.49	22.36 ***
+ HAN encoder	17.61 *** ††	36.91 *** ††	29.35 * †	35.96 * †	22.36 ***
+ HAN decoder	17.39 ***	37.01 *** †††	29.21 *	35.50	22.62 *** †††
+ HAN joint	**17.79** *** †††	**37.24** *** †††	**29.67** ** †	**36.23** ** ††	**22.76** *** †††

BLEU scores. Significance with respect to NMT ∗, and to cache model †.

Higher is better

- Significant improvement on different size datasets (up to 4M)
- Target and source context are complementary

Discourse-level results

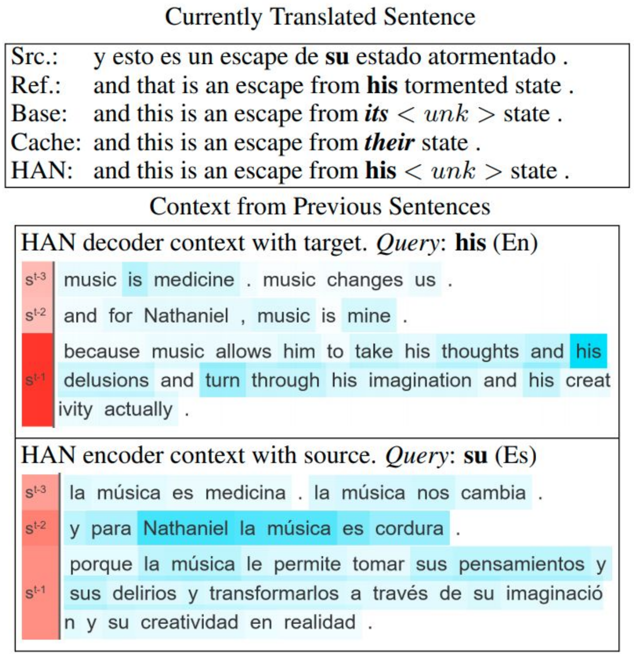
	Coherence	Lexical Cohesion	Pronouns	Nouns
NMT transformer	28.42	47.98	62.84	52.50
+ HAN encoder	28.60	48.35	64.48	53.61
+ HAN decoder	28.78	48.51	64.04	53.55
+ HAN joint	28.82	48.61	64.32	54.19
Human reference	29.79	52.94	100.0	100.0

Higher is better

- Gains across the board especially for noun/pronoun translation
- Still a big gap between human reference and translations

Takeaways [EMNLP 2018]

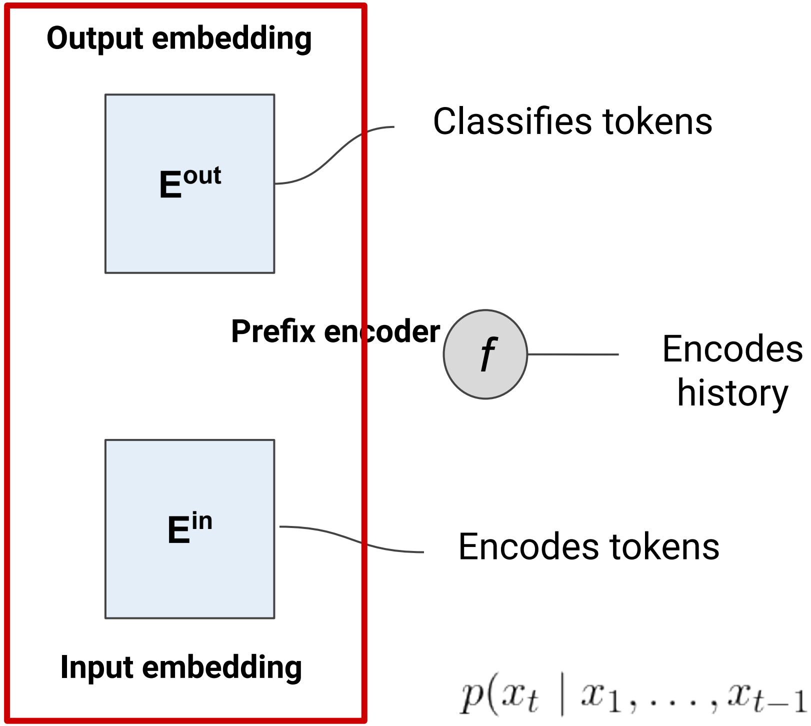
- Incorporating larger context with dynamic hierarchical attention
- Improves both sentence and discourse evaluation metrics
- Provides interpretability in the attention maps for each token



Overview

0. Introduction / Motivation
1. Broadening translation context
2. Promoting data efficiency
3. Reducing computational cost
4. Future work

Decoder language model



- Parameters are not shared across words
- Handle rare words poorly
- Cannot generalize well to new words or domains

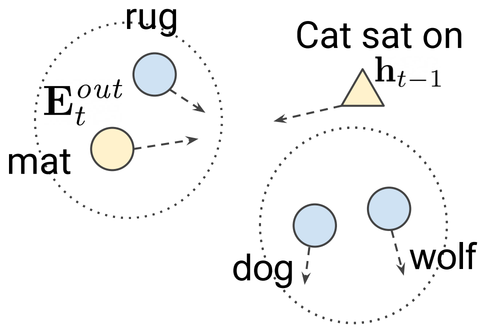
$$p(x_t \mid x_1, \ldots, x_{t-1}) \propto \exp\left(\mathbf{E}^{out}\mathbf{h}_{t-1} + \mathbf{b}\right)$$

Word sharing

- Captures better the output space similarity

- Influences word neighbors during a training update

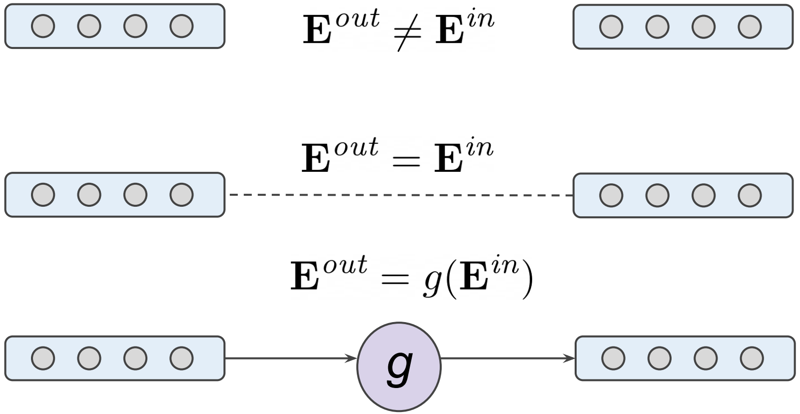
$$-\nabla \sum_{t=1}^{T} \texttt{log}\left[\frac{\texttt{exp}(\mathbf{E}_t^{out}\mathbf{h}_{t-1} + b)}{\sum_j \texttt{exp}(\mathbf{E}_j^{out}\mathbf{h}_{t-1} + b)}\right]$$



Net effect of training signal

(Pappas et al., 2018)

Representing words



	Input		Output	Word sharing	Control power
Lookup table (Zaremba et al., 2014)		$\mathbf{E}^{out} \neq \mathbf{E}^{in}$		✗	✗
Tied lookup table (Press and Wolf., 2017)		$\mathbf{E}^{out} = \mathbf{E}^{in}$		✗	✗
Functional forms (Pappas et al., 2018; Gulurdova et al., 2018;) Bilinear Dual nonlinear		$\mathbf{E}^{out} = g(\mathbf{E}^{in})$		✓	✗

- Increase power only via dim or rank which has the tendency to overfit in certain domains

Deep word sharing [ICML 2019]

Input **Output** **Word sharing**

$$\mathbf{E}_t^{out(1)} = g_0(\mathbf{E}_t^{in})$$

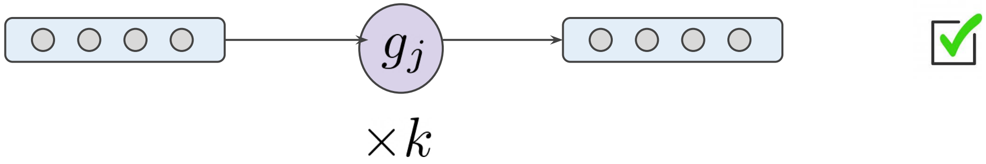
Functions $g_j(\cdot)$ are simple nonlinear transformations

Nikolaos Pappas, James Henderson, Deep Residual Output Layers for Neural Language Generation, ICML 2019.

Deep word sharing [ICML 2019]

Input **Output** **Word sharing**

$$\forall j : 1 \leq j \leq k, \mathbf{E}_t^{out\,(j)} = g_j\left(\mathbf{E}_t^{out\,(j-1)}\right)$$



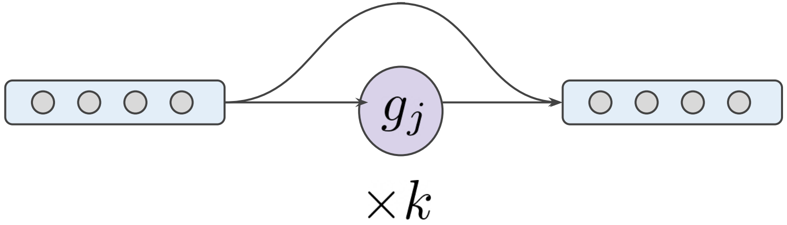
Functions $g_j(\cdot)$ are simple nonlinear transformations

Nikolaos Pappas, James Henderson, Deep Residual Output Layers for Neural Language Generation, ICML 2019.

Deep word sharing [ICML 2019]

Input **Output** **Word sharing**

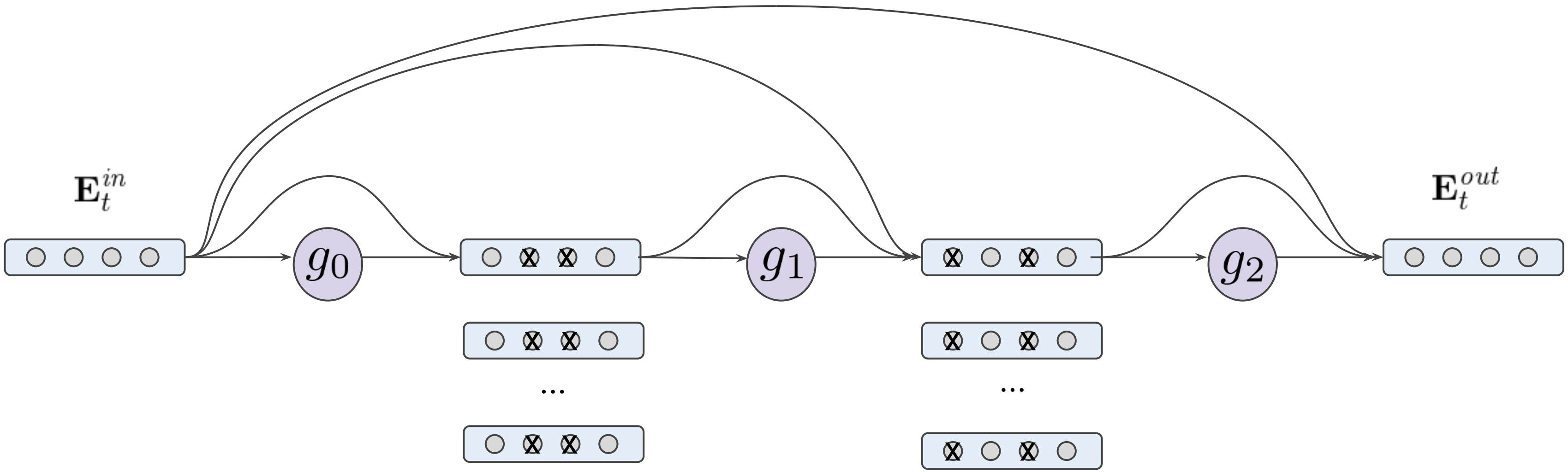
$$\forall j : 1 \leq j \leq k, \mathbf{E}_t^{out(j)} = g_j\left(\mathbf{E}_t^{out(j-1)}\right) + \mathbf{E}_t^{in}$$



Functions $g_j(\cdot)$ are simple nonlinear transformations

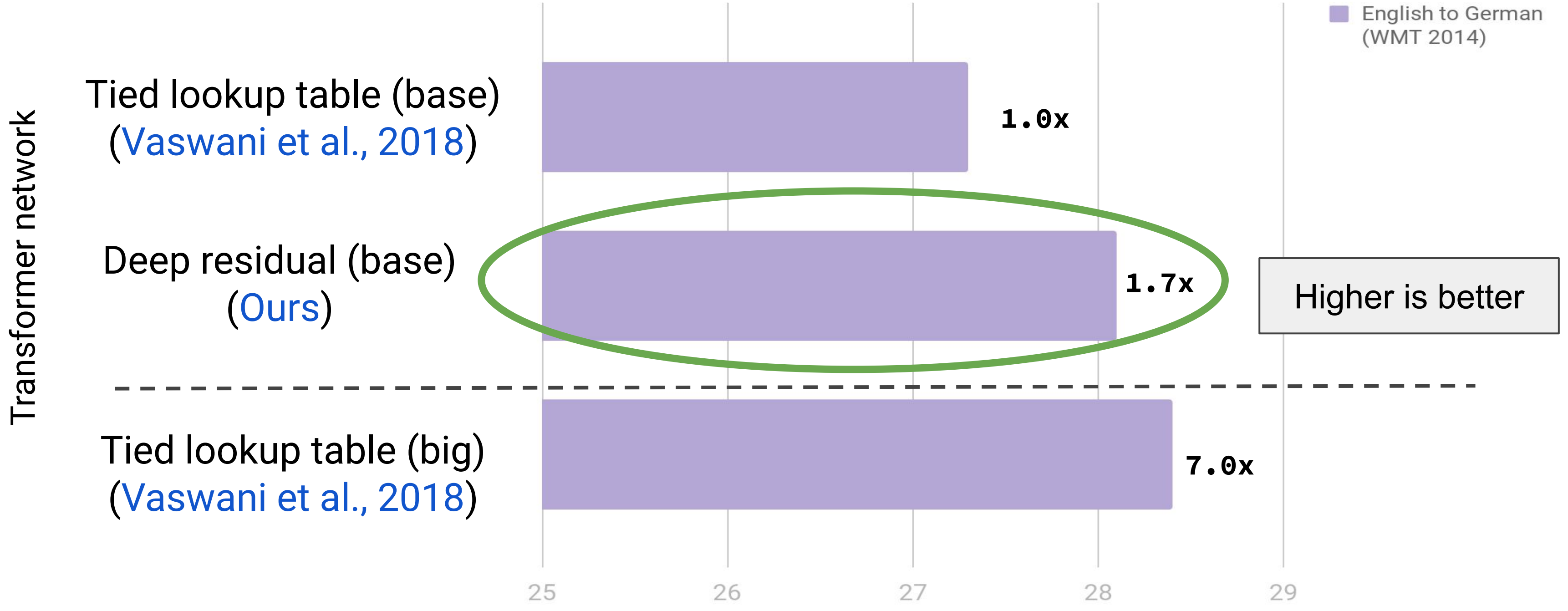
Nikolaos Pappas, James Henderson, Deep Residual Output Layers for Neural Language Generation, ICML 2019.

Unfolded version with depth k = 3

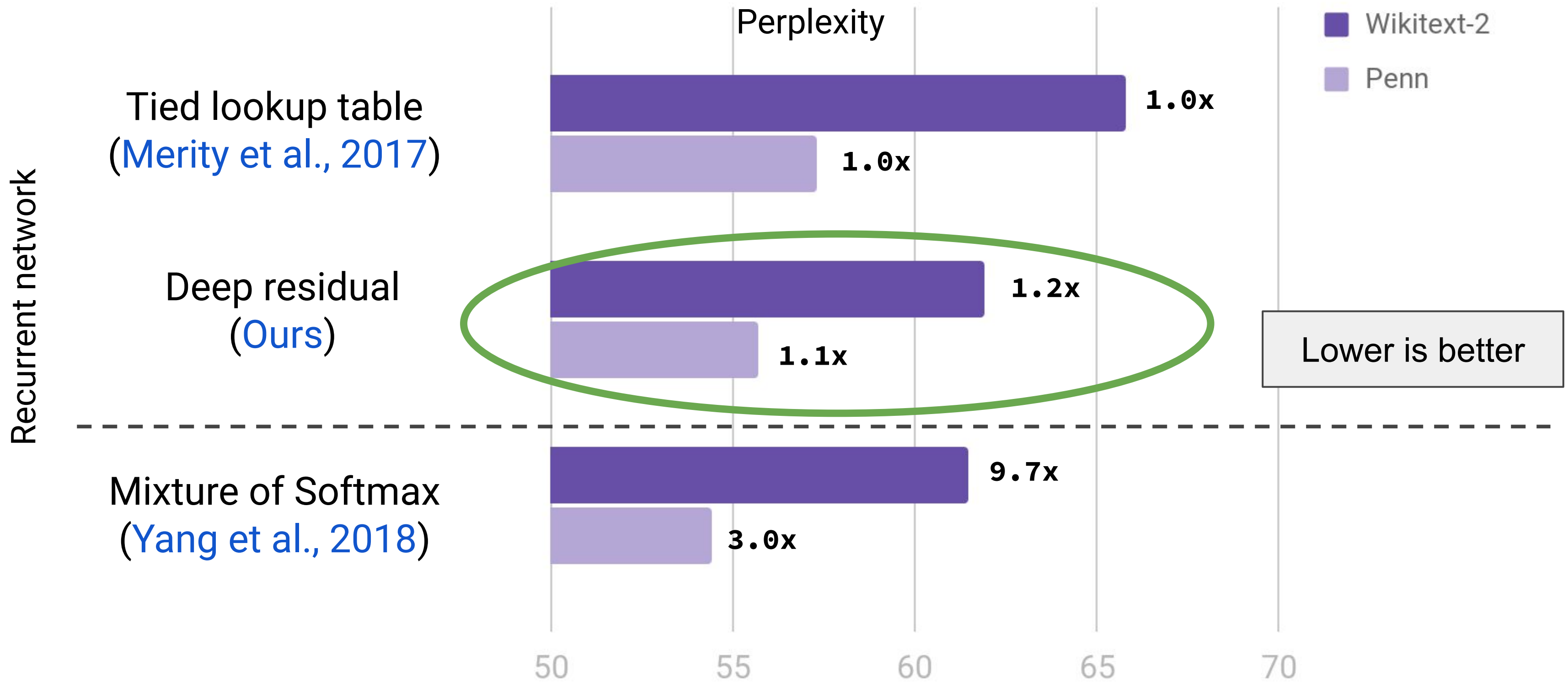


(Merity et al., 2017)

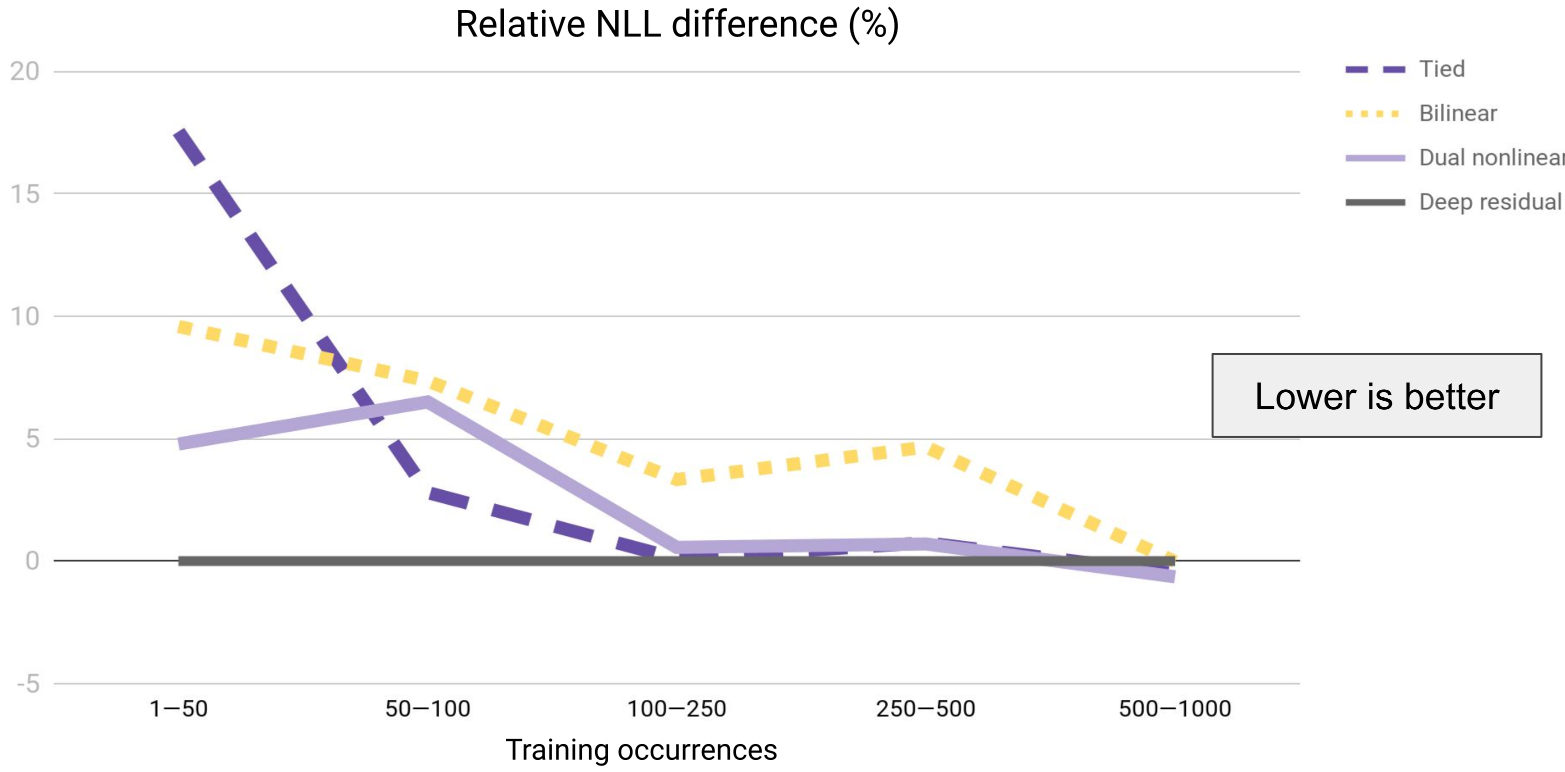
Machine translation



Language modeling



Break down by frequency

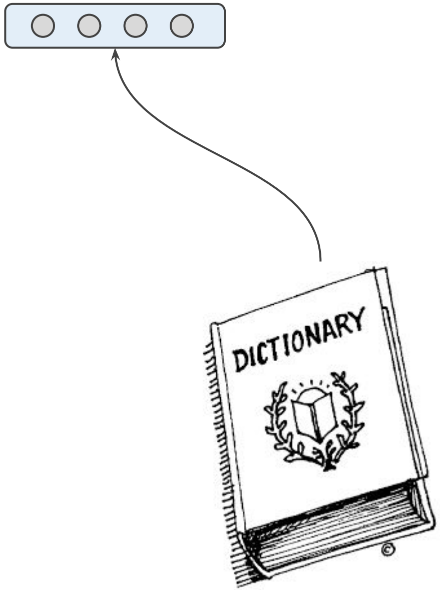


Takeaways **[ICML 2019]**

- Deep word sharing improves speed-quality tradeoff

- Improvement is due to better modeling low-frequency words

Can we further gain by grounding to dictionaries and relaxing the vocabulary assumptions?



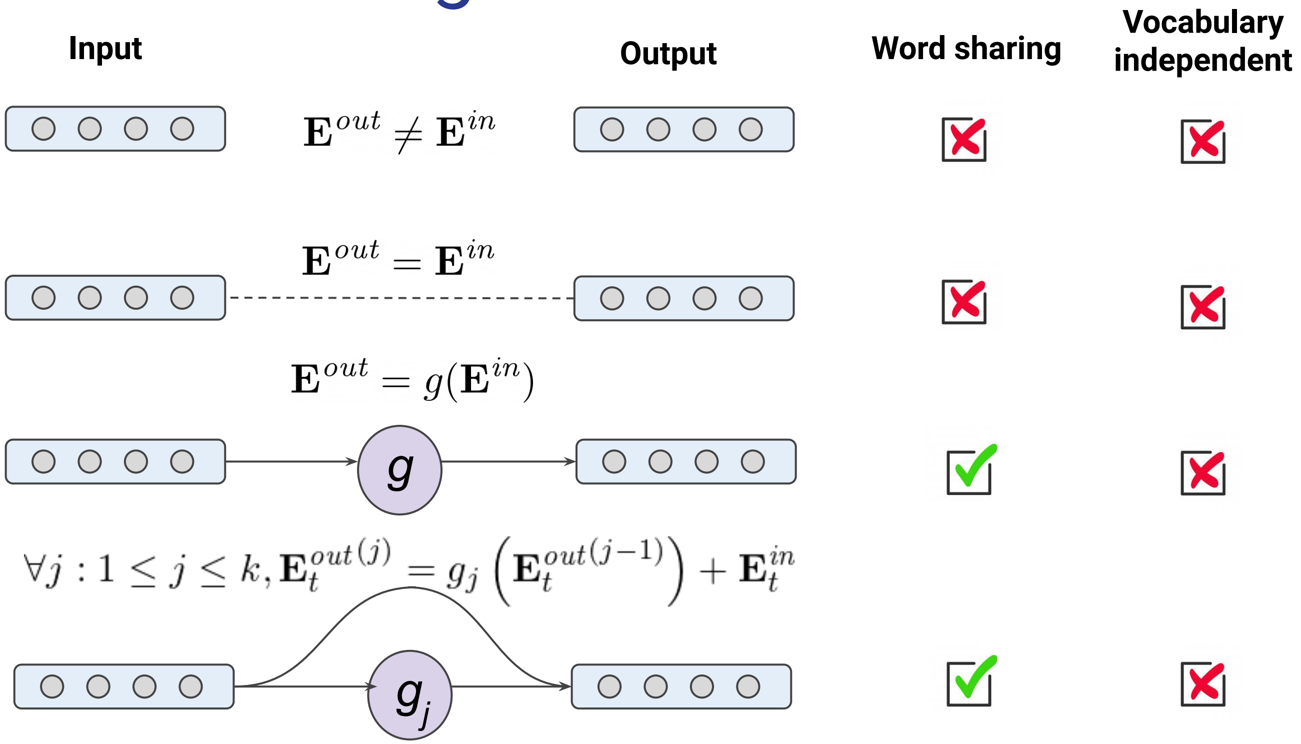
Handling rare or new words

- Character-level models (Cherry et al., 2018; Al-Rfou et al., 2019)
  - ☒ Costly prefix encoders and training
- Data-driven vocabulary selection (Sennrich et al., 2016; Radford et al., 2018)
  - ☒ Linguistically simplistic
  - ☒ Rely on lookup tables
- Local neural cache (Graves et al., 2017a,b)
  - ☑ Low-cost adaptation to rare/new words

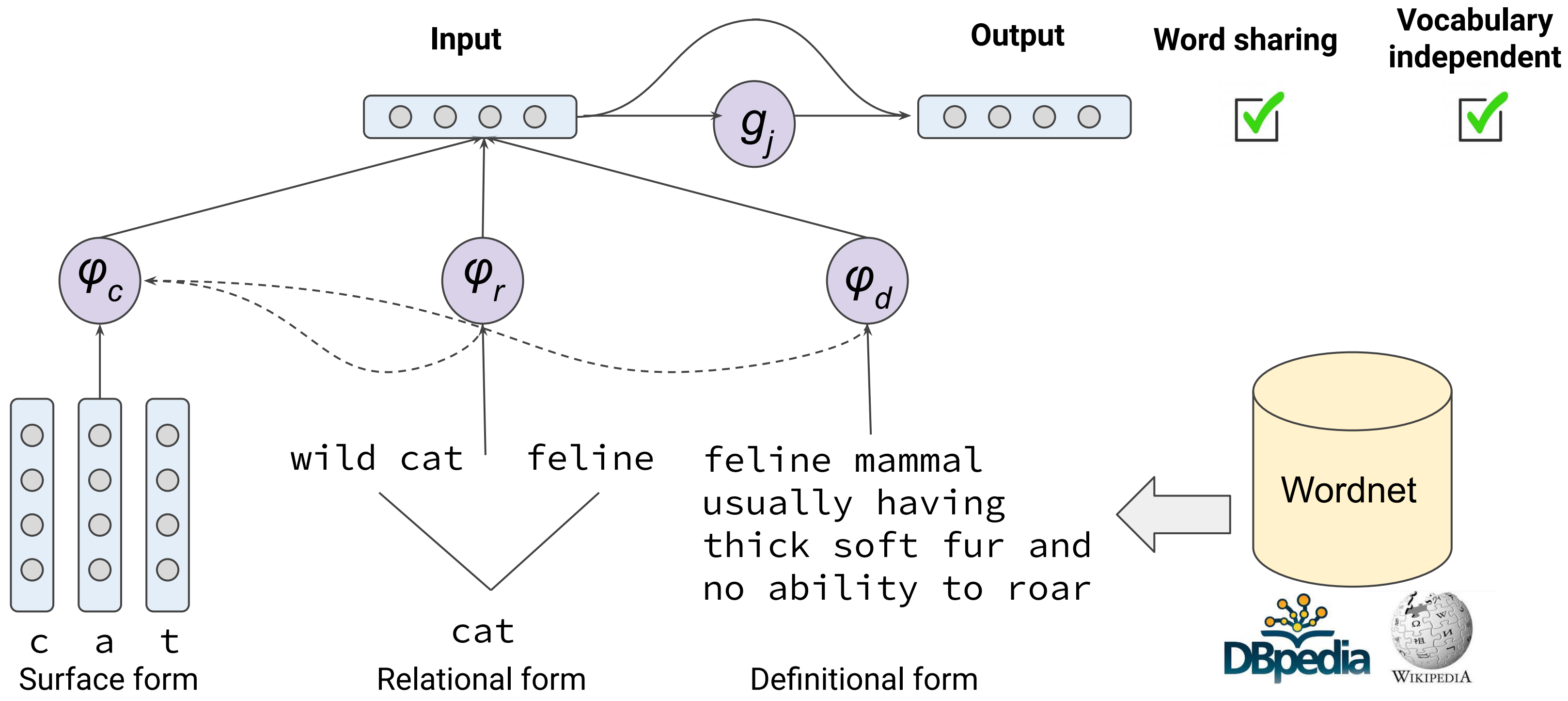
Related work: Word sharing

Lookup tables
(Zaremba et al., 2014;
Press & Wolf,2017)

Compositional /
Functional forms
(Jozefowicz et al., 2016;
Baevski & Auli, 2019;
Pappas & Henderson, 2019)



Grounded compositional outputs [EMNLP 2020]



Nikolaos Pappas, Phoebe Mulcaire, Noah A. Smith, Grounded Compositional Outputs for Adaptive Language Modeling, EMNLP 2020.

Adapting to any vocabulary [EMNLP 2020]

- We first represent the vocabulary with GroC

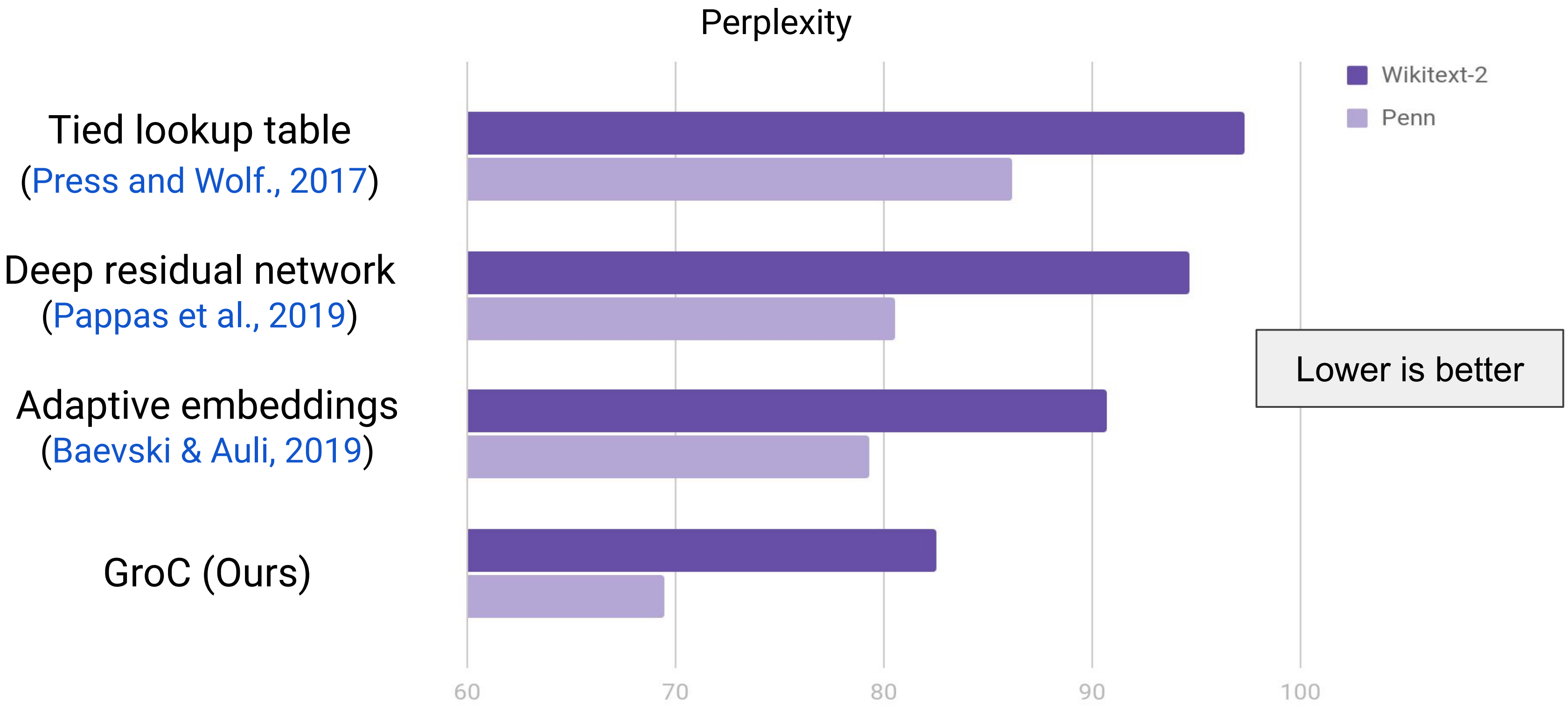
$$\mathbf{E}^{out} = \mathtt{GroC}(\mathcal{V}^*)$$

- Then we estimate the bias for each word *u*

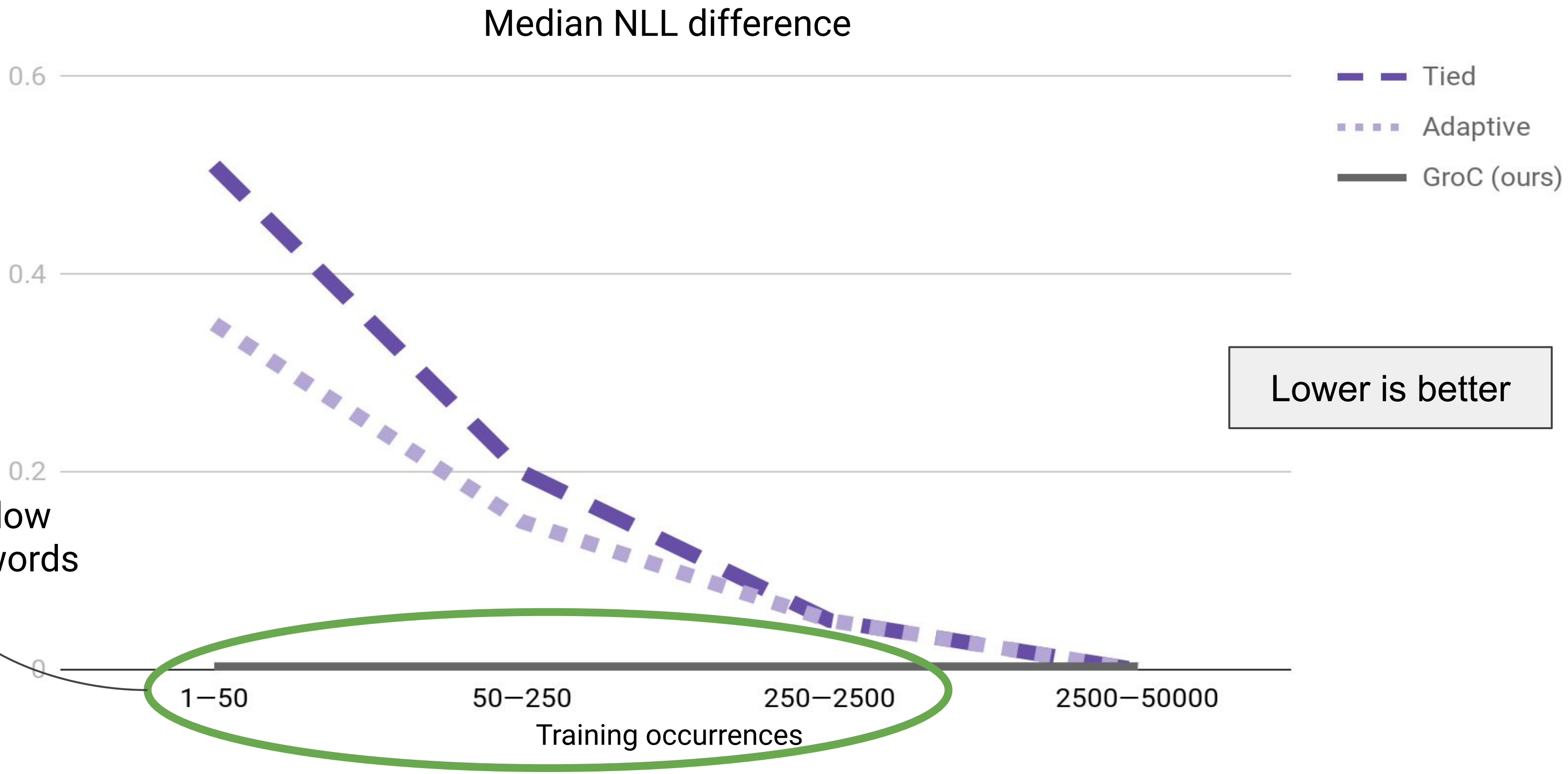
$$b_v = \sigma\left(\mathbf{w} \cdot \mathbf{e}_v^{out} + a\right)$$

$$\boxed{p(X_t = x_t \mid \mathbf{h}_{t-1}) \propto \exp\left(\mathbf{E}^{out}\mathbf{h}_{t-1} + \mathbf{b}\right)}$$

Conventional language modeling



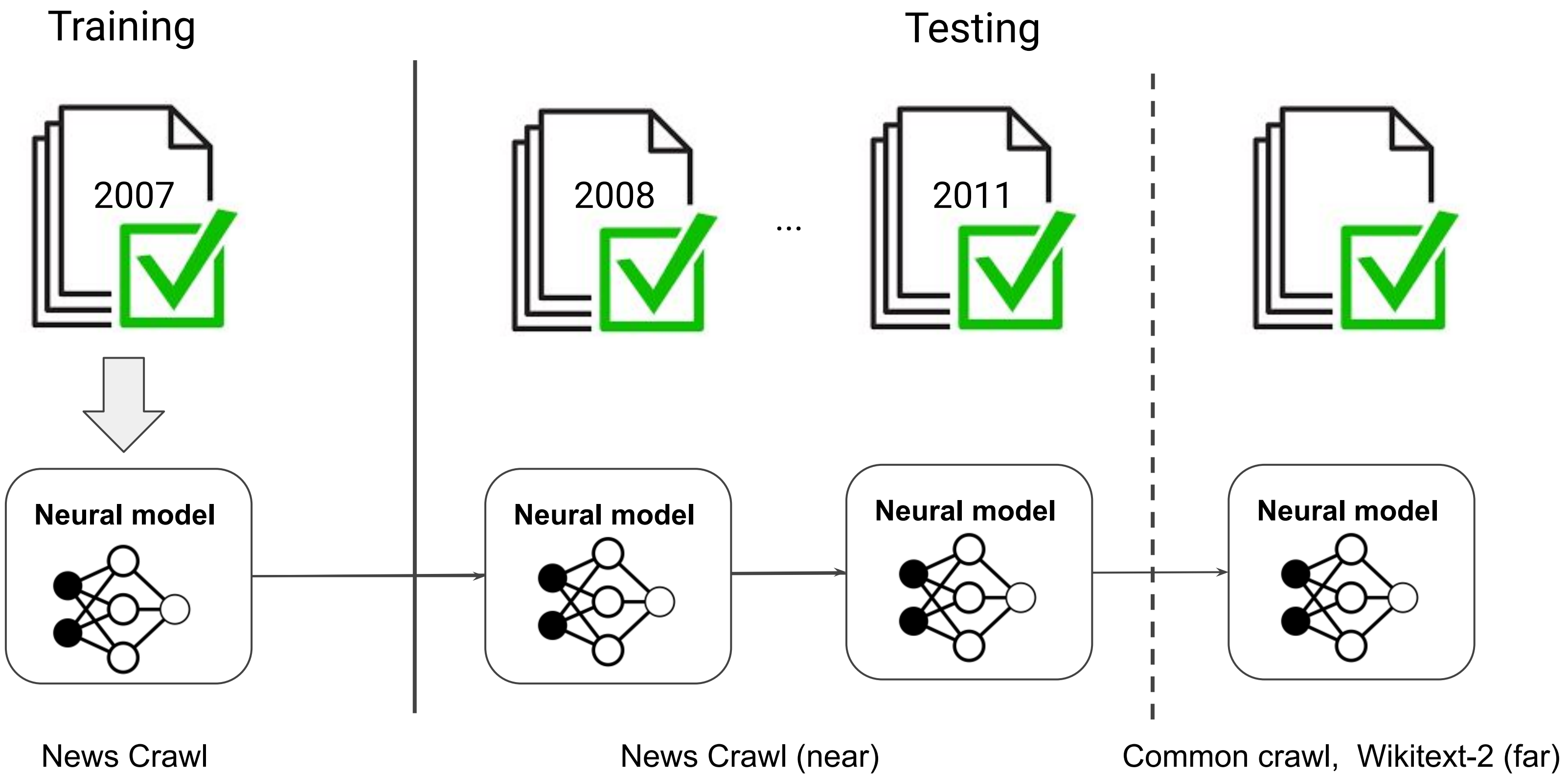
Break down by frequency



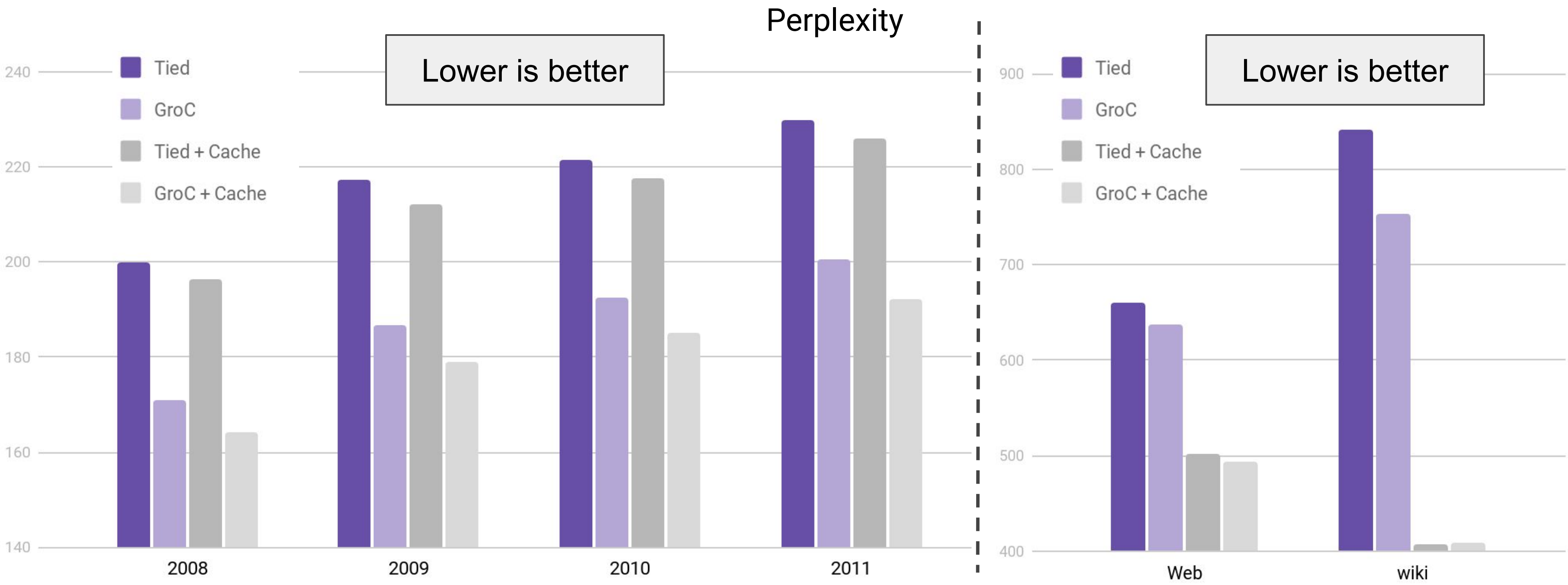
Lower is better

Better on low frequency words

Cross-domain modeling: Zero resources



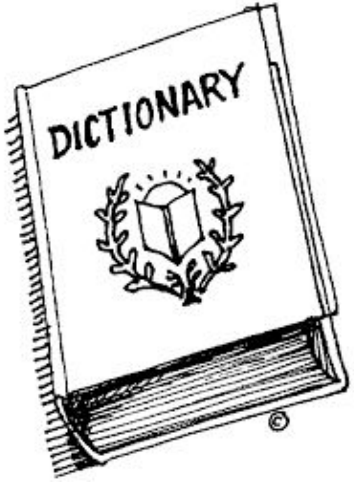
Cross-domain modeling: Zero resources



Takeaways **[EMNLP 2020]**

Grounded compositional word sharing

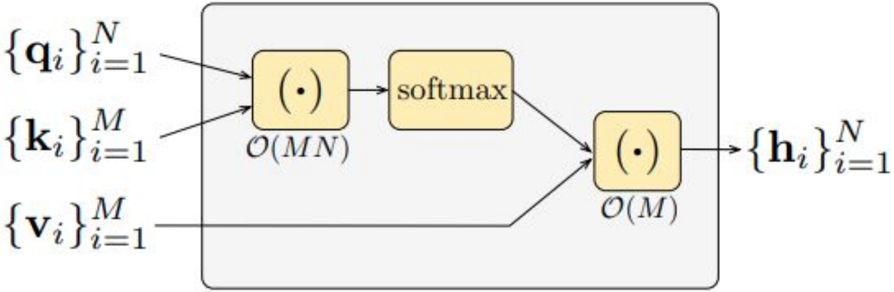
- Creates a compact representation of any vocabulary
- Achieves low perplexity on rare or new words
- Generalizes well to previously unseen domains



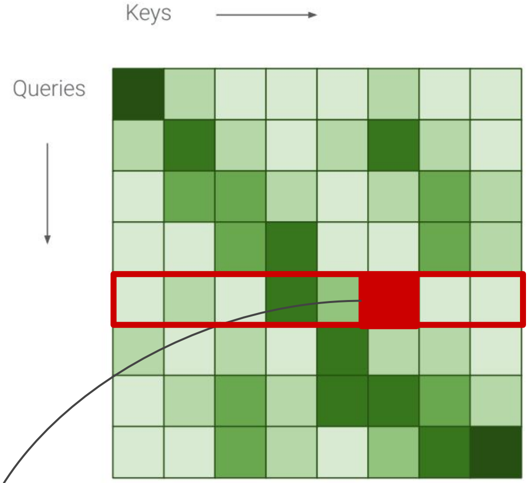
Overview

0. Introduction / Motivation
1. Broadening translation context
2. Promoting data efficiency
3. Reducing computational cost
4. Future work

Softmax attention

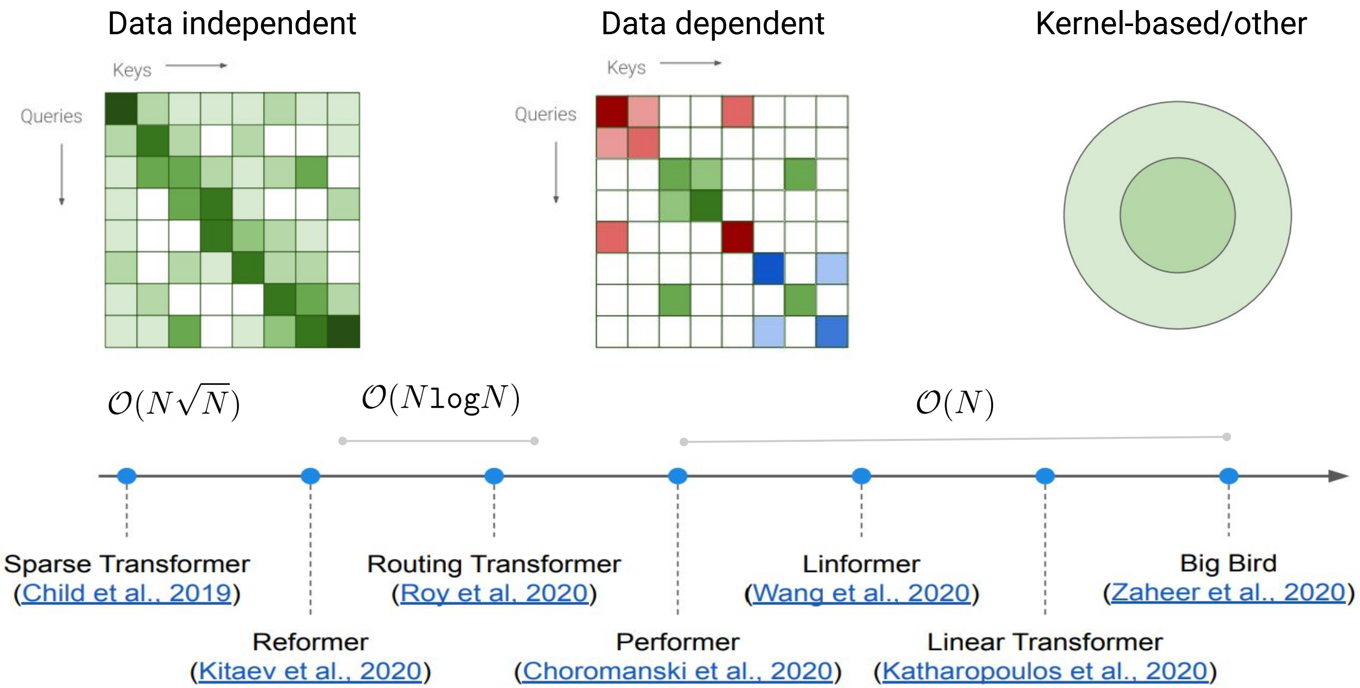


(a) Softmax attention.



$$\text{attn}\left(\mathbf{q}_t, \{\mathbf{k}_i\}, \{\mathbf{v}_i\}\right) = \sum_i \boxed{\frac{\exp\left(\mathbf{q}_t \cdot \mathbf{k}_i / \tau\right)}{\sum_j \exp\left(\mathbf{q}_t \cdot \mathbf{k}_j / \tau\right)}} \mathbf{v}_i^\top$$

Recent progress



* Ilharco et al., High-performance NLP tutorial, EMNLP 2020

Results so far

Model / Paper	Complexity	Decode
Memory Compressed† (Liu et al., 2018)	$\mathcal{O}(n_c^2)$	✓
Image Transformer† (Parmar et al., 2018)	$\mathcal{O}(n.m)$	✓
Set Transformer† (Lee et al., 2019)	$\mathcal{O}(nk)$	✗
Transformer-XL† (Dai et al., 2019)	$\mathcal{O}(n^2)$	✓
Sparse Transformer (Child et al., 2019)	$\mathcal{O}(n\sqrt{n})$	✓
Reformer† (Kitaev et al., 2020)	$\mathcal{O}(n \log n)$	✓
Routing Transformer (Roy et al., 2020)	$\mathcal{O}(n \log n)$	✓
Axial Transformer (Ho et al., 2019)	$\mathcal{O}(n\sqrt{n})$	✓
Compressive Transformer† (Rae et al., 2020)	$\mathcal{O}(n^2)$	✓
Sinkhorn Transformer† (Tay et al., 2020b)	$\mathcal{O}(b^2)$	✓
Longformer (Beltagy et al., 2020)	$\mathcal{O}(n(k+m))$	✓
ETC (Ainslie et al., 2020)	$\mathcal{O}(n_g^2 + nn_g)$	✗
Synthesizer (Tay et al., 2020a)	$\mathcal{O}(n^2)$	✓
Performer (Choromanski et al., 2020)	$\mathcal{O}(n)$	✓
Linformer (Wang et al., 2020b)	$\mathcal{O}(n)$	✗
Linear Transformers† (Katharopoulos et al., 2020)	$\mathcal{O}(n)$	✓
Big Bird (Zaheer et al., 2020)	$\mathcal{O}(n)$	✗

Faster inference on CIFAR10

Higher accuracy on long sequence tasks

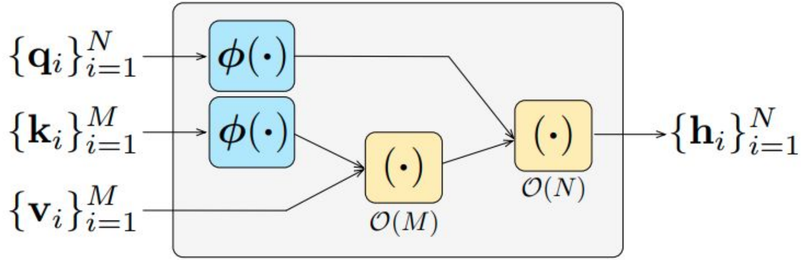
Lower perplexity on LM with longer context

- Transformers can be more memory and compute efficient
- Benefits mostly when they are trained on longer sequences
- Evaluation is often tricky

(Tay et al., 2020)

Random feature attention (in a nutshell) [ICLR subm.]

- ☑ Unbiased approximation of softmax attention
  - 2X faster on MT decoding
  - 17X faster on LM decoding
  - 5X faster on long text classification
- ☑ Realistic speed/quality estimates
  - Moderate and long sequence tasks
  - Measurements with fixed batch size
- ☑ New insights on how to improve attention



$$\textsc{Rfa}\left(\mathbf{q}_t, \{\mathbf{k}_i\}, \{\mathbf{v}_i\}\right) = \frac{\phi\left(\mathbf{q}_t\right)^\top \sum_i \phi\left(\mathbf{k}_i\right) \otimes \mathbf{v}_i}{\phi\left(\mathbf{q}_t\right) \cdot \sum_j \phi\left(\mathbf{k}_j\right)}$$

Hao Peng, **Nikolaos Pappas**, Dani Yogatama, Roy Schwartz, Noah A. Smith, Linkpeng Kong, Random feature attention, ICLR subm.

Random Fourier features

- Can approximate a desired shift-invariant kernel e.g. Gaussian or Arccos

- Let $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^{2D}$ be a nonlinear transformation and $w_i \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$.

$$\phi(\mathbf{x}) = \sqrt{1/D}\left[\sin(\mathbf{w}_1 \cdot \mathbf{x}), \ldots, \sin(\mathbf{w}_D \cdot \mathbf{x}), \cos(\mathbf{w}_1 \cdot \mathbf{x}), \ldots, \cos(\mathbf{w}_D \cdot \mathbf{x})\right]^\top$$

then it provides an unbiased approximation of Gaussian kernel

$$\boxed{\exp(-||\mathbf{x} - \mathbf{y}||^2 / 2\sigma^2) \approx \phi(\mathbf{x}) \cdot \phi(\mathbf{y})} \quad (1)$$

(Rahimi & Recht, 2008)

Random feature attention [ICLR subm.]

- Exponential becomes Gaussian if x, y are normalized (Rawat et al., 2019)

$$\texttt{exp}(x \cdot y/\sigma^2) = \texttt{exp}(1/\sigma^2)\texttt{exp}(-||x-y||^2/2\sigma^2)$$

- Therefore RFA is derived as follows

$$\begin{aligned}
\operatorname{attn}\left(\mathbf{q}_t, \{\mathbf{k}_i\}, \{\mathbf{v}_i\}\right) &= \sum_i \frac{\exp\left(\mathbf{q}_t \cdot \mathbf{k}_i/\sigma^2\right)}{\sum_j \exp\left(\mathbf{q}_t \cdot \mathbf{k}_j/\sigma^2\right)} \mathbf{v}_i^\top \\
&\overset{(1)}{\approx} \sum_i \frac{\boldsymbol{\phi}\left(\mathbf{q}_t\right)^\top \boldsymbol{\phi}\left(\mathbf{k}_i\right) \mathbf{v}_i^\top}{\sum_j \boldsymbol{\phi}\left(\mathbf{q}_t\right) \cdot \boldsymbol{\phi}\left(\mathbf{k}_j\right)} \\
&= \frac{\boldsymbol{\phi}\left(\mathbf{q}_t\right)^\top \sum_i \boldsymbol{\phi}\left(\mathbf{k}_i\right) \otimes \mathbf{v}_i}{\boldsymbol{\phi}\left(\mathbf{q}_t\right) \cdot \sum_j \boldsymbol{\phi}\left(\mathbf{k}_j\right)} = \operatorname{RFA}\left(\mathbf{q}_t, \{\mathbf{k}_i\}, \{\mathbf{v}_i\}\right)
\end{aligned}$$

Random feature attention [ICLR subm.]

- Exponential becomes Gaussian if x, y are normalized (Rawat et al., 2019)

$$\exp(x \cdot y / \sigma^2) = \exp(1/\sigma^2)\exp(-||x-y||^2/2\sigma^2)$$

temperature

- Therefore RFA is derived as follows

$$\begin{aligned}
\operatorname{attn}\left(\mathbf{q}_t, \{\mathbf{k}_i\}, \{\mathbf{v}_i\}\right) &= \sum_i \frac{\exp\left(\mathbf{q}_t \cdot \mathbf{k}_i / \sigma^2\right)}{\sum_j \exp\left(\mathbf{q}_t \cdot \mathbf{k}_j / \sigma^2\right)} \mathbf{v}_i^\top \\
&\overset{(1)}{\approx} \sum_i \frac{\boldsymbol{\phi}\left(\mathbf{q}_t\right)^\top \boldsymbol{\phi}\left(\mathbf{k}_i\right) \mathbf{v}_i^\top}{\sum_j \boldsymbol{\phi}\left(\mathbf{q}_t\right) \cdot \boldsymbol{\phi}\left(\mathbf{k}_j\right)} \\
&= \frac{\boldsymbol{\phi}\left(\mathbf{q}_t\right)^\top \sum_i \boldsymbol{\phi}\left(\mathbf{k}_i\right) \otimes \mathbf{v}_i}{\boldsymbol{\phi}\left(\mathbf{q}_t\right) \cdot \sum_j \boldsymbol{\phi}\left(\mathbf{k}_j\right)} = \operatorname{RFA}\left(\mathbf{q}_t, \{\mathbf{k}_i\}, \{\mathbf{v}_i\}\right)
\end{aligned}$$

Reparameterization trick

$$\widetilde{\mathbf{w}}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d),$$
$$\mathbf{w}_i = \boldsymbol{\sigma} \circ \widetilde{\mathbf{w}}_i.$$

(Kingma & Welling, 2014)

RFA variants [ICLR subm.]

- **Non-causal**: we compute S, z only once for the whole sequence

$$\frac{\phi(\mathbf{q}_t)^\top \overbrace{\sum_i \phi(\mathbf{k}_i) \otimes \mathbf{v}_i}^{\mathbf{S}}}{\phi(\mathbf{q}_t) \cdot \underbrace{\sum_j \phi(\mathbf{k}_j)}_{\mathbf{z}}}$$

Encoder

$\mathcal{O}(M)$

Cross

$\mathcal{O}(M+N)$

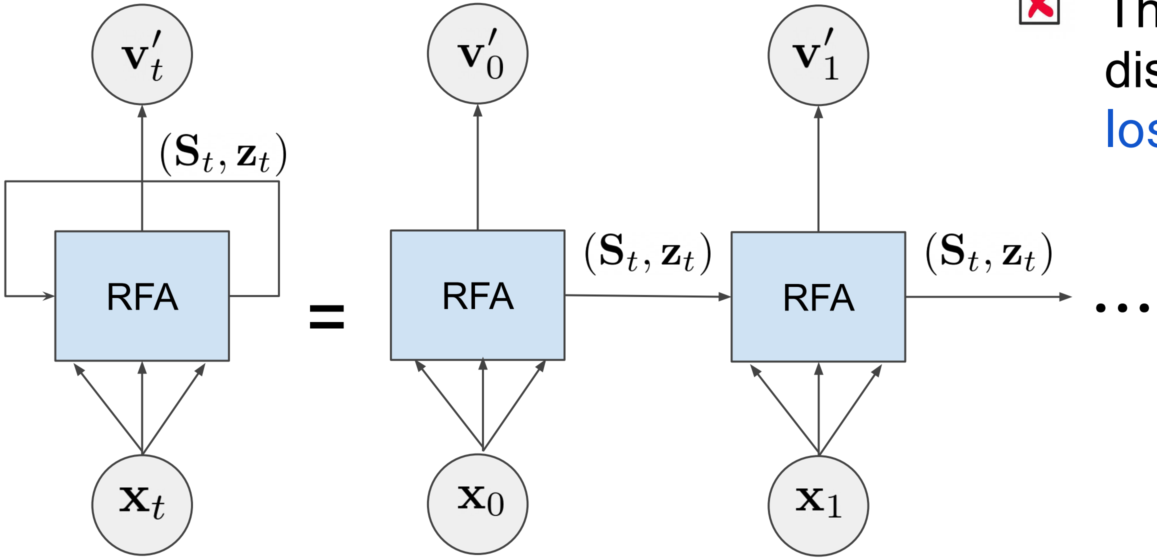
- **Causal**: we compute S_t and z_t iteratively at each step

$$\frac{\phi(\mathbf{q}_t)^\top \overbrace{\sum_{i \leq t} \phi(\mathbf{k}_i) \otimes \mathbf{v}_i}^{\mathbf{S}_t}}{\phi(\mathbf{q}_t) \cdot \underbrace{\sum_{j \leq t} \phi(\mathbf{k}_j)}_{\mathbf{z}_t}}$$

Decoder

$\mathcal{O}(N)$

Recurrent formulation [ICLR subm.]

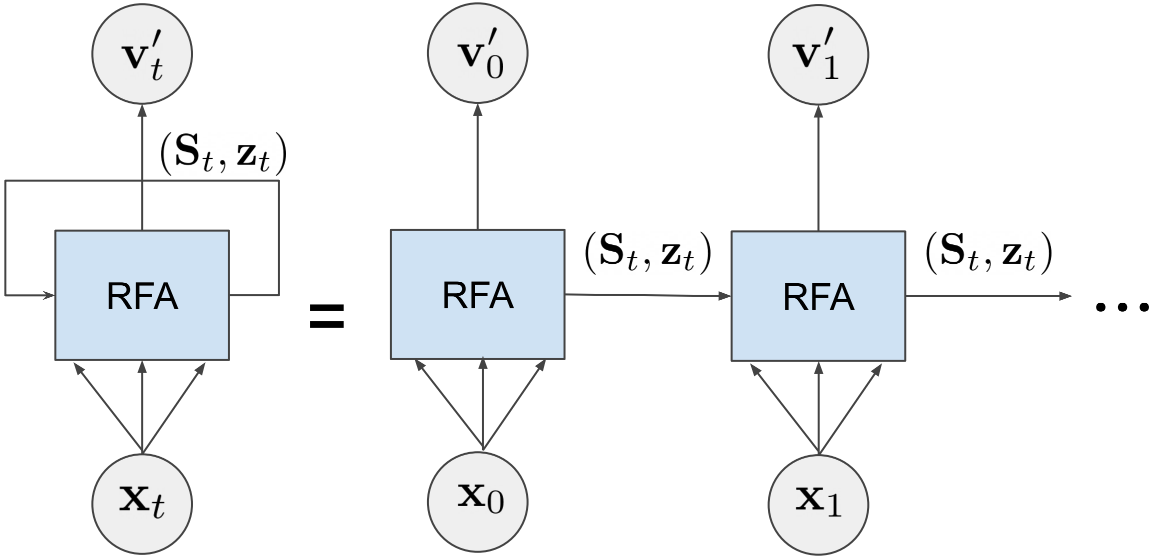


- There is no explicit modeling of distance or locality (Katharopoulos et al., 2020)

$$\mathbf{S}_t = \mathbf{S}_{t-1} + \phi(\mathbf{k}_t) \otimes \mathbf{v}_t$$

$$\mathbf{z}_t = \mathbf{z}_{t-1} + \phi(\mathbf{k}_t)$$

Gated-RFA: Learning with recency bias [ICLR subm.]



$$
\begin{aligned}
g_t &= \operatorname{sigmoid}(\mathbf{w}_g \cdot \mathbf{x}_t + b_g), \\
\mathbf{S}_t &= g_t \, \mathbf{S}_{t-1} + (1 - g_t) \, \boldsymbol{\phi}(\mathbf{k}_t) \otimes \mathbf{v}_t, \\
\mathbf{z}_t &= g_t \, \mathbf{z}_{t-1} + (1 - g_t) \, \boldsymbol{\phi}(\mathbf{k}_t).
\end{aligned}
$$

Machine translation

Model	WMT14		IWSLT14	
	EN-DE	EN-FR	DE-EN	Speed
BASE	28.1	39.0	34.6	1.0×
ϕ_{elu} (Katharopoulos et al., 2020)	21.3	34.0	29.9	2.0×
RFA-Gaussian	28.0	39.2	34.5	1.8×
RFA-arccos	28.1	38.9	34.4	1.9×
RFA-GATE-Gaussian	28.1	39.0	34.6	1.8×
RFA-GATE-arccos	28.2	39.2	34.4	1.9×

Higher is better

BLEU scores on MT.

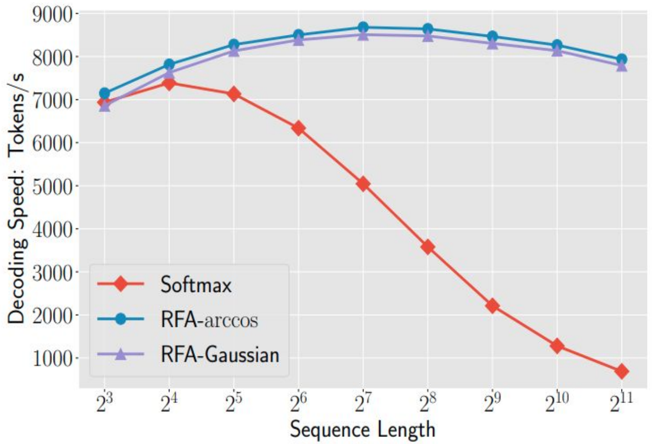
- Double the speed for short sequences with similar quality (2X speedup)
- Superiority to linear transformer shows the importance of feature map

Language modeling

Model	Small Dev.	Small Test	Big Dev.	Big Test
BASE	33.0	34.5	24.5	26.2
ϕ_{elu} (Katharopoulos et al., 2020)	38.4	40.1	28.7	30.2
RFA-Gaussian	33.6	35.7	25.8	27.5
RFA-arccos	36.0	37.7	26.4	28.1
RFA-GATE-Gaussian	**31.3**	**32.7**	**23.2**	**25.0**
RFA-GATE-arccos	**32.8**	**34.0**	24.8	26.3
RFA-GATE-Gaussian-Stateful	**29.4**	**30.5**	**22.0**	**23.5**

Lower is better

Perplexity on Wikitext-103.

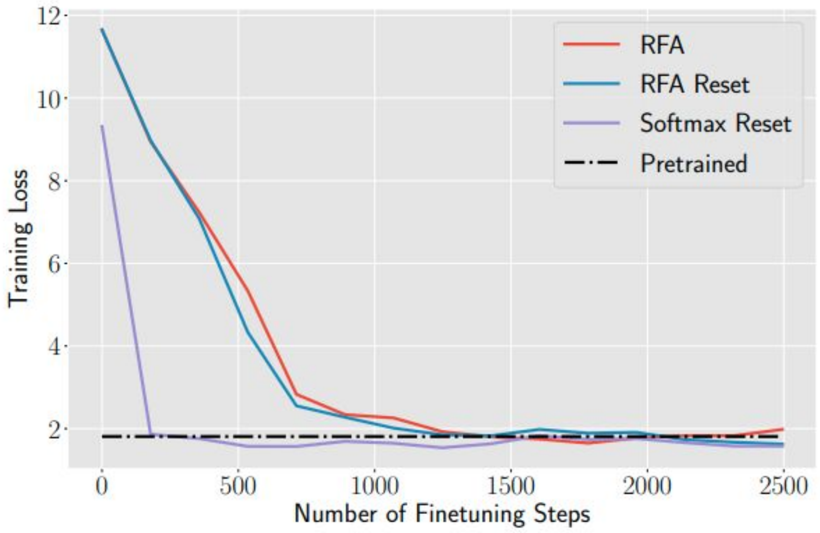


Higher is better

Decoding speed

- RFA-gate is better than baseline with up to 17X decoding speed
- Competitive speed-quality tradeoff in the long range arena (5X speedup)

“Streamlining” pretrained models

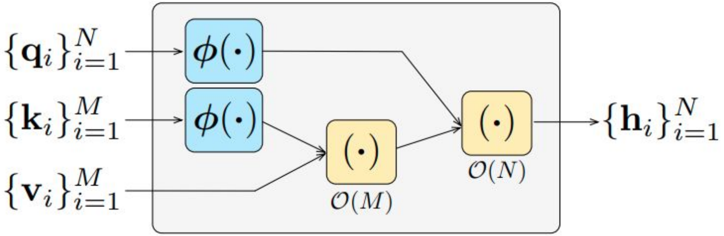


Finetuning RFA from a pretrained softmax model.

- Pretrained softmax parameters are as useful as random ones
- RFA can recover the pretraining loss with a few iterations
- Potential to reduce finetuning cost for large models (GPT3)

Takeaways **[ICLR subm.]**

- General component with linear complexity for attending sequences
- Competitive trade-offs on both long and moderate length sequences
- New scalable attention variant that learns with recency bias



Random feature attention.

Acknowledgments

L. Miculicich

D. Ram

J. Henderson

A. Popescu-Belis

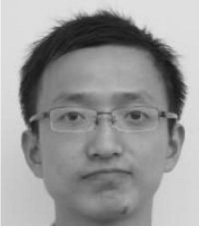
N. A. Smith

H. Peng

P. Mulcaire

D. Yogatama

R. Schwartz



L. Kong

UWNLP

UNIVERSITY of WASHINGTON

FNSNF

Horizon 2020
European Union Funding
for Research & Innovation

Thank you!