Moving towards end-to-end systems

– Attention Mechanism & Pretrained Technic

Shared by：庞亮，熊睿彬，郝长盈

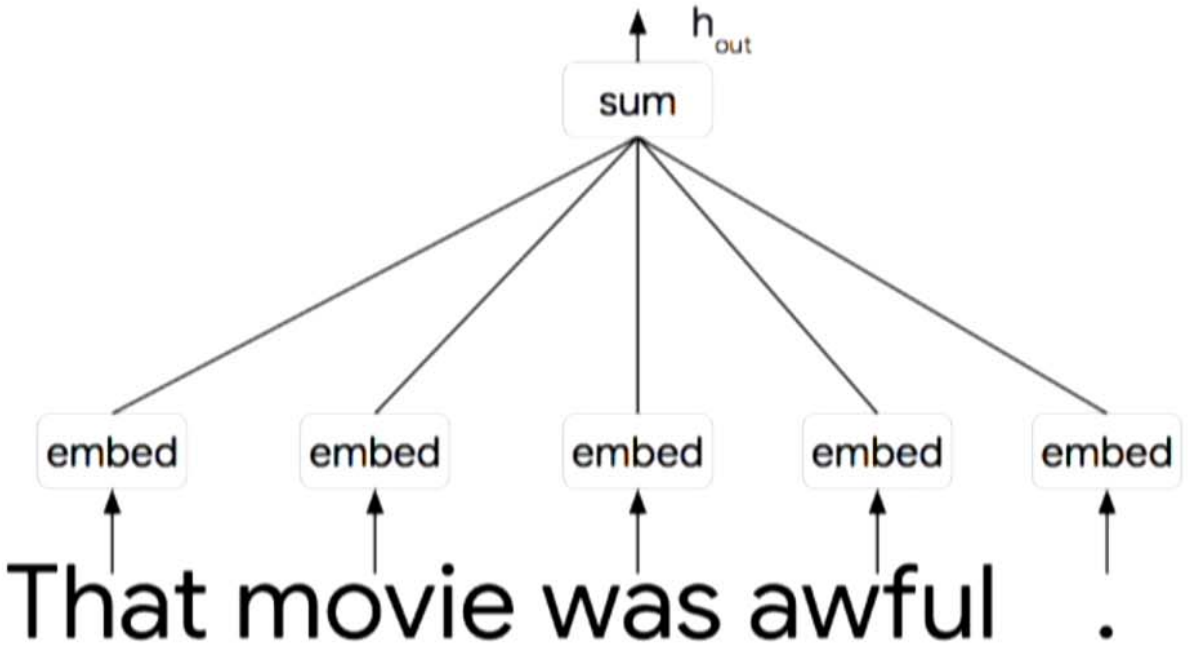
Roadmap

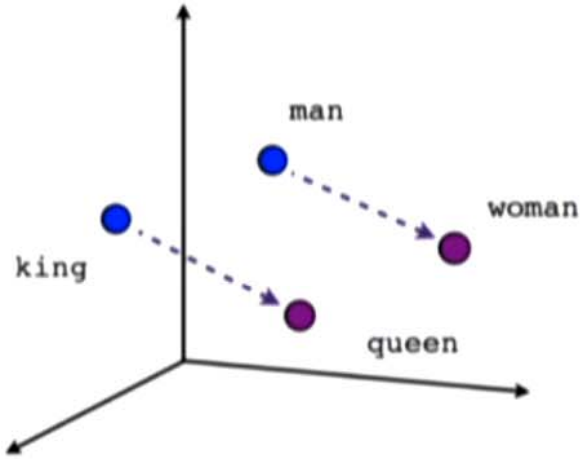
- **Language model**
- **Attention**
- **Elmo**
- **Bert**
- **QANet**

Roadmap

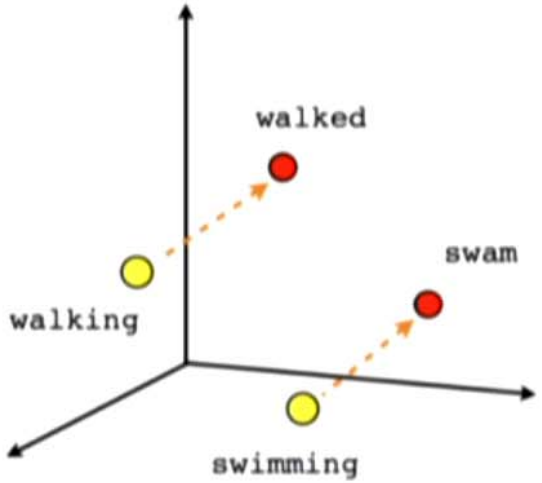
- **Language model**
- Attention
- Elmo
- Bert
- QANet

Bag of words

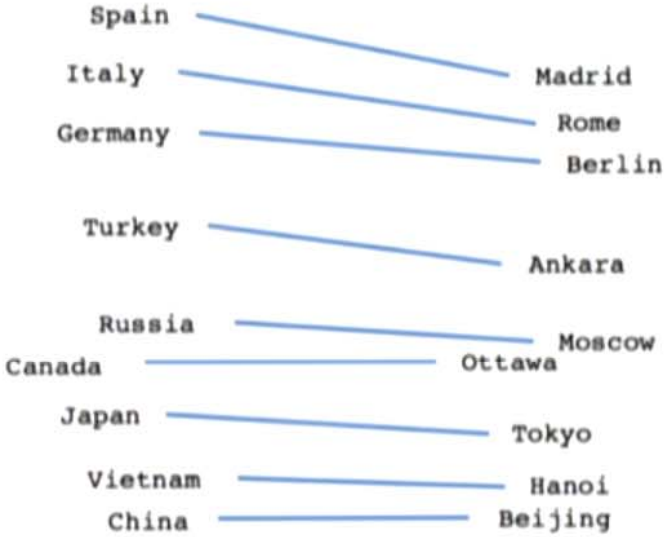




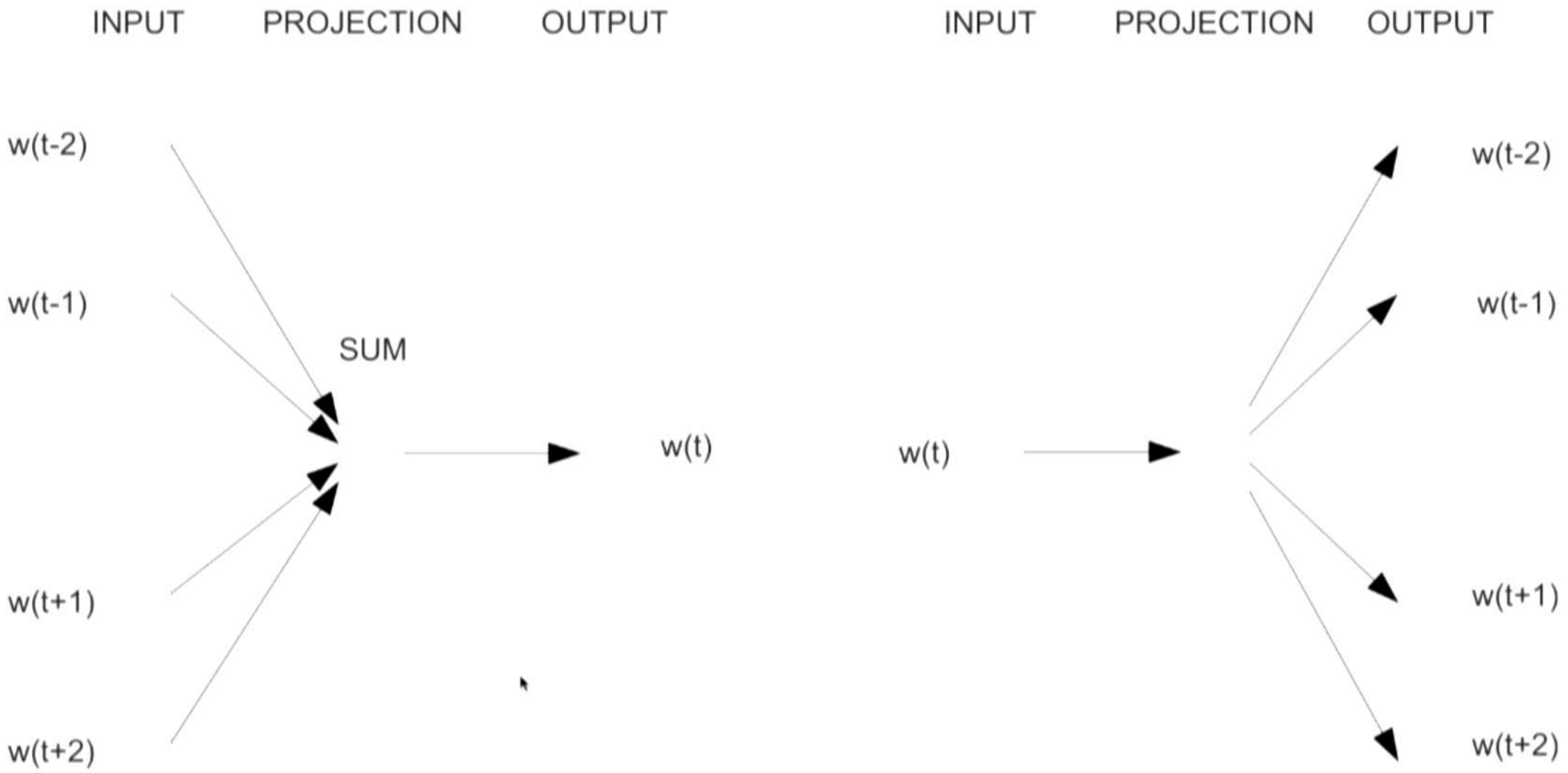
Male-Female



Verb tense



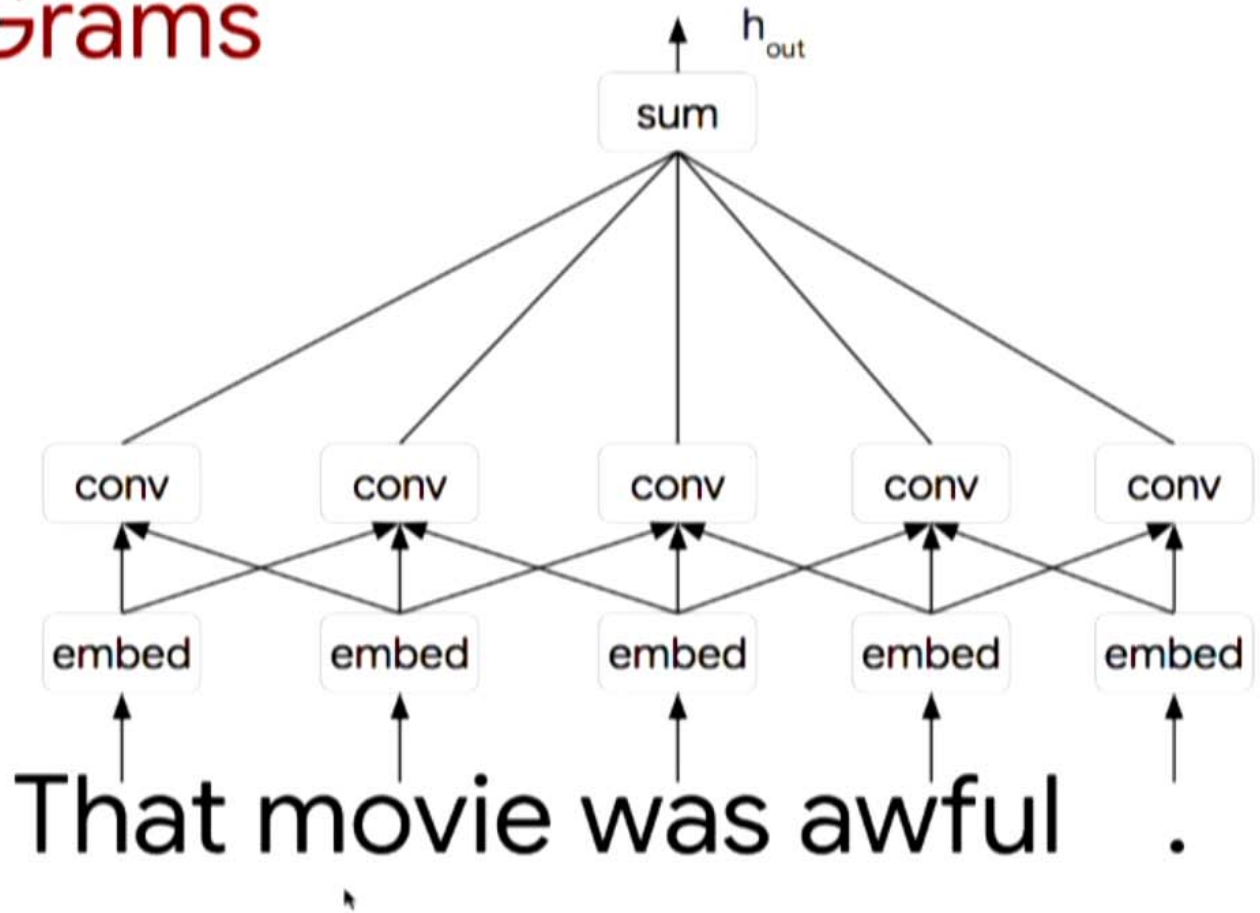
Country-Capital



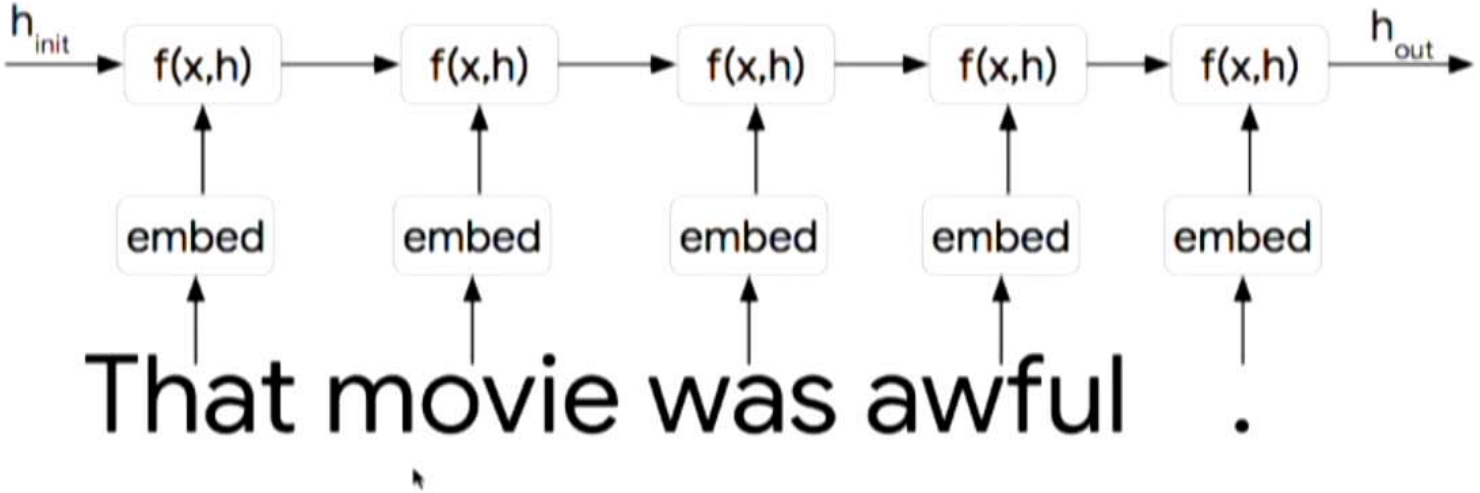
Continuous bag-of-words and skip-gram architectures (Mikolov et al., 2013a; 2013b)

CBOW **Skip-gram**

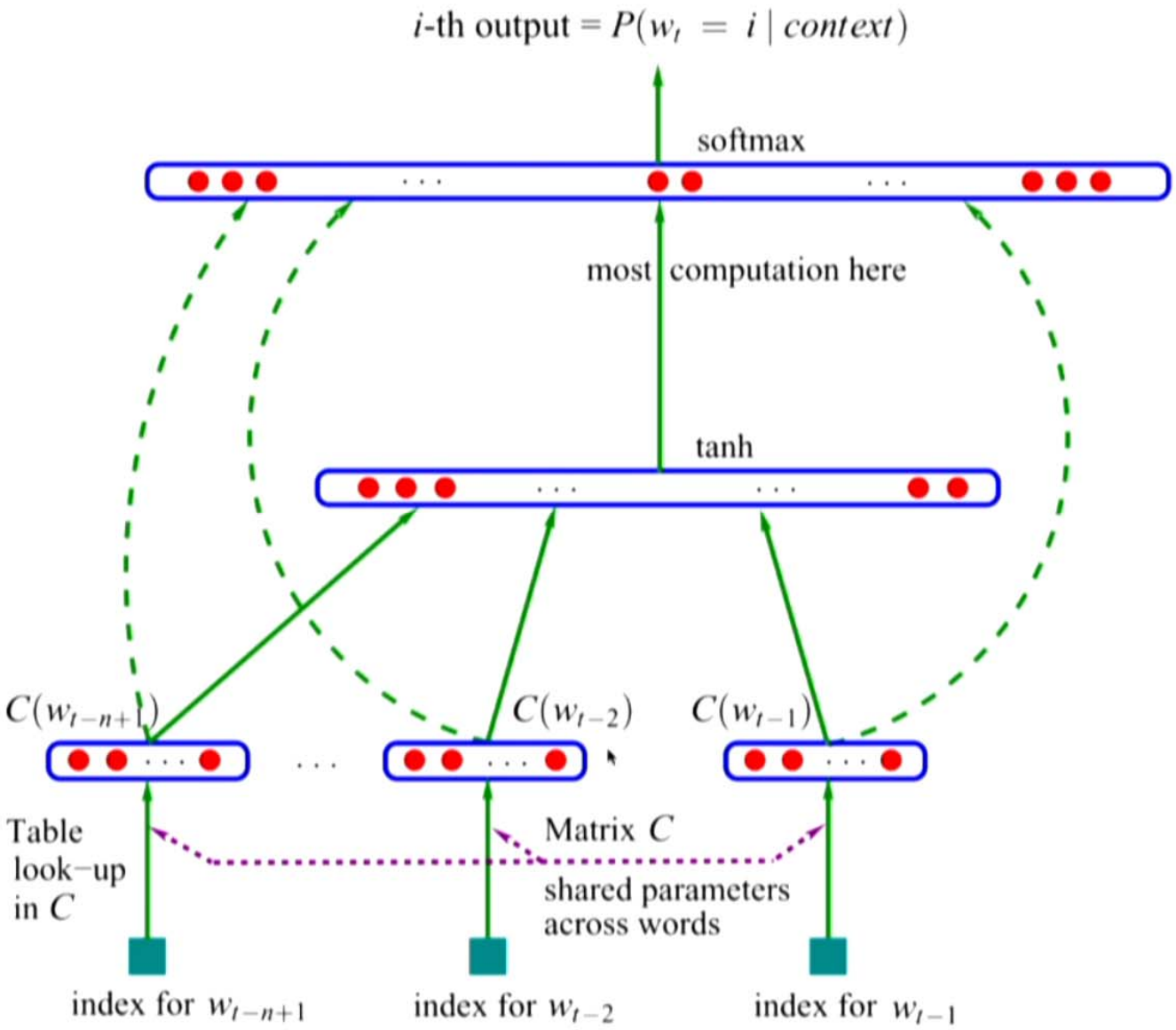
Bag of N-Grams



Recurrent Neural Networks

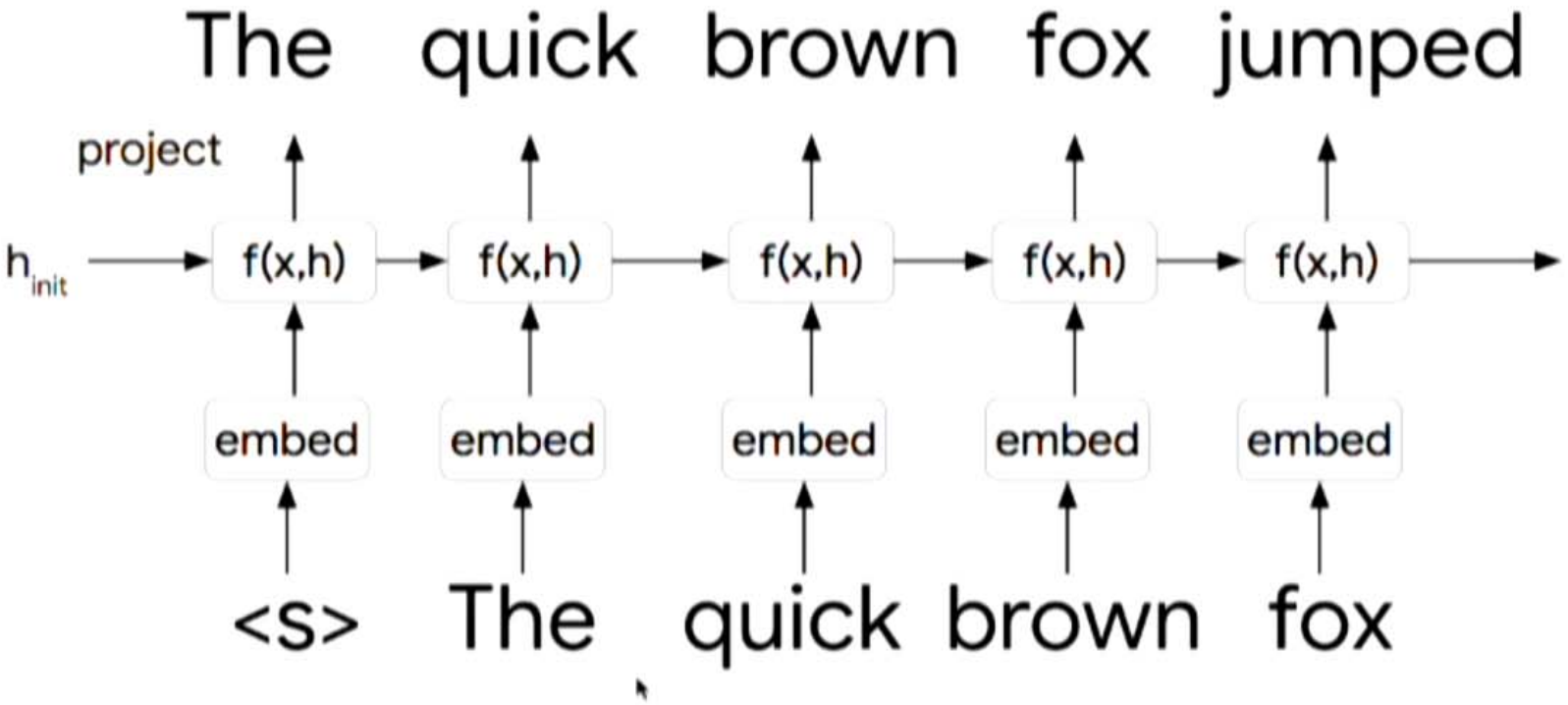


The quick brown fox jumped over the lazy dog



A feed-forward neural network language model (Bengio et al., 2001; 2003)

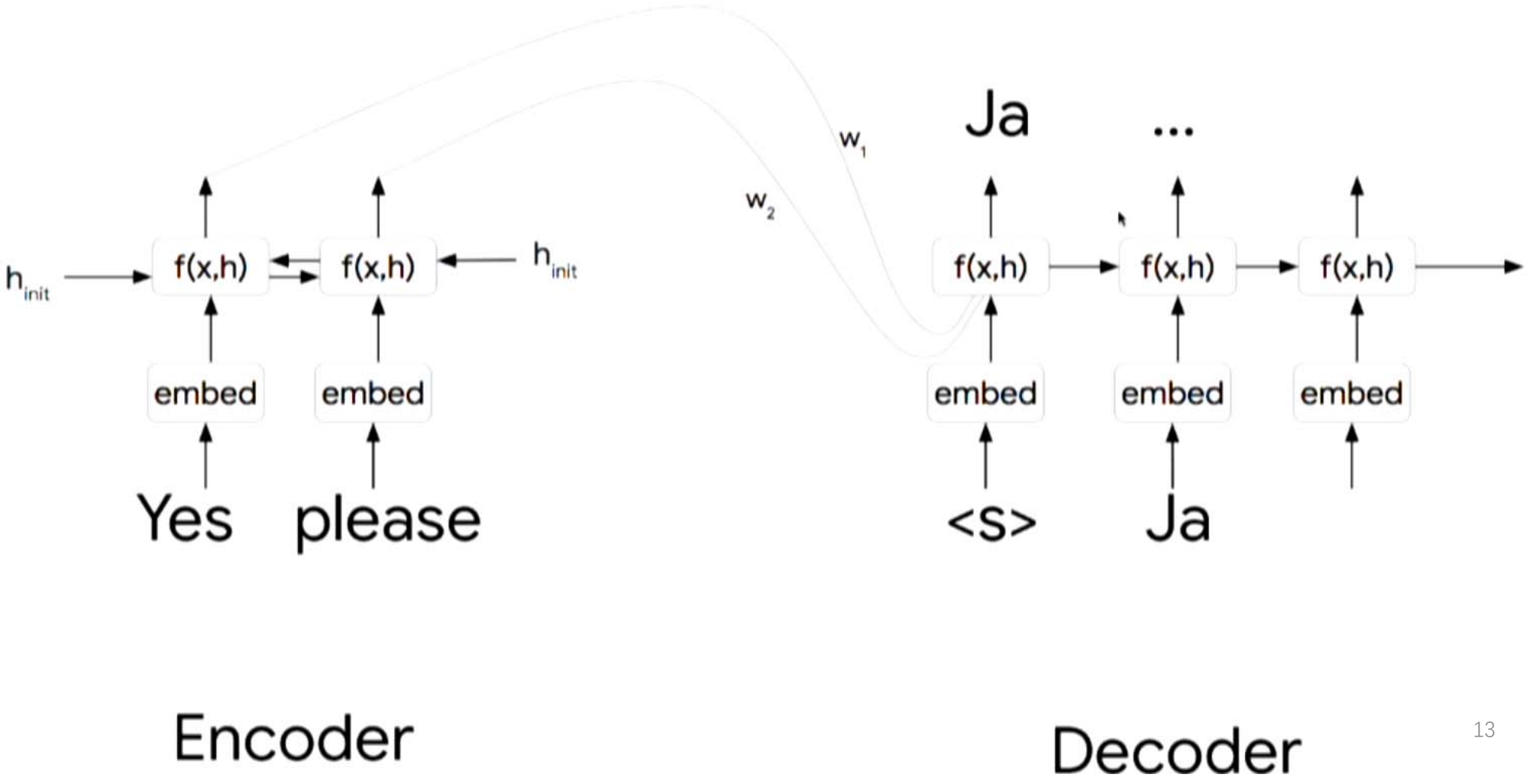
Language Models

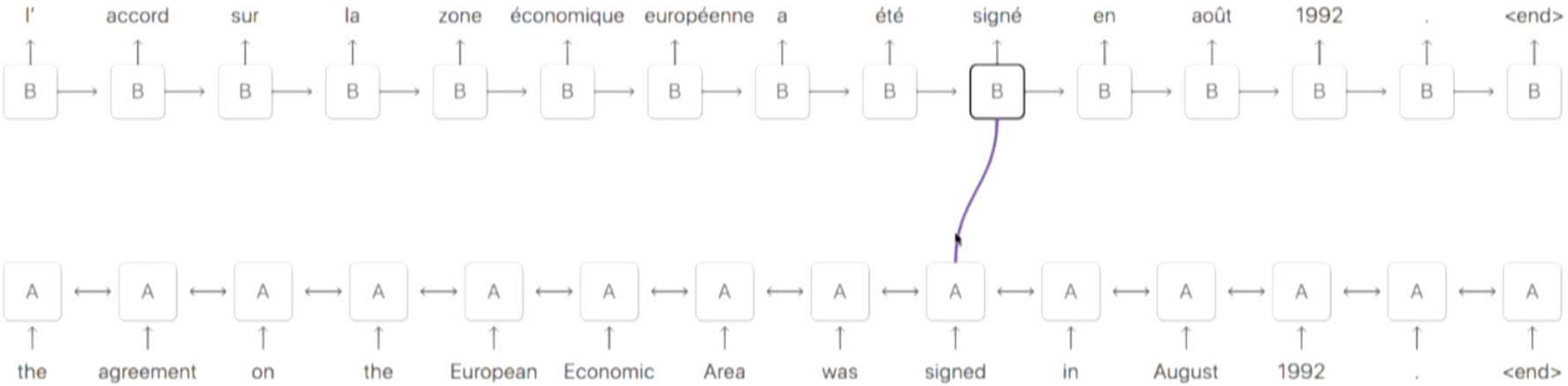


Roadmap

- Language model
- **Attention**
- Elmo
- Bert
- QA net

Seq2Seq + Attention

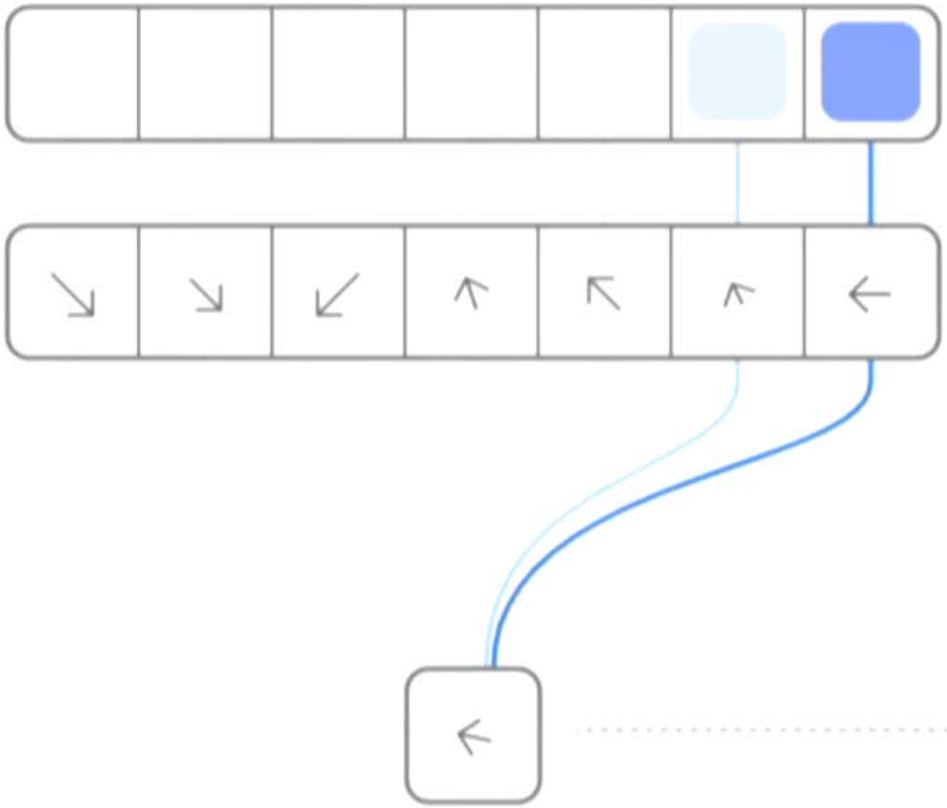




A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

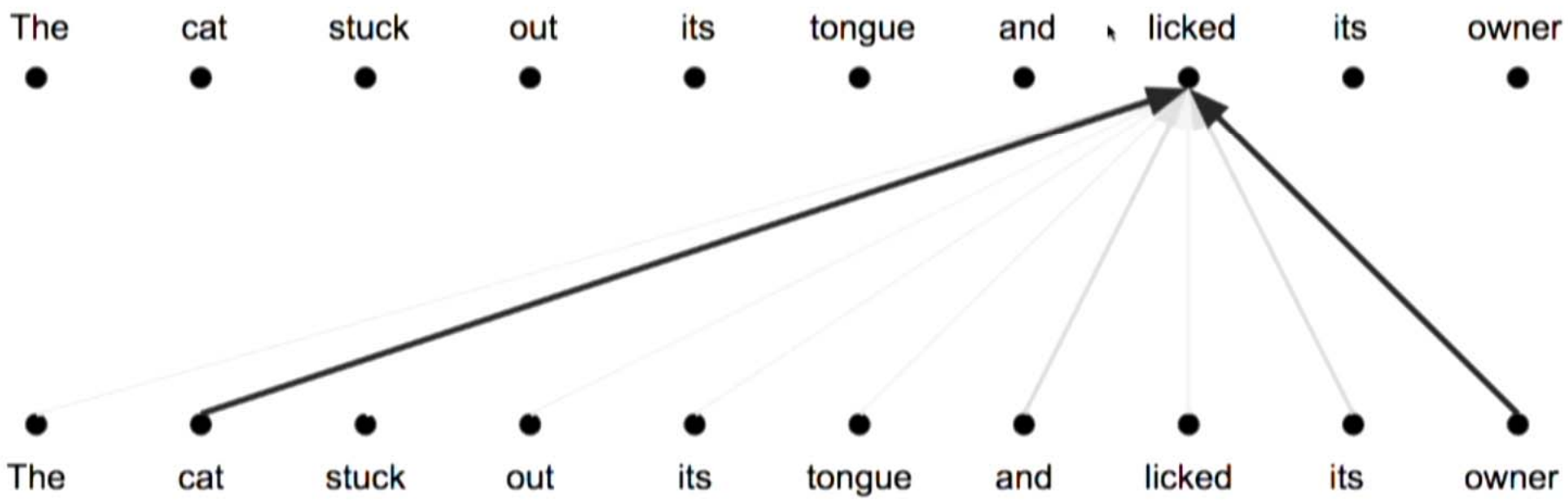


The RNN gives an attention distribution which describe how we spread out the amount we care about different memory positions.

The read result is a weighted sum.

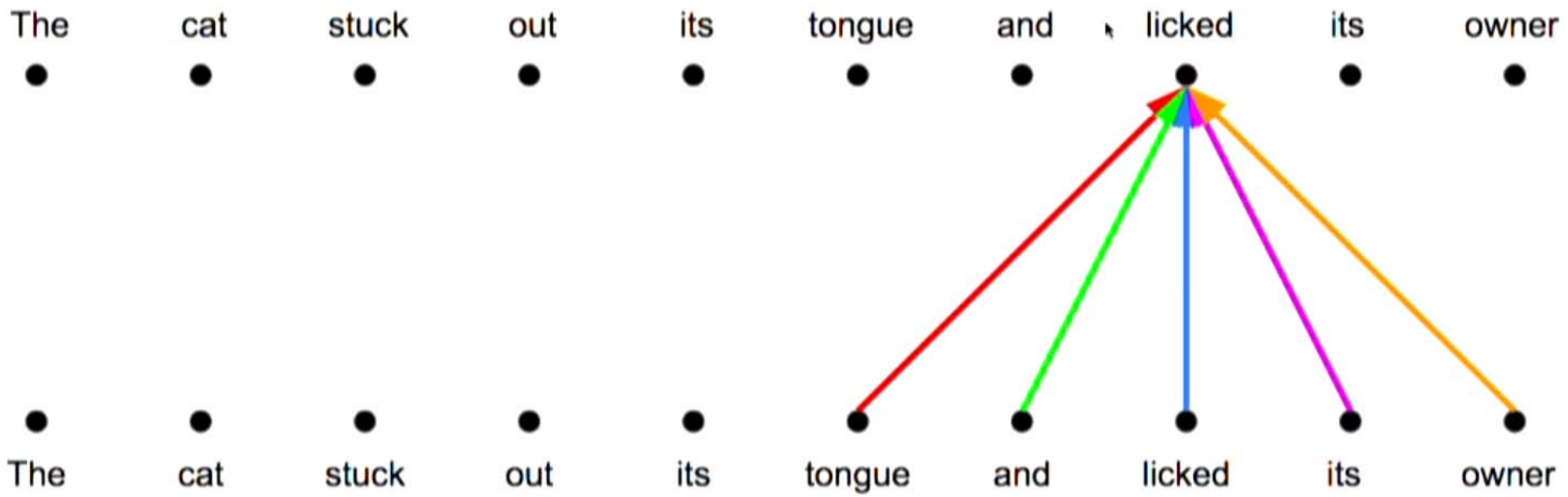
$$r \leftarrow \sum_i a_i M_i$$

Attention: a weighted average



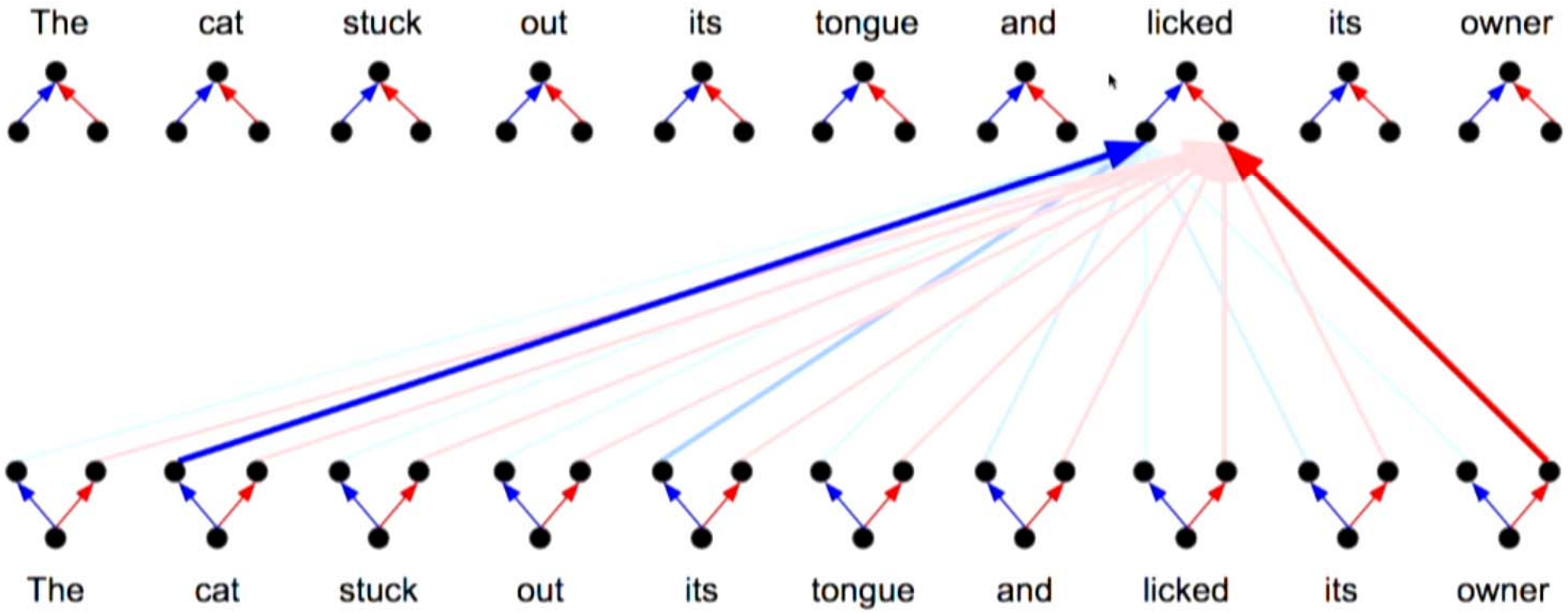
Convolution:

Different linear transformations by relative position.

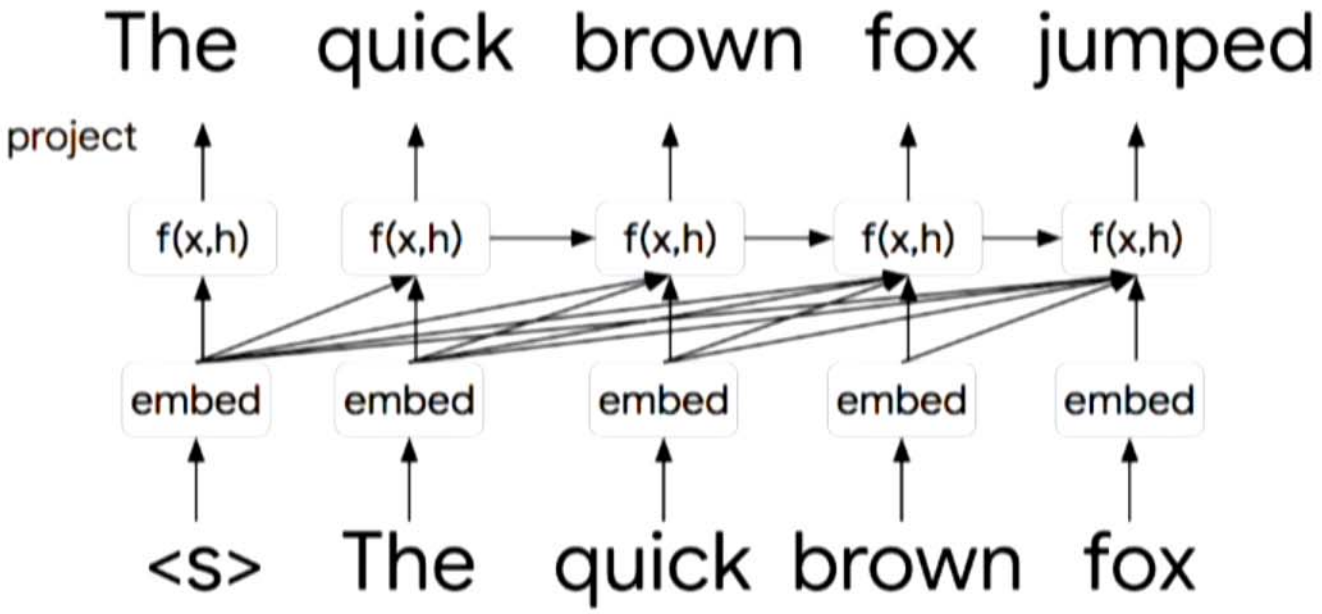


Multi-head Attention

Parallel attention layers with different linear transformations on input and output.



Language Models with attention



Attention Is All You Need

Attention Is All You Need

Ashish Vaswani[*]
Google Brain
avaswani@google.com

Noam Shazeer[*]
Google Brain
noam@google.com

Niki Parmar[*]
Google Research
nikip@google.com

Jakob Uszkoreit[*]
Google Research
usz@google.com

Llion Jones[*]
Google Research
llion@google.com

Aidan N. Gomez[*] [†]
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser[*]
Google Brain
lukaszkaiser@google.com

Illia Polosukhin[*] [‡]
illia.polosukhin@gmail.com

Attention Is All You Need

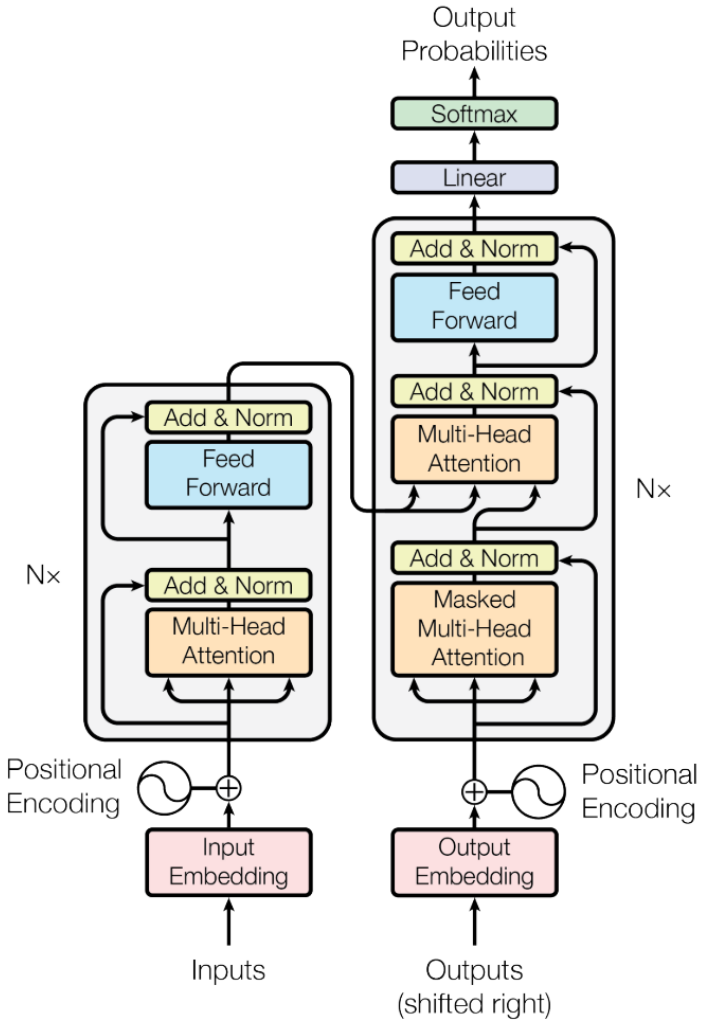
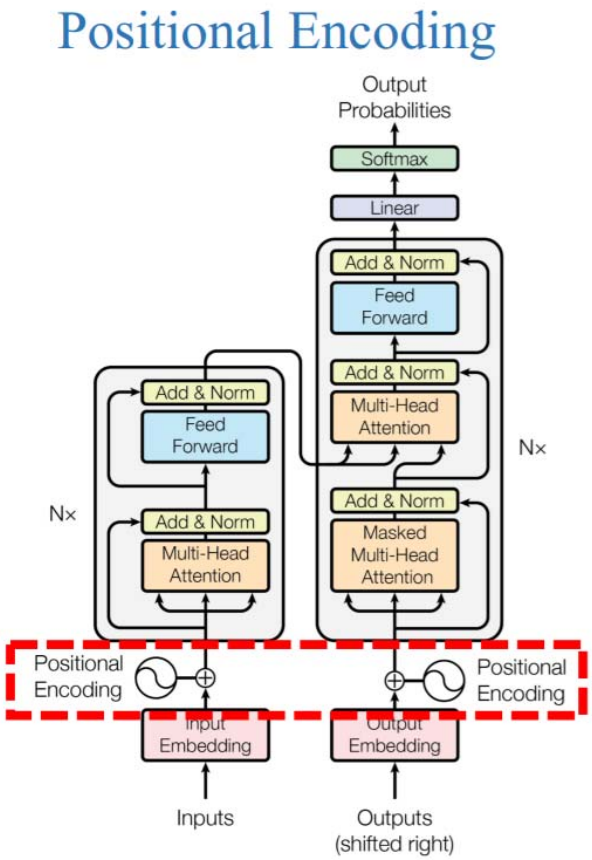


Figure 1: The Transformer - model architecture.

Attention Is All You Need



$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

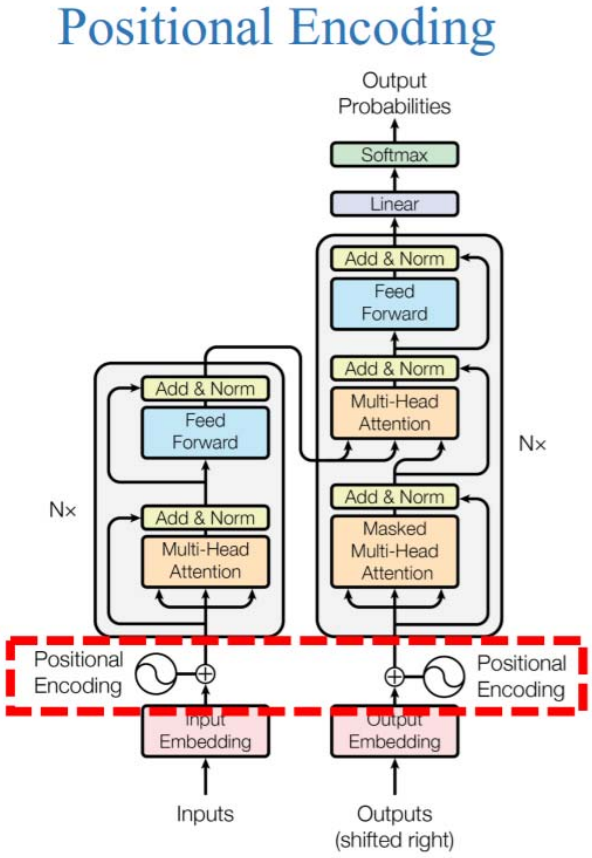
- pos is the position, and i is the dimension
- Chose this function because it would allow the model to easily learn to attend by relative positions

$$sin(\alpha+\beta) = sin\alpha cos\beta + cos\alpha sin\beta$$

$$cos(\alpha+\beta) = cos\alpha cos\beta - sin\alpha sin\beta$$

- PE(pos+k) can be represented as a linear function of PE(pos) and PE(k)

Attention Is All You Need



Why not using learned positional embeddings

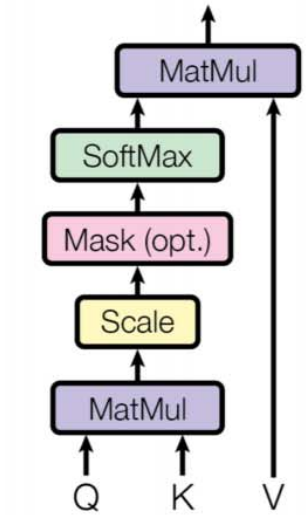
	N	d_{model}	d_{ff}	h	d_k	d_v	P_{drop}	ϵ_{ls}	train steps	PPL (dev)	BLEU (dev)	params $\times 10^6$
base	6	512	2048	8	64	64	0.1	0.1	100K	4.92	25.8	65
(D)							0.0			5.77	24.6	
							0.2			4.95	25.5	
								0.0		4.67	25.3	
								0.2		5.47	25.7	
(E)	positional embedding instead of sinusoids									4.92	25.7	

- two versions produced nearly identical results
- Sinusoids may allow the model to extrapolate to sequence lengths longer than the ones encountered during training.

Attention Is All You Need

Multi-Head Attention

Scaled Dot-Product Attention



Query Key - Values

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

$[|Q| \times d_k] \times [d_k \times |K|] \times [|K| \times d_v]$

softmax row-wise

$= [|Q| \times d_v]$

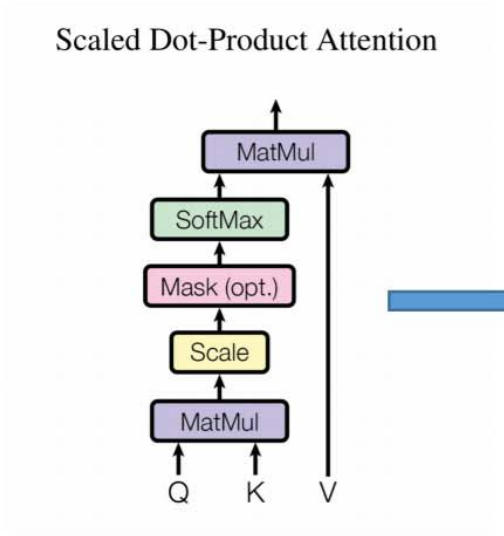
Q K^T V

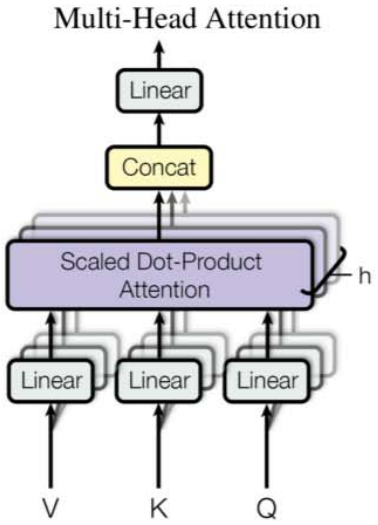
Output is computed as a weighted sum of the Values

Weight is computed by a compatibility function with Query and corresponding Key

Attention Is All You Need

Multi-Head Attention





$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,...,\text{head}_\text{h})W^O$$

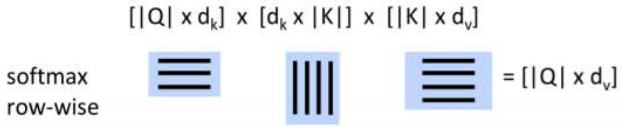
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

$$W_i^Q \in \mathbb{R}^{d_k \times \frac{d_k}{h}}$$

$$W_i^K \in \mathbb{R}^{d_k \times \frac{d_k}{h}}$$

$$W_i^V \in \mathbb{R}^{d_v \times \frac{d_v}{h}}$$

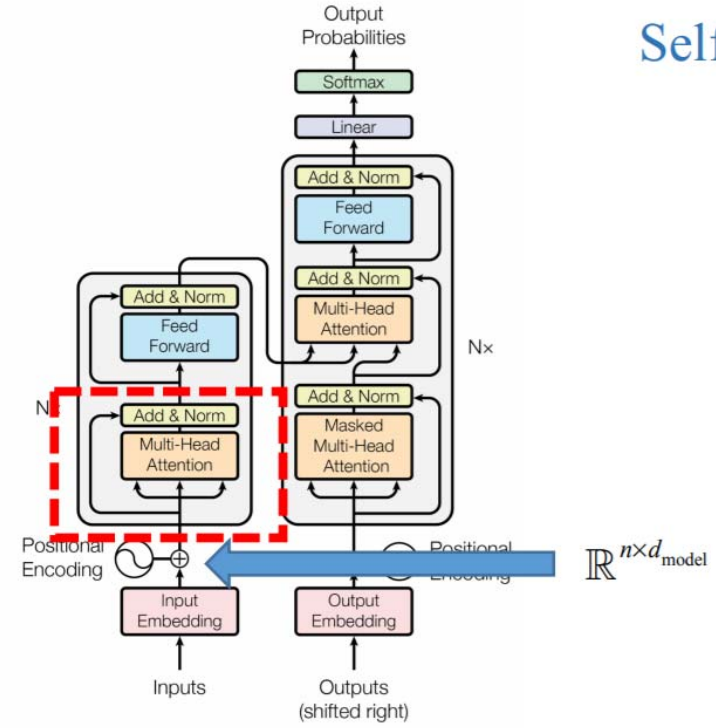
$$head_i \in \mathbb{R}^{|Q| \times \frac{d_v}{h}}$$



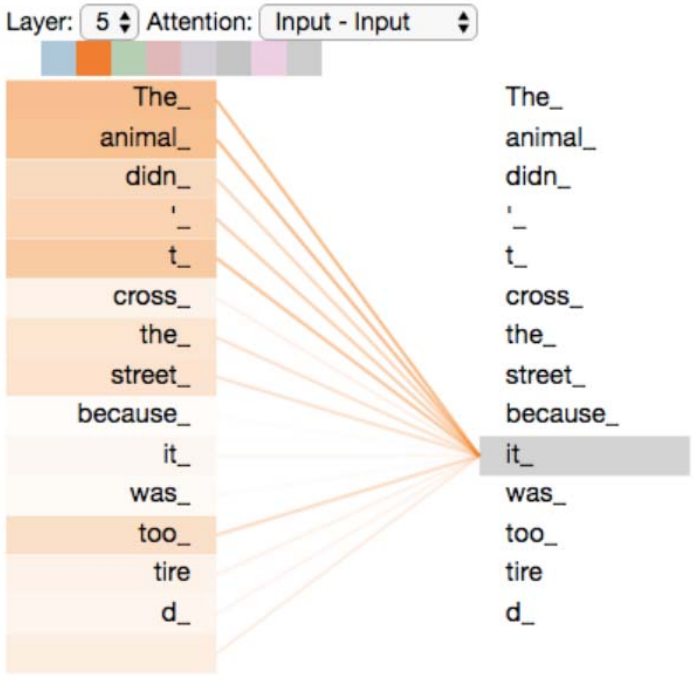
Jointly attend to information from different representation subspaces at different positions

Attention Is All You Need

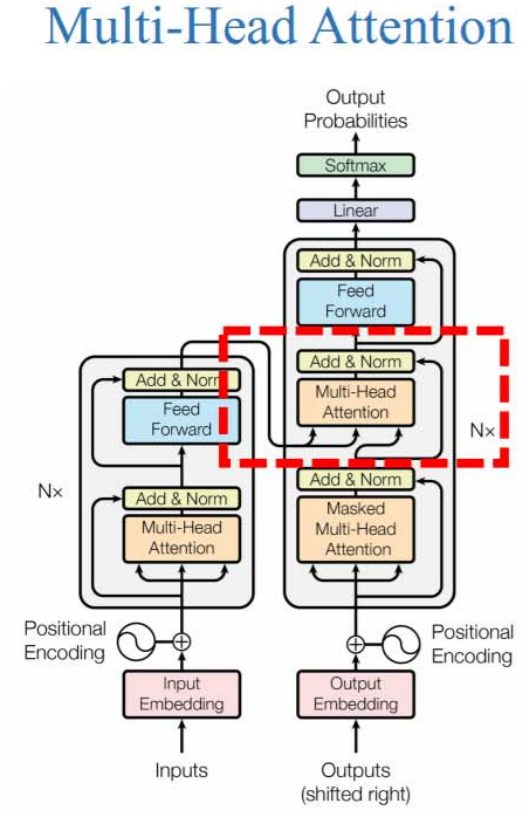
Multi-Head Attention

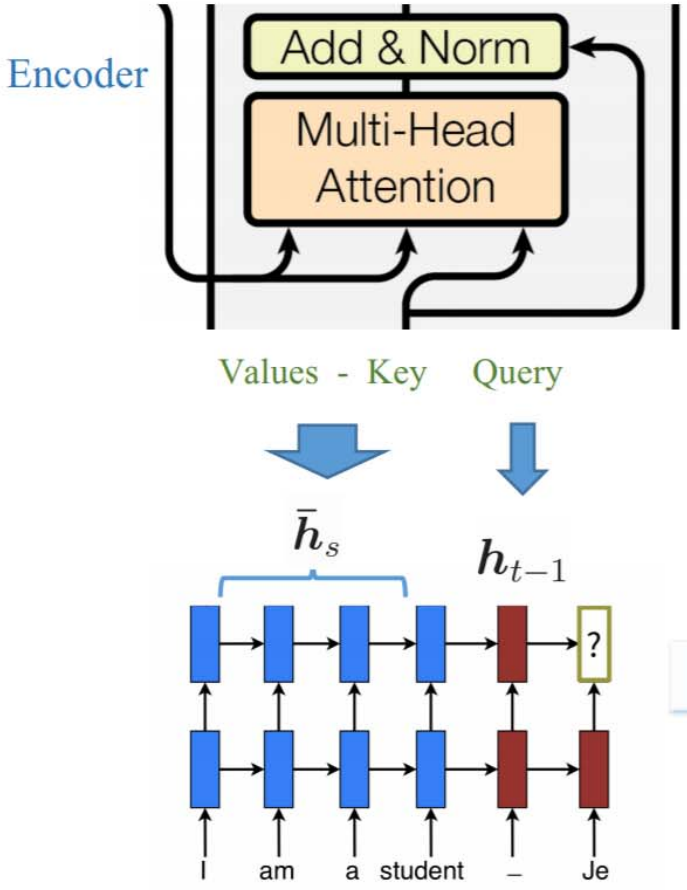


Self-Attention:

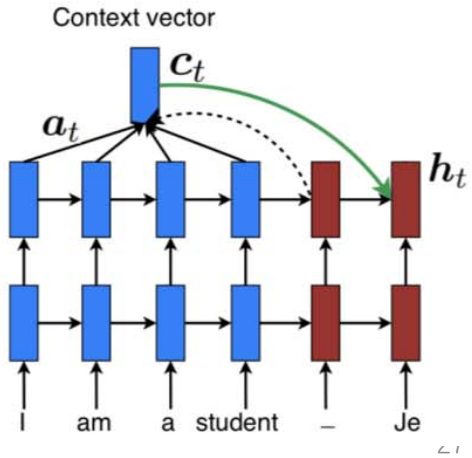


Attention Is All You Need



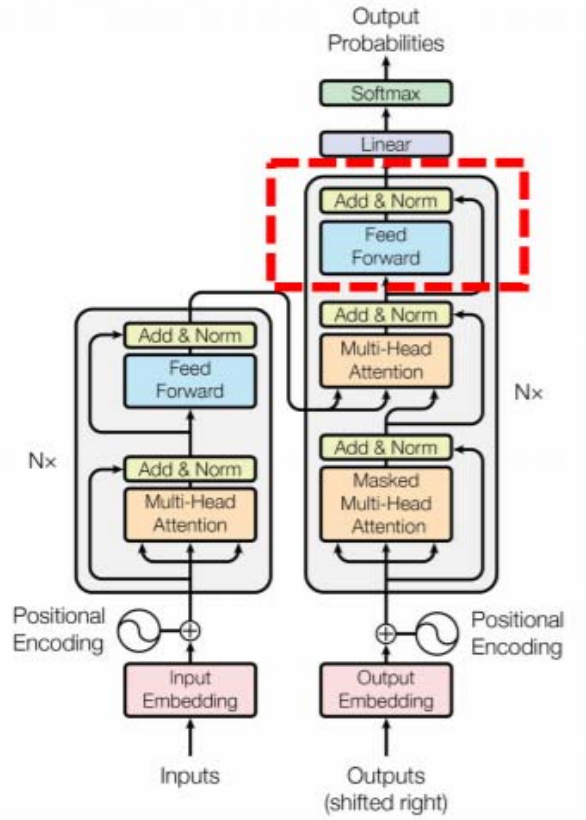


$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



Attention Is All You Need

Feed-Forward :



$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

Attention Is All You Need

Add & Norm

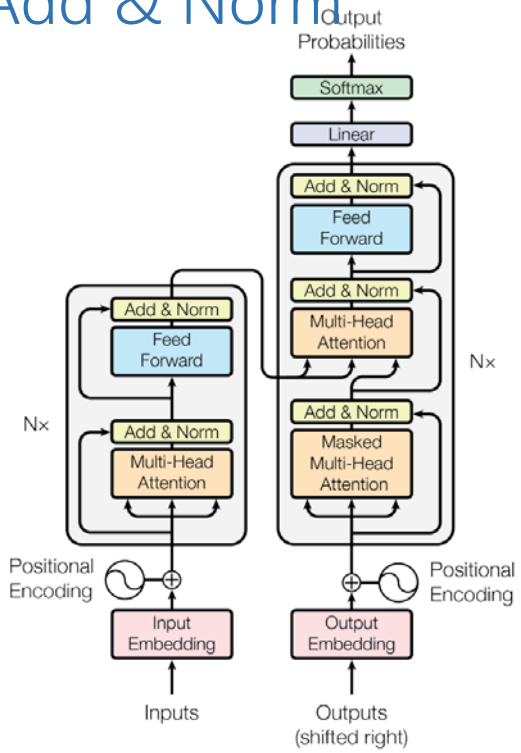
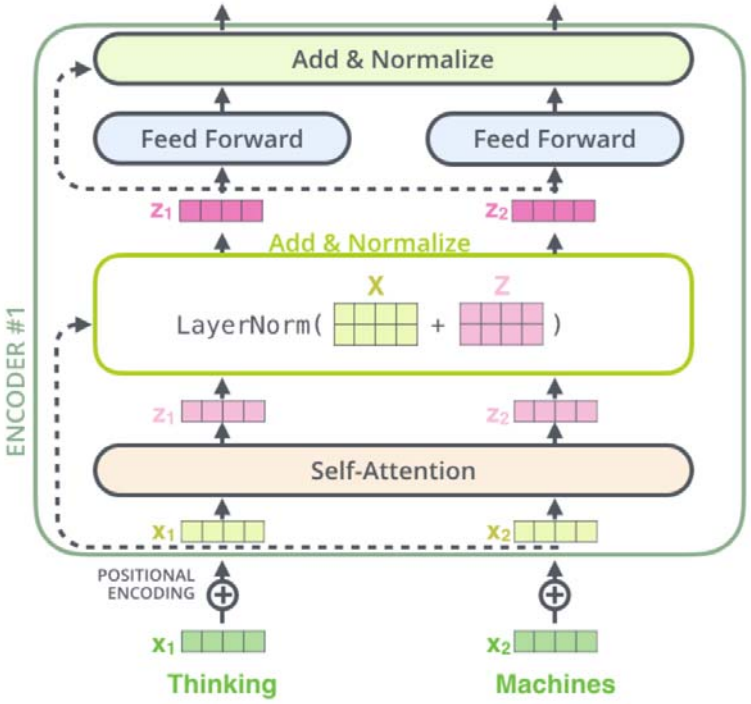


Figure 1: The Transformer - model architecture.



$$LayerNorm(x + sublayer(x))$$

- Residual connection is employed around each of the two sub-layers, followed by layer normalization
- By introducing residual connections, deeper model can be builded.
- Layernorm is much faster than Batchnorm

Attention Is All You Need

Masked Multi-Head Attention

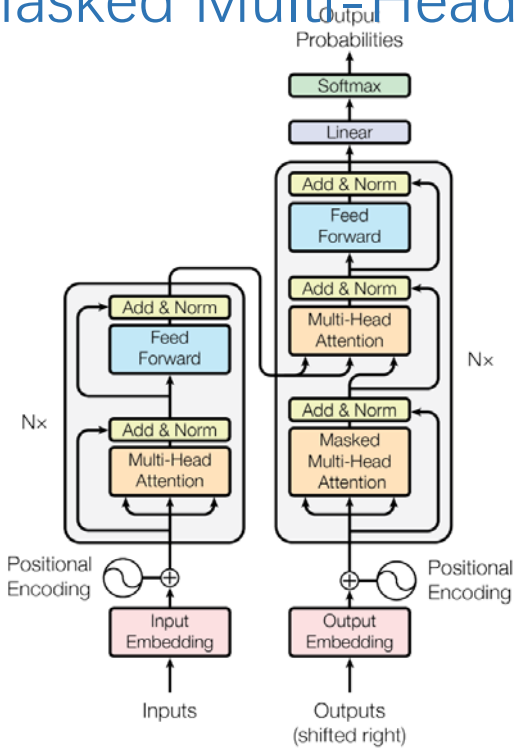
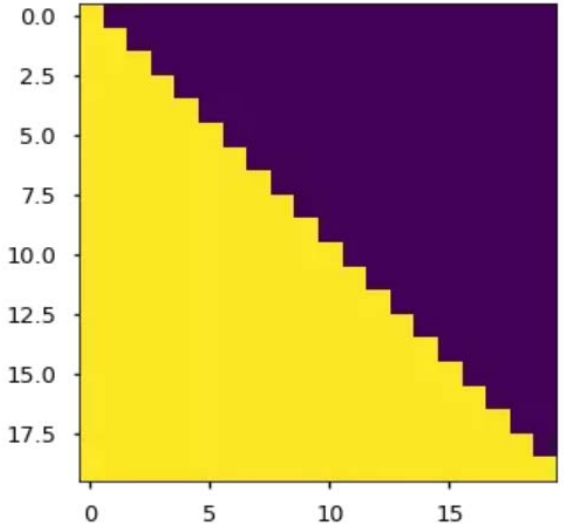


Figure 1: The Transformer - model architecture.

- In decoder, the auto-regressive property need to be preserved by preventing leftward information flow.
- Solution: Mask



Attention Is All You Need

Why self-attention ?

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

- Long dependency: self-attention layer connects all positions with a constant number of sequentially executed operations.

Attention Is All You Need

Why self-attention ?

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

- Parallelization: Encoder is natural parallel, the decoder can be parallel in the training stage.

Attention Is All You Need

Why self-attention ?

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

- Computational Complexity: When the sequence length n is smaller than the representation dimensionality d, self-attention is faster.
- Restricted Self-Attention: considering only a neighborhood of size r. It reduce the computational complexity, but increase the maximum path length.

Attention Is All You Need

Experiment

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

Attention Is All You Need

Conclusion

- Advantages
  - Learn long dependency easier than other model.
  - Efficient training.
  - Deeper model can be constructed cause the residual connection.
  - There is no problem of overlapping information that may exist within the RNN.
- Disadvantages
  - Maybe not “attention is all you need”
  - …

Think about

- What does attention mechanism learn ?
- Why do we need self attention ?

Roadmap

- Language model
- Attention
- **Elmo**
- Bert
- QANet

Deep contextualized word representations

ELMo (Embeddings from Language Models)

2018 NAACL outstanding paper award

- Propose a new deep contextualized word representation
- models both
  (1) complex characteristics of word use (e.g., syntax and semantics), and
  (2) how these uses vary across linguistic contexts (i.e., to model polysemy)
- word vectors are learned **functions of the internal states** of a deep bidirectional language model, which is pretrained on a large text corpus
- can improve existing neural models in various NLP tasks
- can capture more abstract linguistic characteristics in the higher lever of layers

- Example

	Source	Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	Chico Ruiz made a spectacular play on Alusik 's grounder {...}	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent play .
	Olivia De Havilland signed to do a Broadway play for Garson {...}	{...} they were actors who had been handed fat roles in a successful play , and had talent enough to fill the roles competently , with nice understatement .

Table 4: Nearest neighbors to "play" using GloVe and the context embeddi[illegible] from a biLM.

Glove mostly learns Sport-related context

Elmo can distinguish the word Sense based on the context

- Bidirectional language models

$$p(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} p(t_k \mid t_1, t_2, \ldots, t_{k-1})$$

$$p(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} p(t_k \mid t_{k+1}, t_{k+2}, \ldots, t_N)$$

$$\sum_{k=1}^{N} \Big(\log p(t_k \mid t_1, \ldots, t_{k-1}; \Theta_x, \overrightarrow{\Theta}_{LSTM}, \Theta_s) + \log p(t_k \mid t_{k+1}, \ldots, t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s) \Big)$$

- ELMo

$$R_k = \{\mathbf{x}_k^{LM}, \overrightarrow{\mathbf{h}}_{k,j}^{LM}, \overleftarrow{\mathbf{h}}_{k,j}^{LM} \mid j = 1, \ldots, L\}$$
$$= \{\mathbf{h}_{k,j}^{LM} \mid j = 0, \ldots, L\},$$

$$\mathbf{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM}$$

Task	Baseline	Last Only	All layers	
			λ=1	λ=0.001
SQuAD	80.8	84.7	85.0	**85.2**
SNLI	88.1	89.1	89.3	**89.5**
SRL	81.6	84.1	84.6	**84.8**

ELMo is a task specific representation. A down-stream task learns weighting parameters

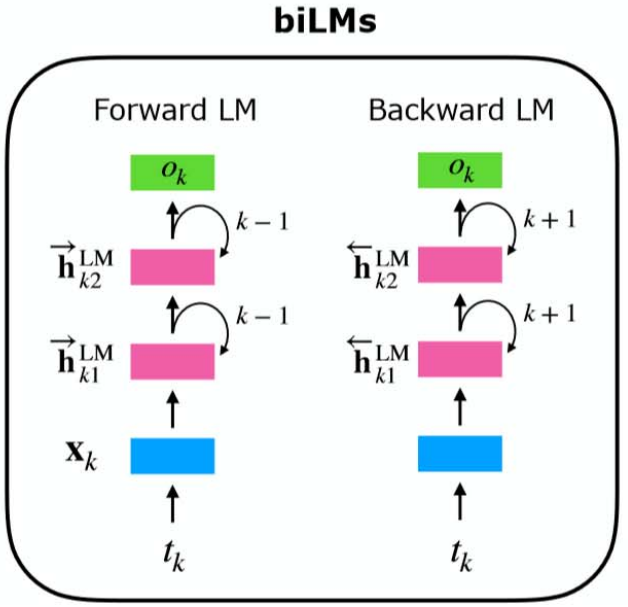
adding $\lambda\|\mathbf{w}\|_2^2$ to the loss

$$\mathbf{ELMo}_k^{task} = \gamma^{task} \times \sum \{ s_2^{task} \times \mathbf{h}_{k2}^{LM}, \; s_1^{task} \times \mathbf{h}_{k1}^{LM}, \; s_0^{task} \times \mathbf{h}_{k0}^{LM} \; ([\mathbf{x}_k; \mathbf{x}_k])$$

Concatenate hidden layers

$[\overrightarrow{\mathbf{h}}_{kj}^{LM}; \overleftarrow{\mathbf{h}}_{kj}^{LM}]$

Unlike usual word embeddings, ELMo is assigned to every *token* instead of a *type*



- Evaluation on supervised NLP tasks

Question answering, Textual entailment, Semantic role labeling
Coreference resolution, Named entity extraction, Sentiment analysis

Task	Previous SOTA		Our baseline	ELMo + baseline	Increase (absolute/ relative)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

In every task considered, simply adding ELMo establishes a new state-of-the-art result.

- Where to include Elmo

- $[\mathbf{x}_k; \mathbf{ELMo}_k^{task}]$
- $[\mathbf{h}_k; \mathbf{ELMo}_k^{task}]$

Task	Input Only	Input & Output	Output Only
SQuAD	85.1	**85.6**	84.8
SNLI	88.9	**89.5**	88.7
SRL	**84.7**	84.3	80.9

Table 3: Development set performance for SQuAD, SNLI and SRL when including ELMo at different locations in the supervised model.

- What information does biLM's representations capture

Model	F_1
WordNet 1st Sense Baseline	65.9
Raganato et al. (2017a)	69.9
Iacobacci et al. (2016)	**70.1**
CoVe, First Layer	59.4
CoVe, Second Layer	*64.7*
biLM, First layer	67.4
biLM, Second layer	*69.0*

Table 5: All-words fine grained WSD F_1. For CoVe and the biLM, we report scores for both the first and second layer biLSTMs.

Model	Acc.
Collobert et al. (2011)	97.3
Ma and Hovy (2016)	97.6
Ling et al. (2015)	**97.8**
CoVe, First Layer	*93.3*
CoVe, Second Layer	92.8
biLM, First Layer	*97.3*
biLM, Second Layer	96.8

Table 6: Test set POS tagging accuracies for PTB. For CoVe and the biLM, we report scores for both the first and second layer biLSTMs.

Higher layer biLM: Semantic
Lower layer biLM: Syntax

- Conclusion

  Language Modeling is effective in constructing contextualized representation (could be helpful for a variety of tasks);

  Outputs of all Layers are useful.

Roadmap

- Language model
- Attention
- Elmo
- **Bert**
- QANet

BERT

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

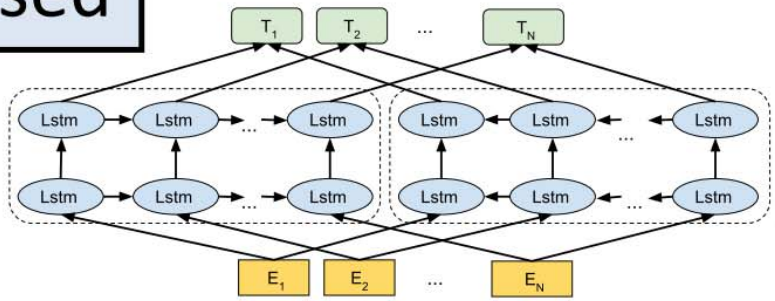
BERT

- Language model pre-training has shown to be effective for improving many natural language processing tasks
- There are two existing strategies for applying pre-trained language representations to downstream tasks:
  - Feature-based Approaches：uses tasks-specific architectures that include the pre-trained representations as additional features.
  - Fine-tuning Approaches：trained on the downstream tasks by simply fine-tuning the pretrained parameters

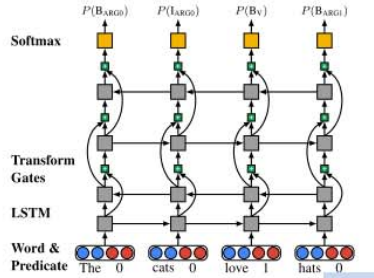
BERT

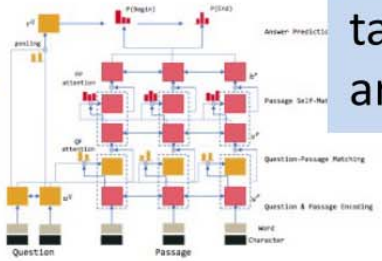
Feature-based

ELMo



shallow concatenation of left-to-right and right-to-left



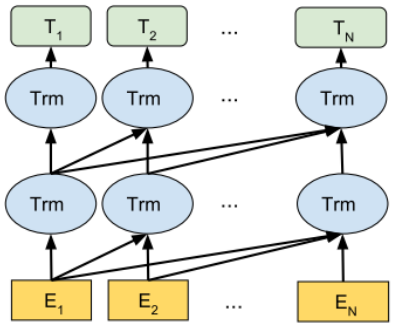


task-specific architecture

BERT

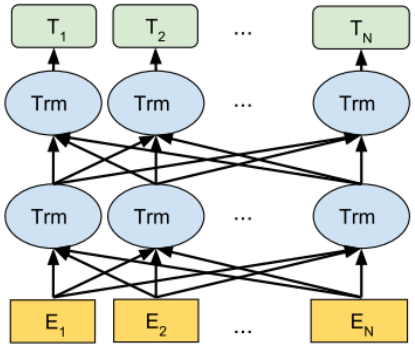
Fine-tuning

GPT

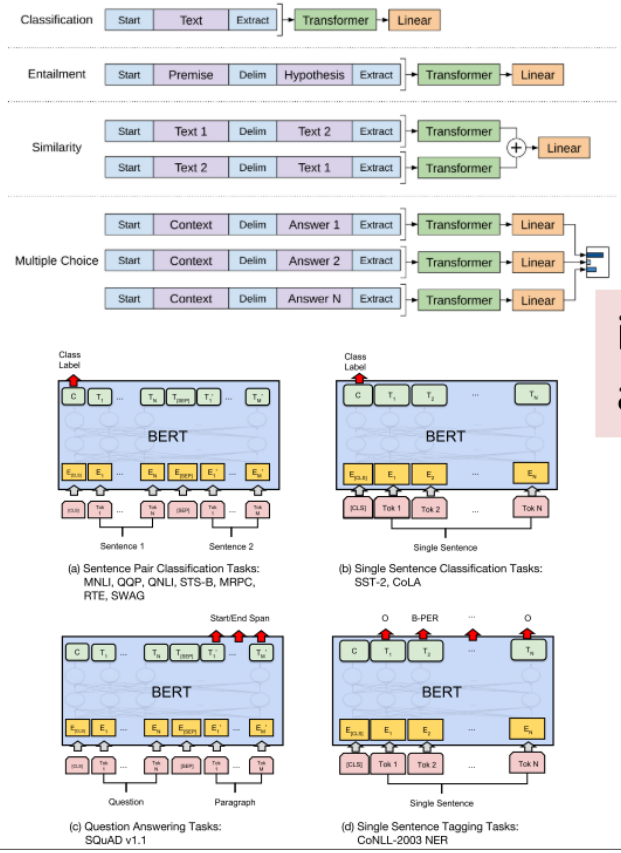


left-to-right language model

BERT

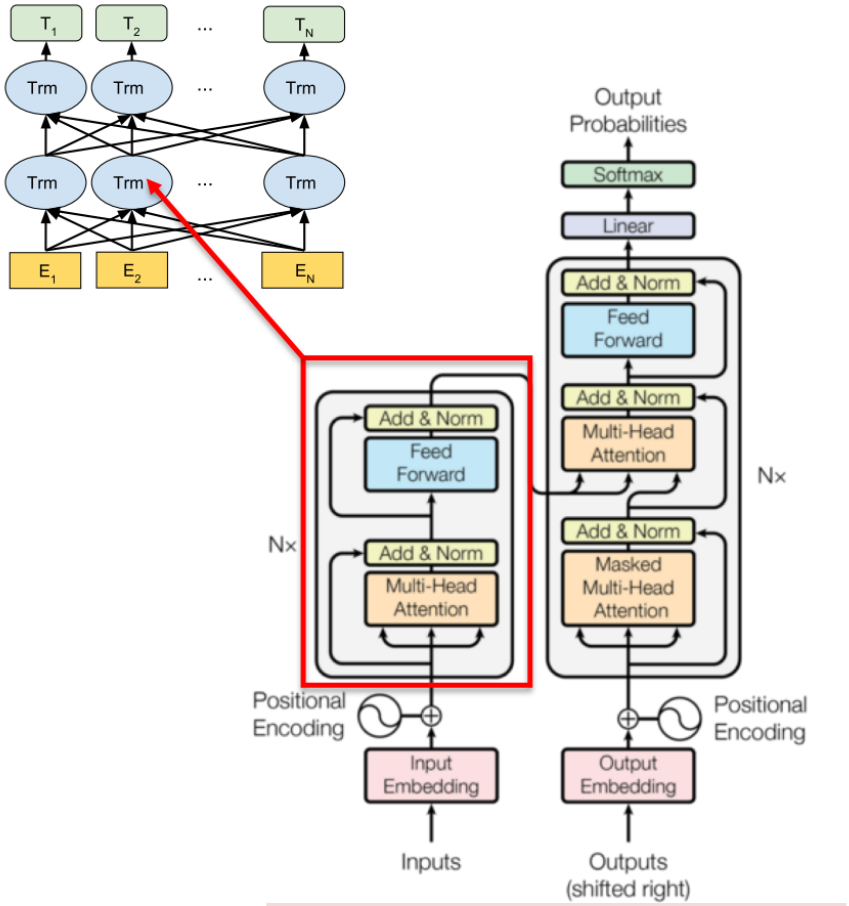


bidirectional conditioning



integrated architecture

BERT-Model Architecture



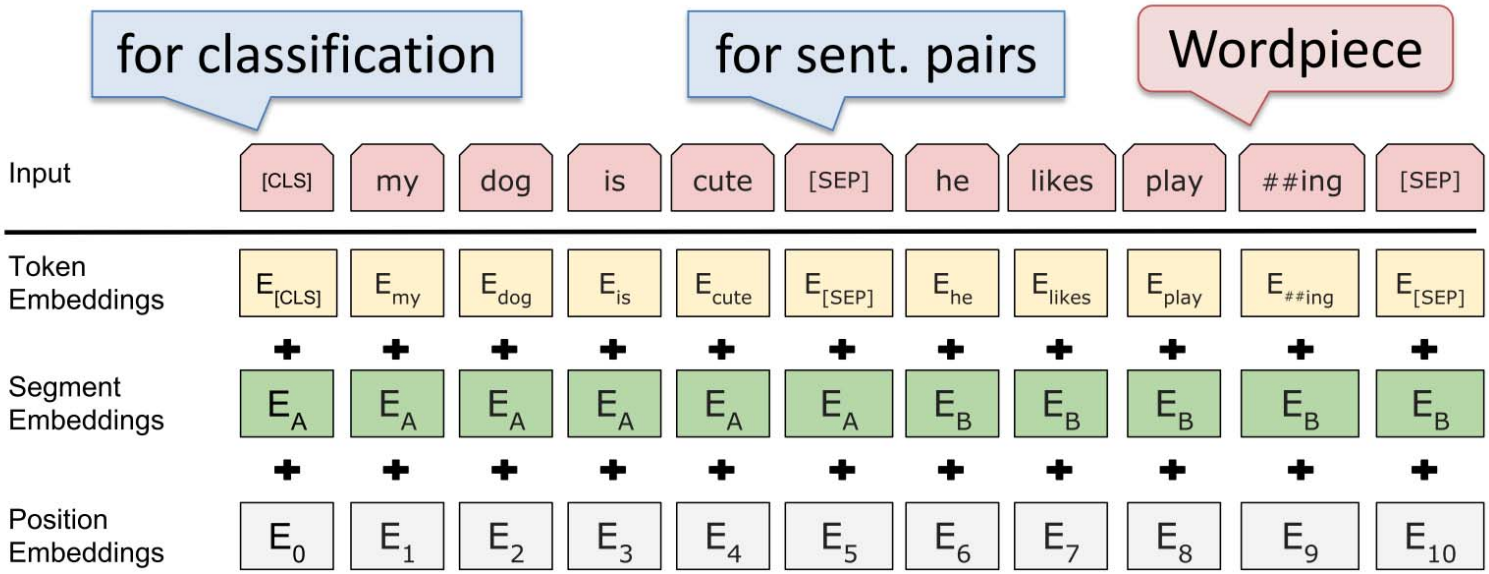
Transformer [Vaswani+ 2017]

- *L*: # of layers
- *H*: hidden size
- *A*: # of self-attention heads

same as GPT

- BERT$_{\text{BASE}}$: *L*=12, *H*=768, *A*=12
- BERT$_{\text{LARGE}}$: *L*=24, *H*=1024, *A*=16

BERT-Input Representation



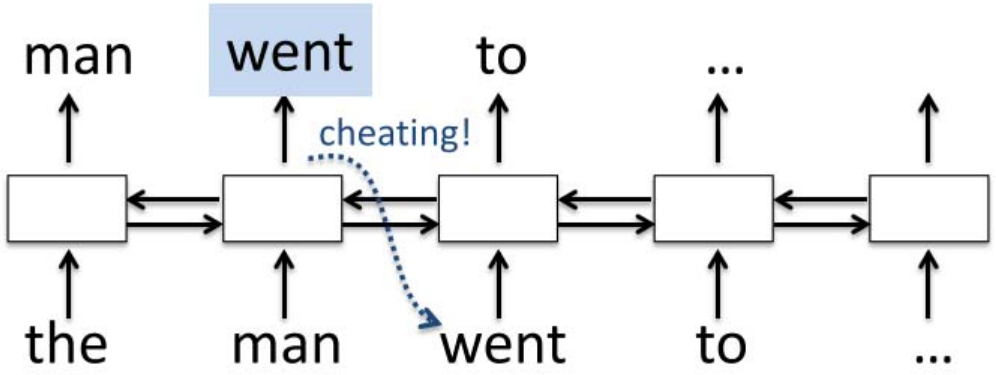
- Sentence pairs are packed together into a single sequence, The sentences are differentiated by [SEP] and segment embeddings.
- Position Embeddings: Learnable parameters, initialize randomly (different from transformer). The embedding supported sequence lengths up to 512 tokens.
- The final hidden state corresponding to [CLS] is used as the aggregate sequence representation for classification tasks.
- 30,000 token vocabulary.

BERT-Pretrained Task

- Standard Language Model (LM) is left-to-right or right-to-light

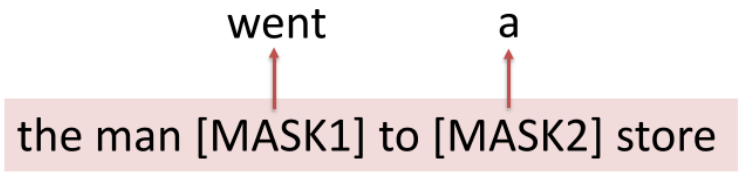
  → “deeply bidirectional” is better

- If deeply bidirectional conditioning is adopted in a standard LM, “see itself” problem arises



BERT-Pretrained Task 1

- Solution: Masked LM (Cloze Task)
- Mask 15% tokens, and predict them given deep bidirectional representations
- But, the strategy introducing mismatch between pre-training and finetuning.
- Solution:
  - 80% of the time: Replace the word with the [MASK] token
  - 10% of the time: Replace the word with a random word
  - 10% of the time: Keep the word unchanged,



BERT-Pretrained Task 2

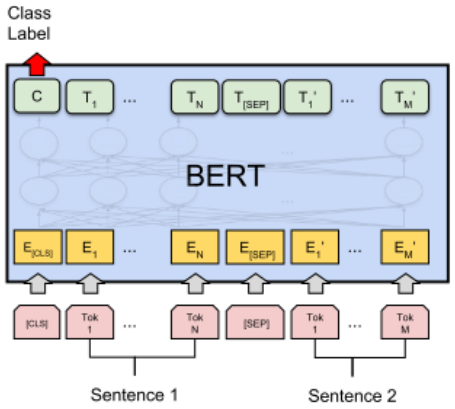
- Understanding the relation between sentences is important in QA and Inference

→ Next sentence prediction task

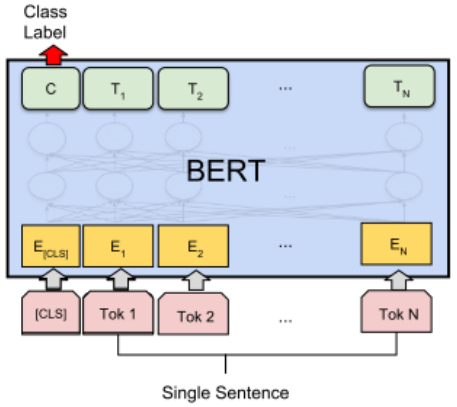
[CLS] the man went to the store [SEP] he bought a gallon of milk
Label: IsNext

[CLS] the man [MASK] to the store [SEP] penguin [MASK] are flight ##less birds [SEP]
Label: NotNext

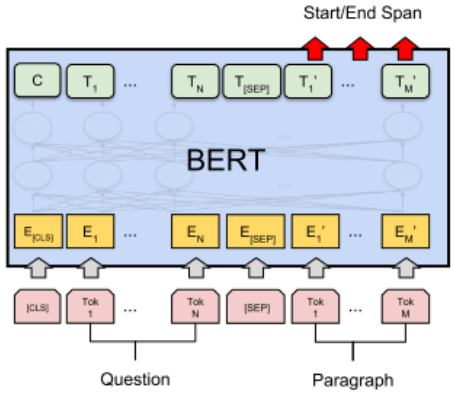
BERT-Fine Tuning



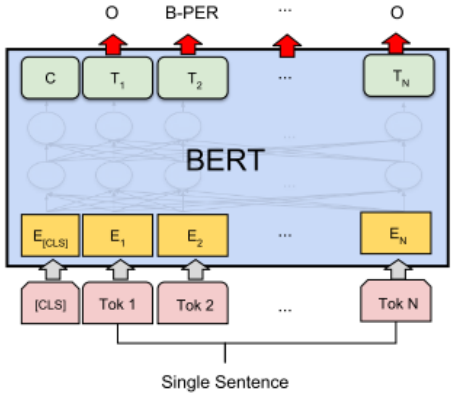
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



(b) Single Sentence Classification Tasks: SST-2, CoLA



(c) Question Answering Tasks: SQuAD v1.1



(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

BERT-GLUE Result

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**91.1**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**81.9**

Table 1: GLUE Test results, scored by the GLUE evaluation server. The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set. OpenAI GPT = (L=12, H=768, A=12); $\text{BERT}_{\text{BASE}}$ = (L=12, H=768, A=12); $\text{BERT}_{\text{LARGE}}$ = (L=24, H=1024, A=16). BERT and OpenAI GPT are single-model, single task. All results obtained from https://gluebenchmark.com/leaderboard and https://blog.openai.com/language-unsupervised/.

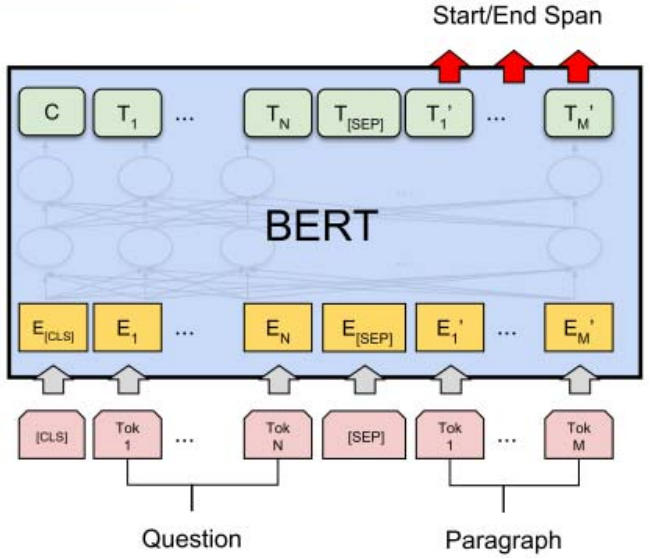
BERT-SQUAD Result

start vector

token embedding

$$P_i = \frac{e^{S \cdot T_i}}{\sum_j e^{S \cdot T_j}}$$

start prob.



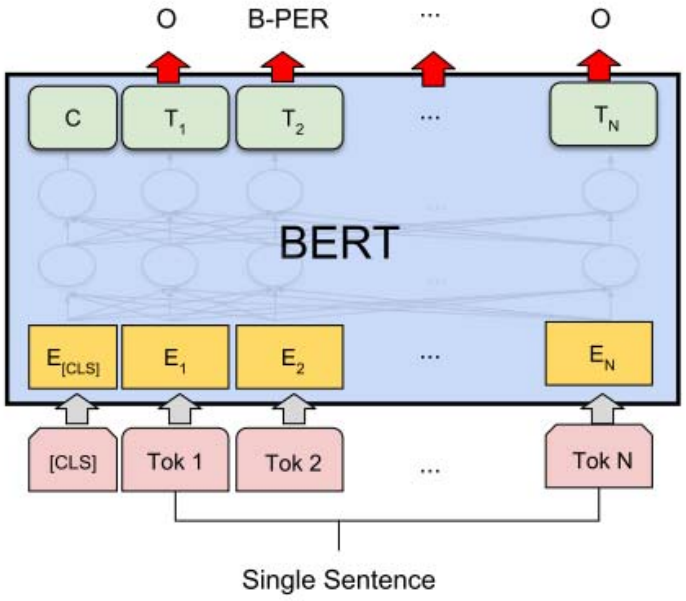
System	Dev EM	Dev F1	Test EM	Test F1
Leaderboard (Oct 8th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
#1 Single - nlnet	-	-	83.5	90.1
#2 Single - QANet	-	-	82.5	89.3
Published				
BiDAF+ELMo (Single)	-	85.8	-	-
R.M. Reader (Single)	78.9	86.3	79.5	86.6
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

BERT-Name Entity Recognition Result

```
Jim    Hen    ##son  was  a  puppet  ##eer
I-PER  I-PER  X      O    O  O       X
```

no prediction

no prediction



System	Dev F1	Test F1
ELMo+BiLSTM+CRF	95.7	92.2
CVT+Multi (Clark et al., 2018)	-	92.6
$\text{BERT}_{\text{BASE}}$	96.4	92.4
$\text{BERT}_{\text{LARGE}}$	**96.6**	**92.8**

BERT - Ablation Studies

Tasks	Dev Set MNLI-m (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	SQuAD (F1)
$\text{BERT}_{\text{BASE}}$	84.4	88.4	86.7	92.7	88.5
No NSP	83.9	84.9	86.5	92.6	87.9
LTR & No NSP	82.1	84.3	77.5	92.1	77.8
+ BiLSTM	82.1	84.1	75.7	91.6	84.9

Hyperparams				Dev Set Accuracy		
#L	#H	#A	LM (ppl)	MNLI-m	MRPC	SST-2
3	768	12	5.84	77.9	79.8	88.4
6	768	3	5.24	80.6	82.2	90.7
6	768	12	4.68	81.9	84.8	91.3
12	768	12	3.99	84.4	86.7	92.9
12	1024	16	3.54	85.7	86.9	93.3
24	1024	16	3.23	86.6	87.8	93.7

- **No NSP**: A model which is trained using the "masked LM" (MLM) but without the "next sentence prediction" (NSP)
- **LTR & No NSP**: A model which is trained using a Left-to-Right (LTR) LM, rather than an MLM
- **+BiLSTM**: adding a randomly initialized BiLSTM on top of "LTR & No NSP" for finetuning

- The larger the model, the better the performance.

BERT-Feature-Based Approach with BERT

Layers	Dev F1
Finetune All	96.4
First Layer (Embeddings)	91.0
Second-to-Last Hidden	95.6
Last Hidden	94.9
Sum Last Four Hidden	95.9
Concat Last Four Hidden	96.1
Sum All 12 Layers	95.5

CoNLL-2013 NER

only 0.3 F1
→ BERT is also effective for the feature-based approach

- The advantages of the feature-based approach
  - not all NLP tasks can be easily be represented by a Transformer encoder architecture
  - computational benefits

Conclusion

- Very deep pretrained bidirectional language model can benefit the downstream task model.

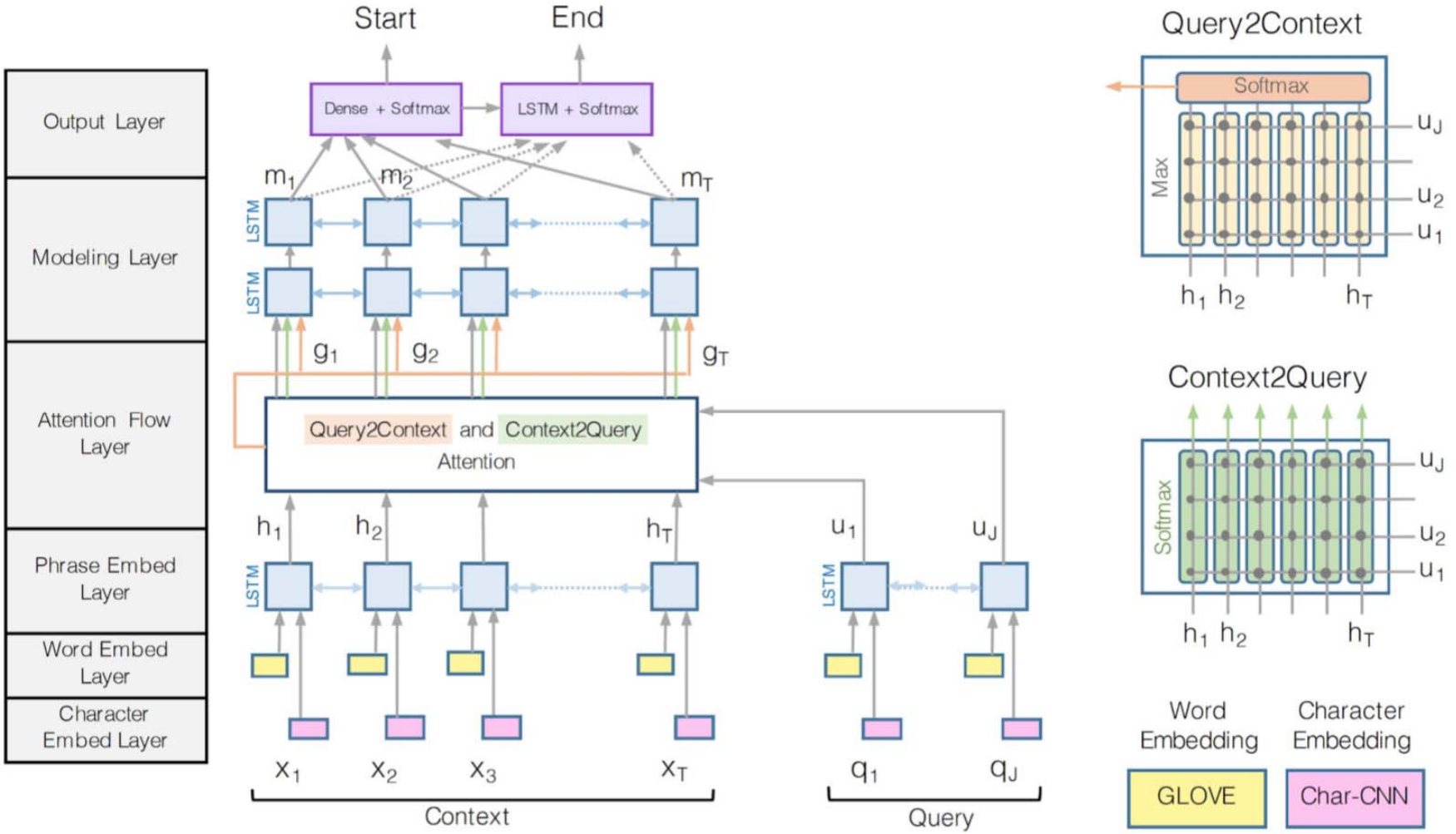
Think about

- What core information of the language dose BERT capture?
- How does this information work on other tasks?

Roadmap

- Language model
- Attention
- Elmo
- Bert
- **QANet**

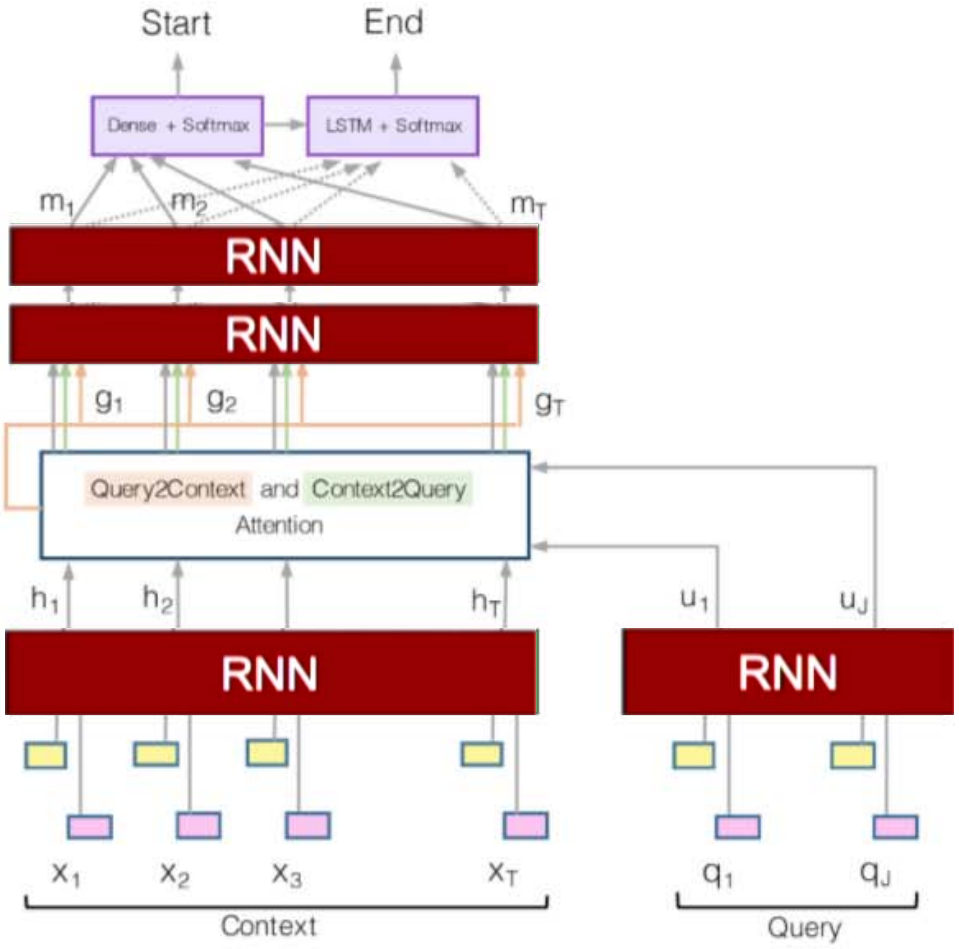
General framework neural QA Systems

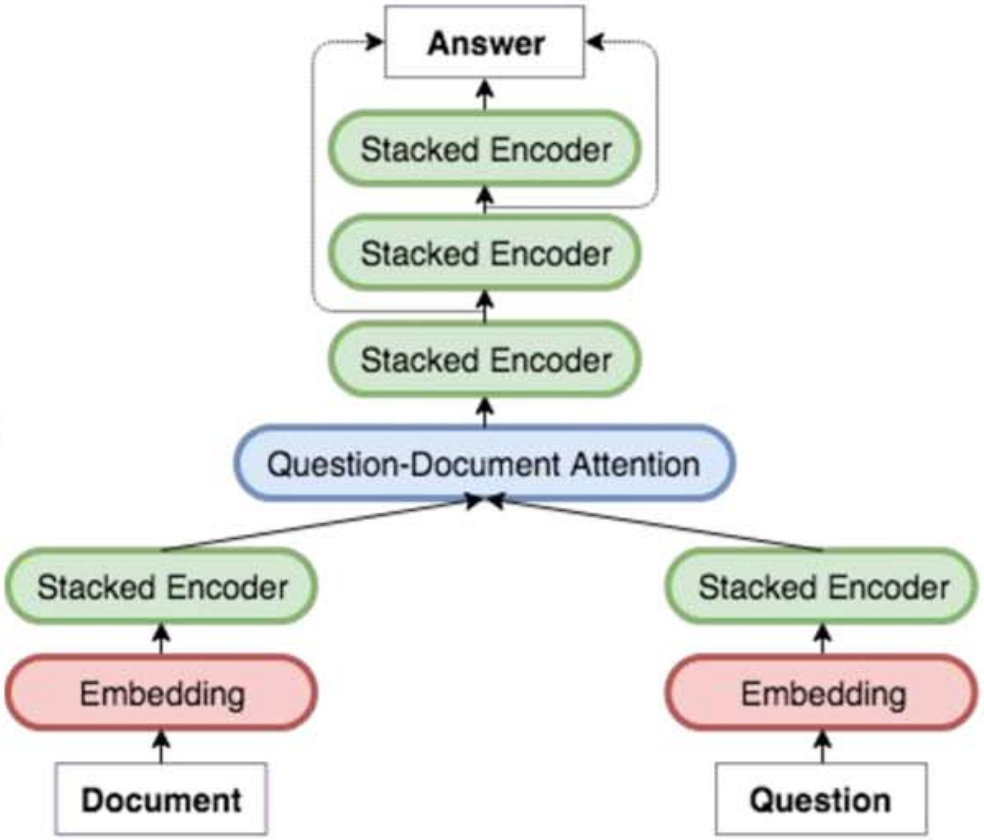


Bi-directional Attention Flow (BiDAF)

[Seo et al., ICLR'17]

General (Doc, Question) → Answer Model

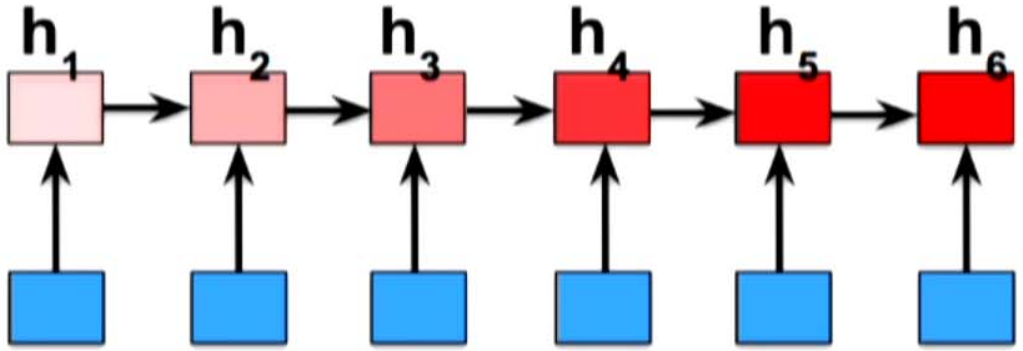




Challenges of RNN

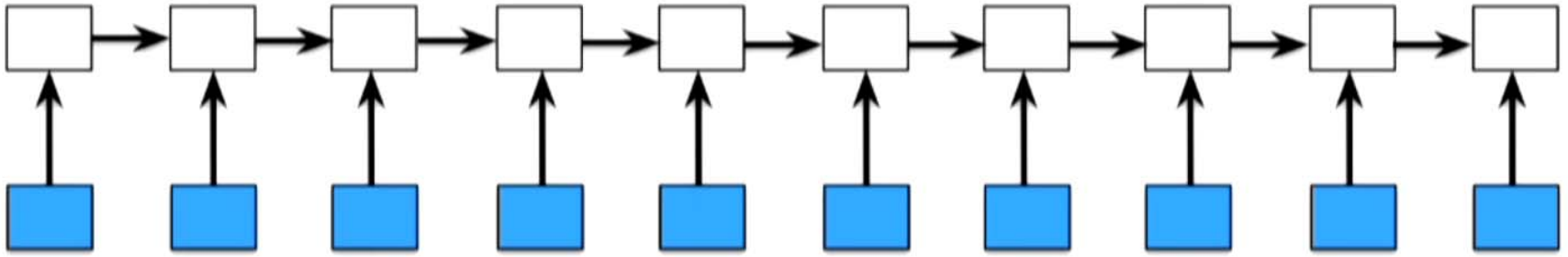
First challenge: hard to capture long dependency

Being a long-time fan of Japanese film, I expected more than this. I can't really be bothered to write too much as this movie is just so poor. The story might be the dullest romantic little something ever, only I couldn't stand the awful acting, the crass lifeless pacing, and the standard "gooey" Japanese story. If you've noticed how many Japanese movies use characters, plots and twists that seem too "different", forcedly so, then steer clear of this movie. Seriously, a 12-year old could have told you how this movie was going to move along, and that's not a good thing in my book. Fans of "Beat" Takeshi: his part in this movie is not really more than a cameo, and unless you're a rabid fan, you don't need to suffer through this waste of film.

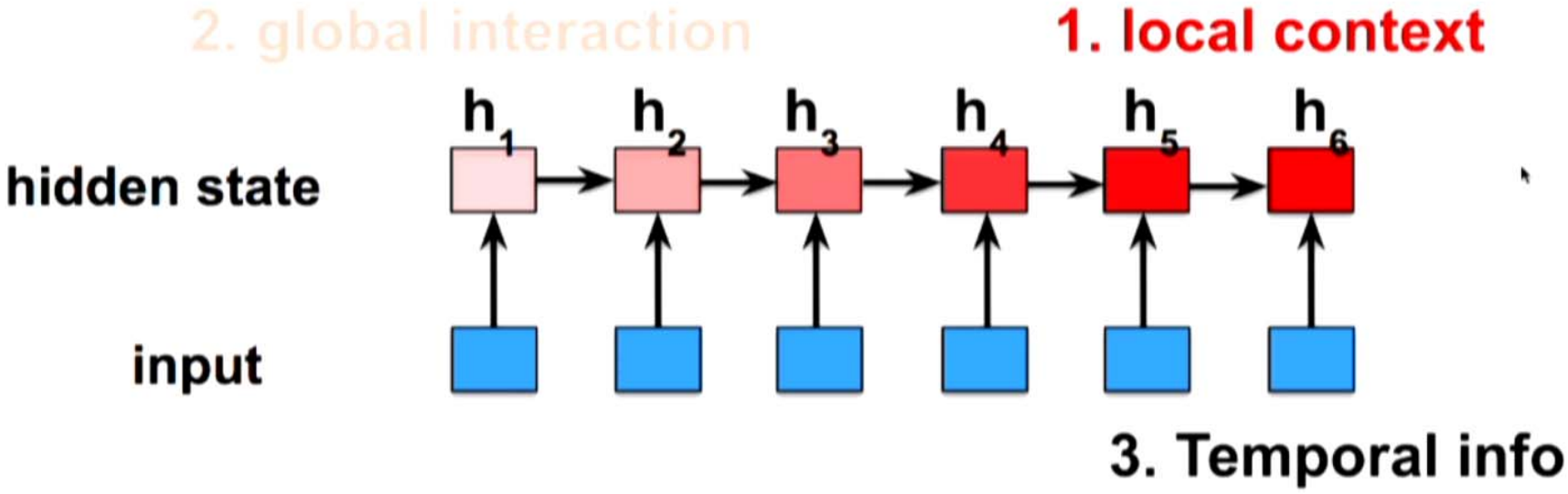


Second challenge: hard to compute in parallel

Strictly Sequential!



What do RNNs Capture?

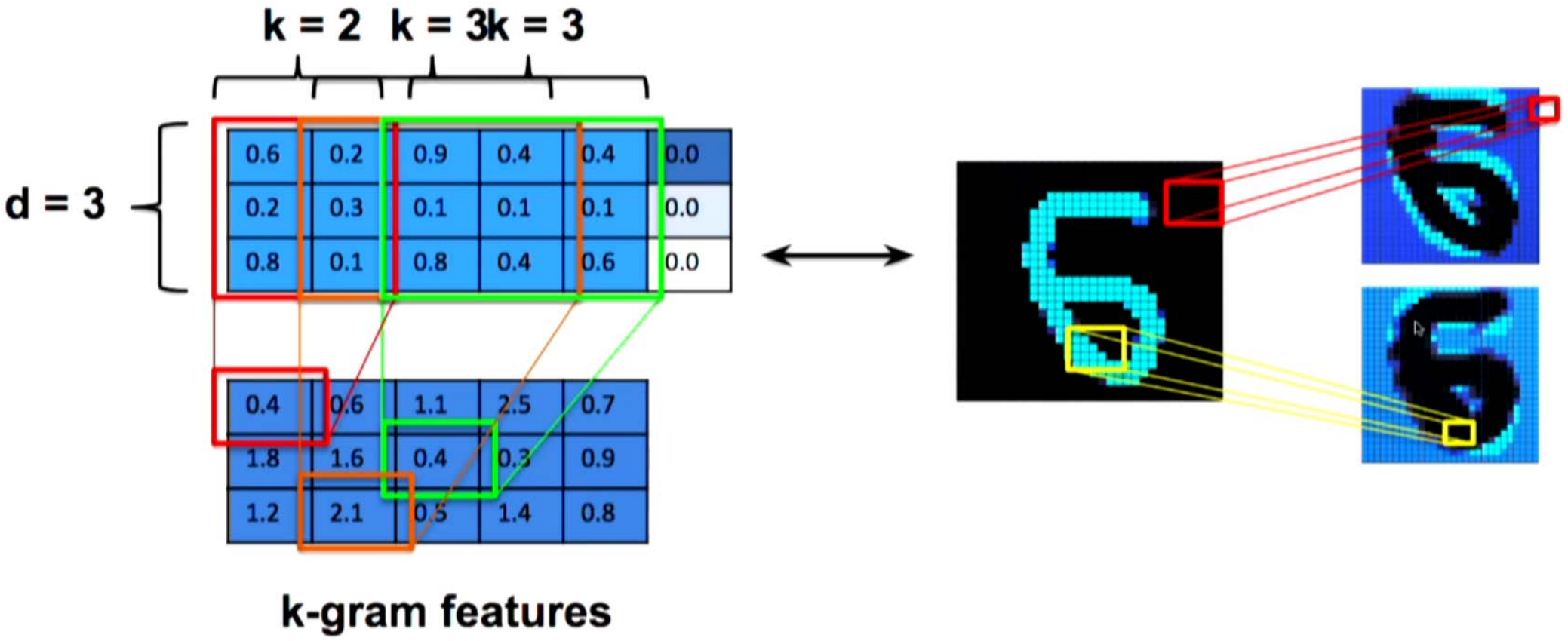


Substitution?

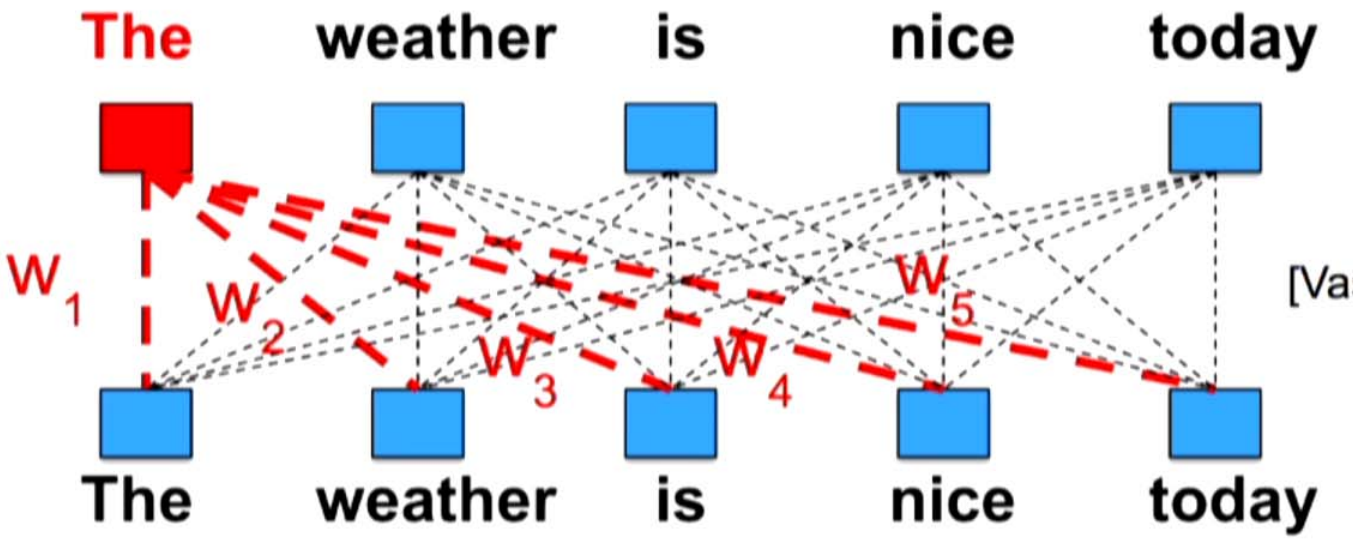
- **Idea #1:**

  Combining convolution & self-attention

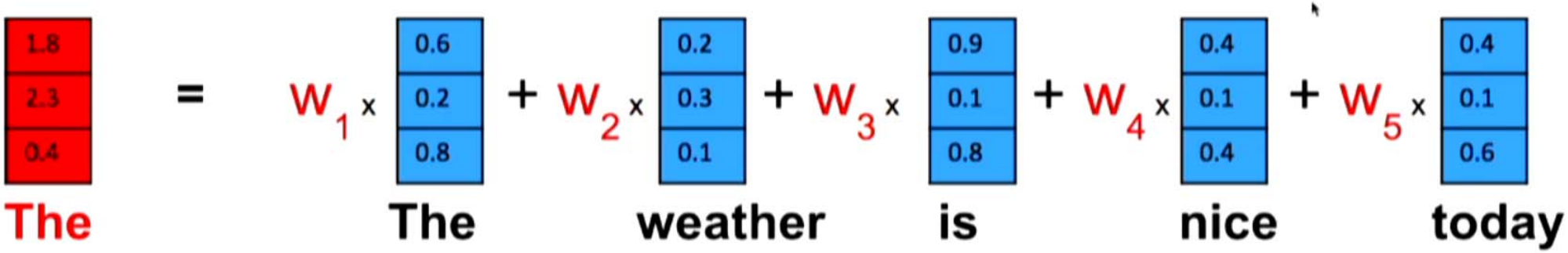
Convolution: Capturing Local Context



Fully parallel!

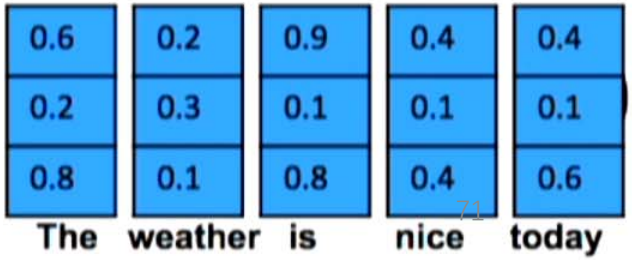


[Vaswani et al., NIPS'17]

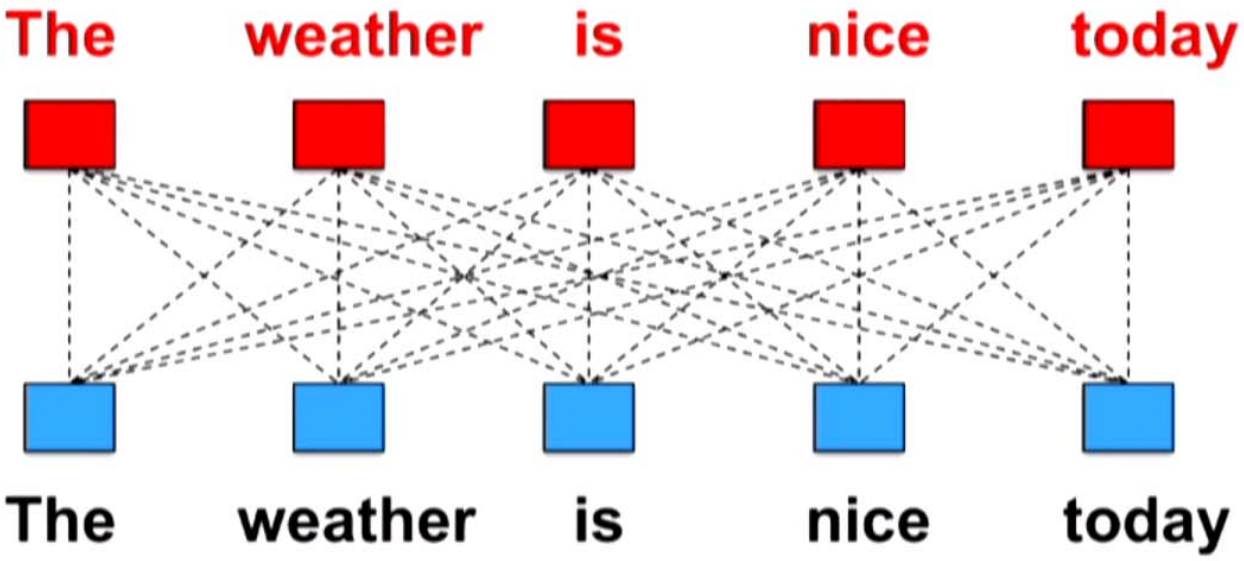


w_1, w_2, w_3, w_4, w_5 = softmax (

x

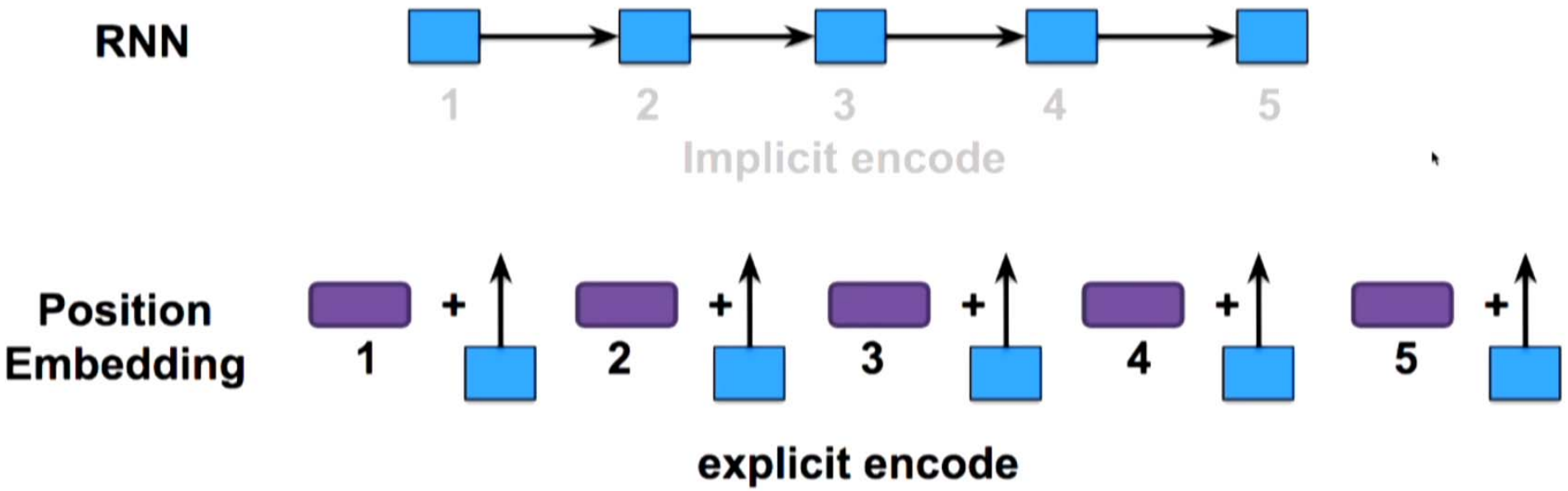


)

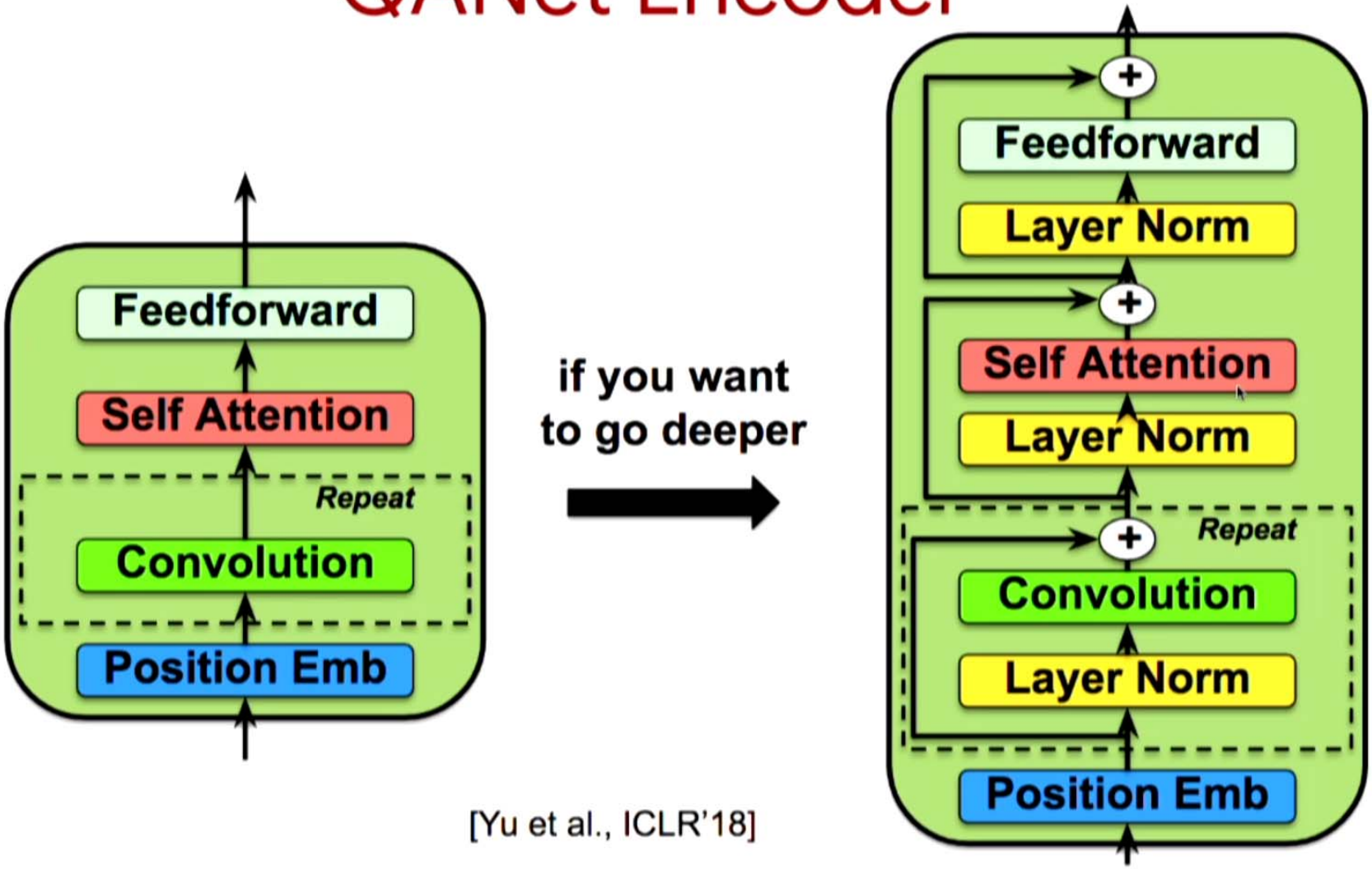


Self-attention is fully parallel & all-to-all!

Explicitly Encode Temporal Info

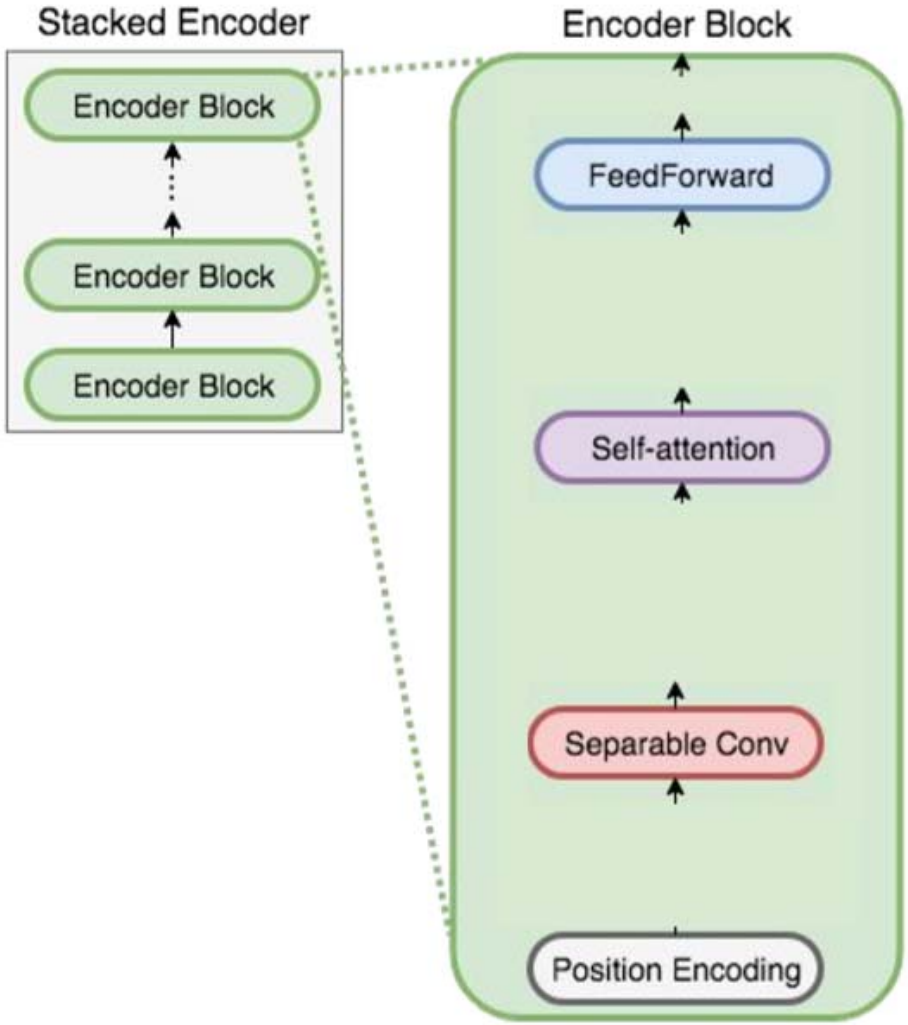


QANet Encoder



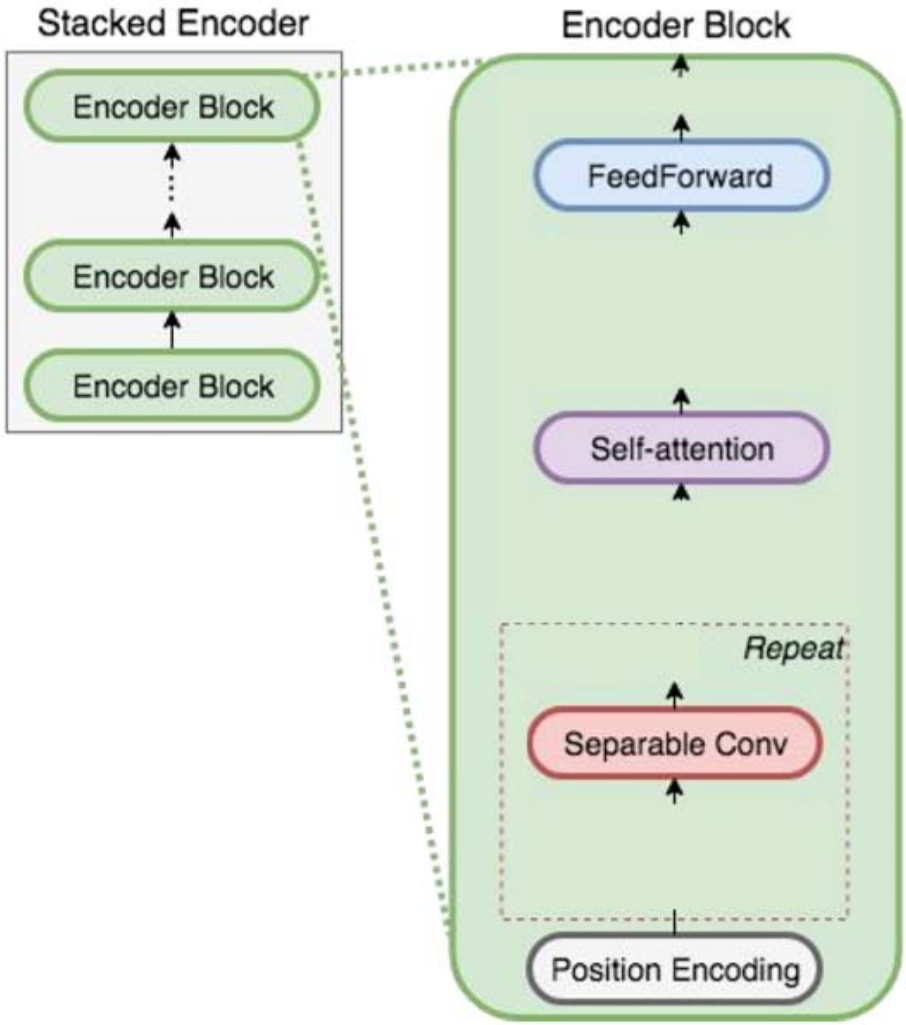
[Yu et al., ICLR'18]

QANet – No Recurrence



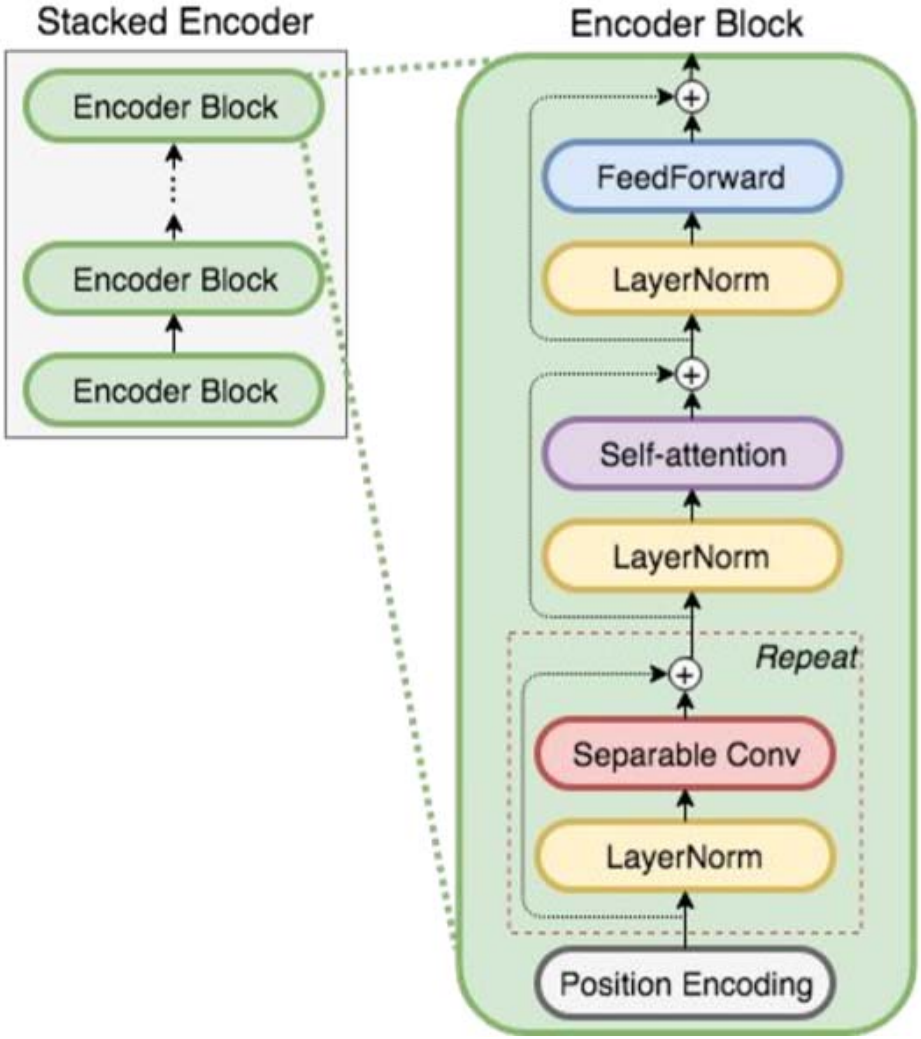
- **Very fast!**
  - Training: 3x - 13x
  - Inference: 4x - 9x

QANet – Separable Convolution



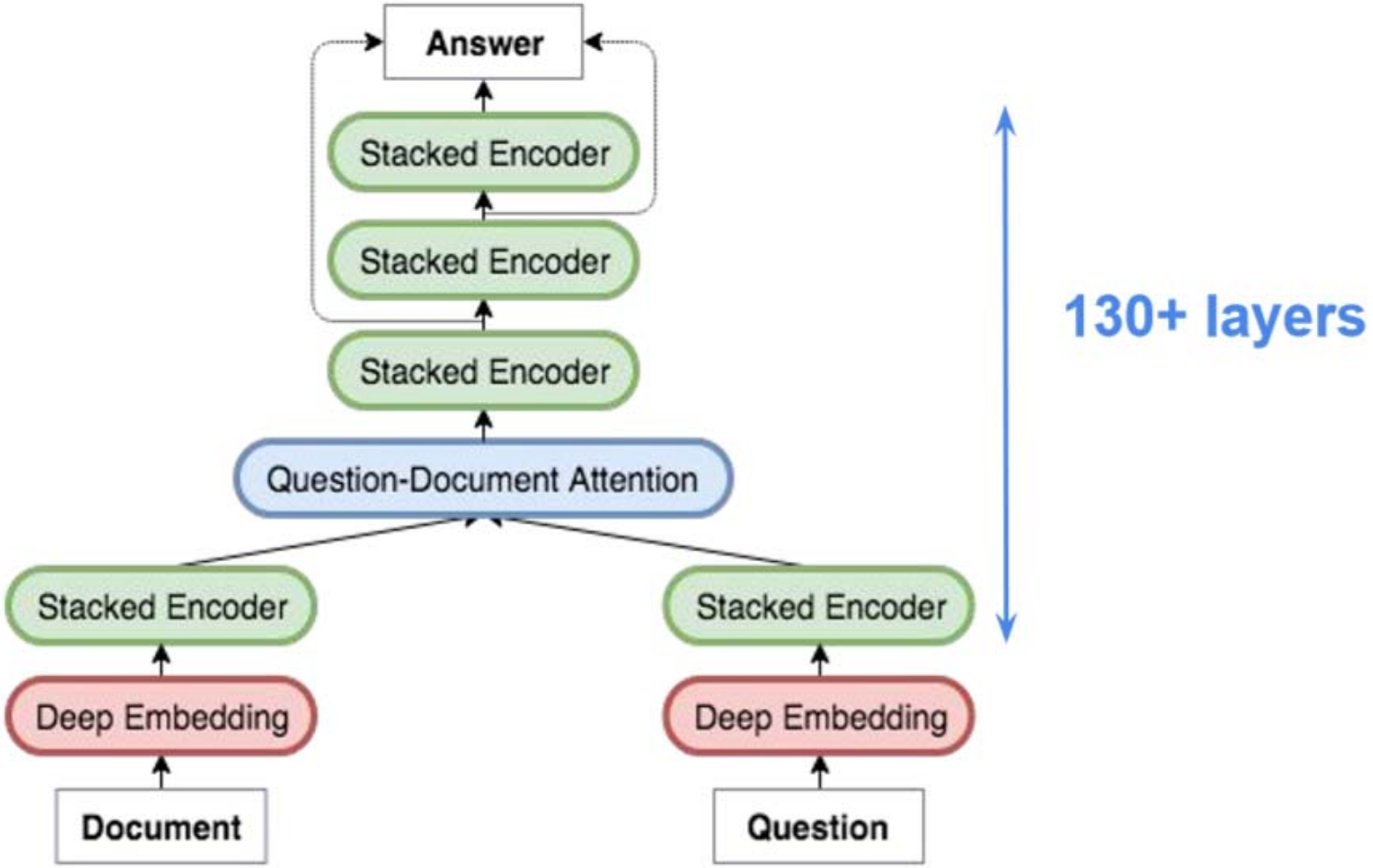
- **Faster & better:**
  - sep → normal: -0.7 F_1
- **Important:**
  - No separable conv: -2.7 F_1
  - No self-attention: -1.3 F_1

QANet – Lots of Regularizations



- Layer norm everywhere
- Residual connections everywhere
- L_2 regularization

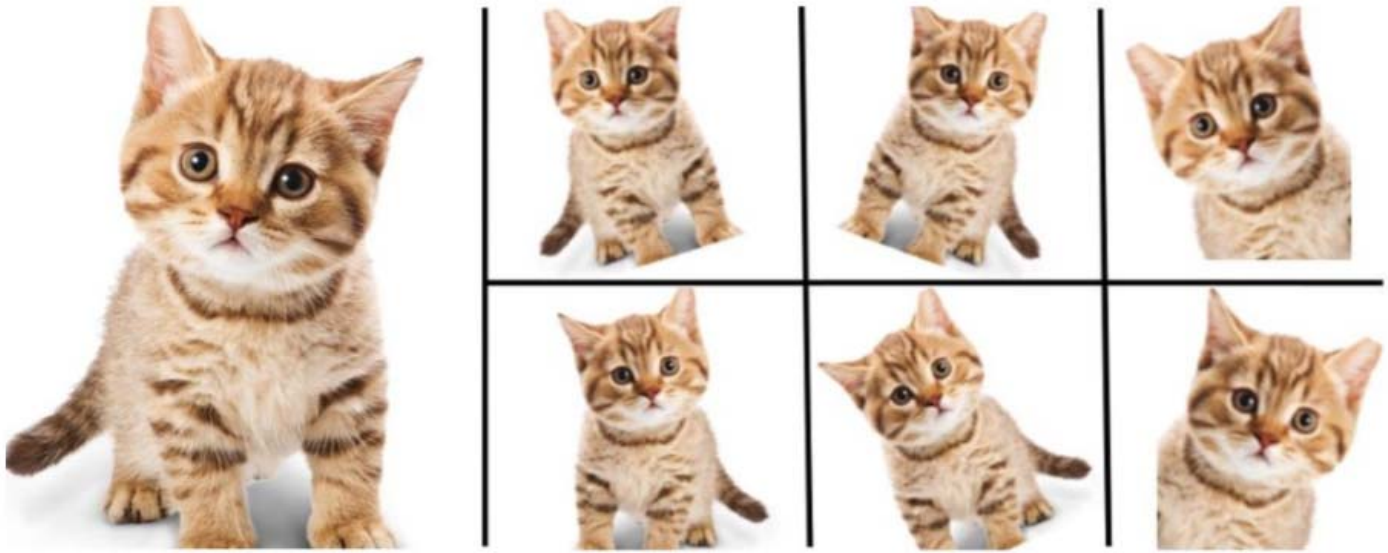
QANet – 130+ layers (Deepest NLP NN)



- **Idea #2:**

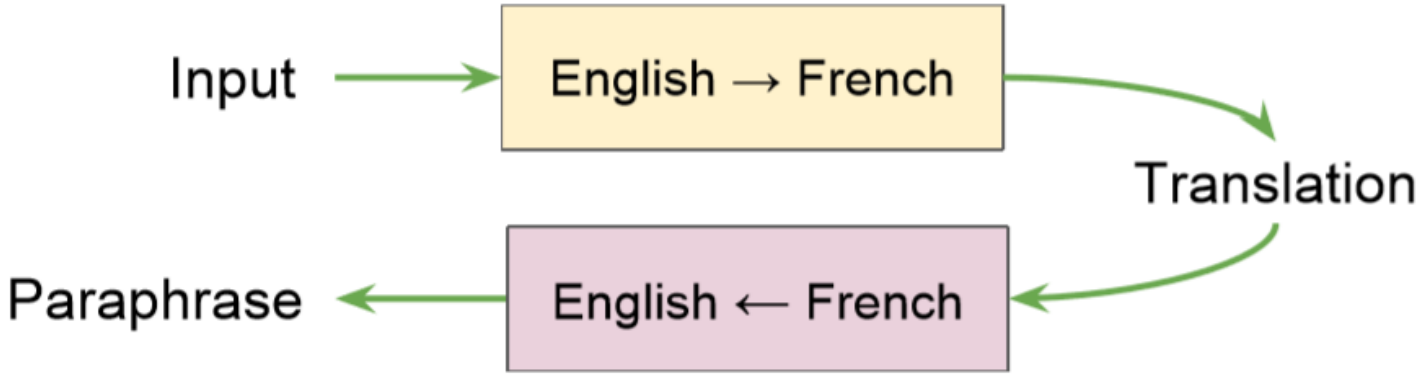
  Data Augmentation for Text

Data augmentation: popular in vision, but not NLP



More data with NMT back-translation

Previously, tea had been used primarily for Buddhist monks to stay awake during meditation.



QANet augmentation

Any other methods except for back-translation for data augmentation?

Question

What individuals live at Fatima House at Notre Dame?

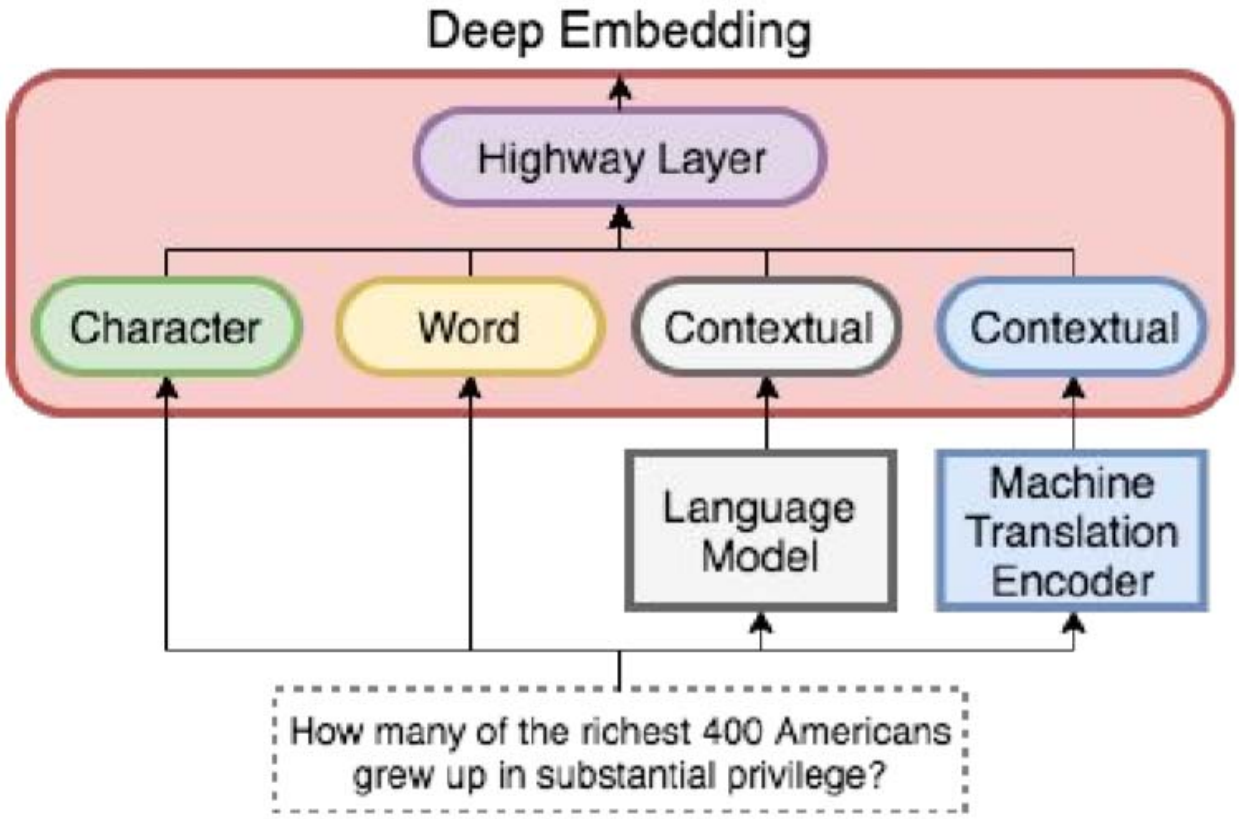
	Context	Answer
Original	… **Retired priests and brothers** reside in Fatima House (a former retreat center), Holy Cross House, as well as Columba Hall near the Grotto. ...	Retired priests and brothers
en-**de**-en	… In the Fatima House (a former retreat), the **tired priests and the brothers** are found in the magnificent church of Saint Cross, the Columbian Hall in the vicinity of the cave. ...	tired priests and the brothers

- **Idea #3:**

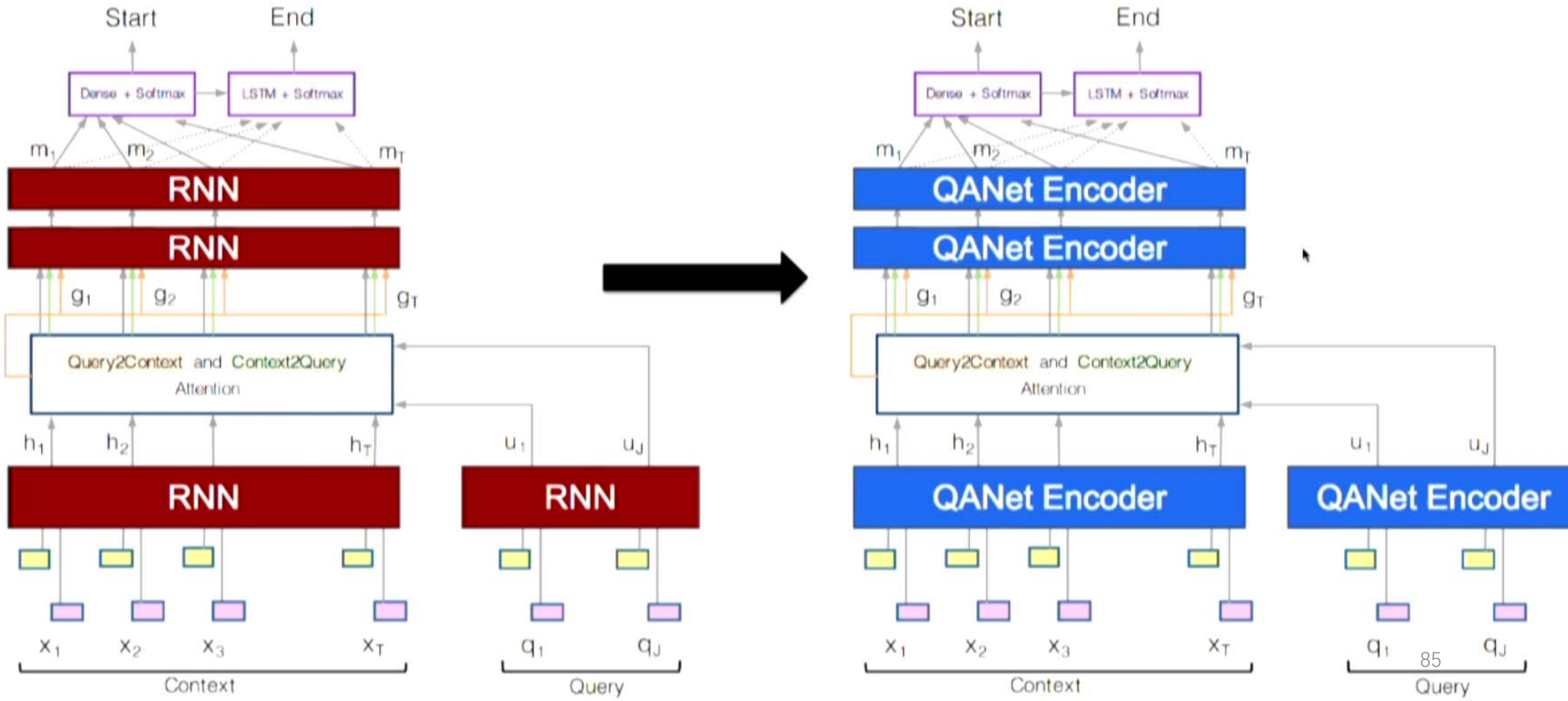
  Deep Embedding through Transfer Learning

Transfer learning for richer presentation

- Pretrained language model (ELMo, [Peters et al., NAACL'18])
  - + 4.0 F1
- Pretrained machine translation model (CoVe [McCann, NIPS'17])
  - + 0.3 F1



Base Model (*BiDAF*) → QANet



QANet – *3 key ideas*

- Deep Architecture without RNN
  - 130-layer (Deepest in NLP)
- Transfer Learning
  - leverage unlabeled data
- Data Augmentation
  - with back-translation

#1 on SQuAD (Mar-Aug 2018)

SQuAD1.1 Leaderboard

Since the release of SQuAD1.0, the community has made rapid progress, with the best models now rivaling human performance on the task. Here are the ExactMatch (EM) and F1 scores evaluated on the test set of v1.1.

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar et al. '16)	82.304	91.221
1 Sep 26, 2018	nlnet (ensemble) *Microsoft Research Asia*	**85.954**	**91.677**
2 Jul 11, 2018	QANet (ensemble) *Google Brain & CMU*	84.454	90.490
3 Jul 08, 2018	r-net (ensemble) *Microsoft Research Asia*	84.003	90.147
4 Sep 09, 2018	nlnet (single model) *Microsoft Research Asia*	83.468	90.133
4 Jun 20, 2018	MARS (ensemble) *YUANFUDAO research NLP*	83.982	89.796
5 Mar 19, 2018	QANet (ensemble) *Google Brain & CMU*	83.877	89.737
6 Sep 01, 2018	MARS (single model) *YUANFUDAO research NLP*	83.185	89.547
7 Jun 20, 2018	QANet (single) *Google Brain & CMU*	82.471	89.306
7 May 09, 2018	MARS (single model) *YUANFUDAO research NLP*	82.587	88.880

Thanks !