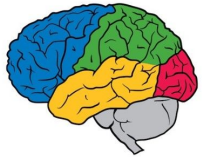


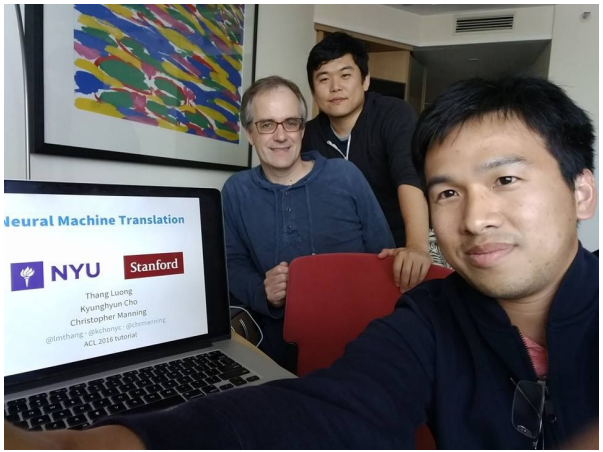
Transformer & Question Answering

Thang Luong

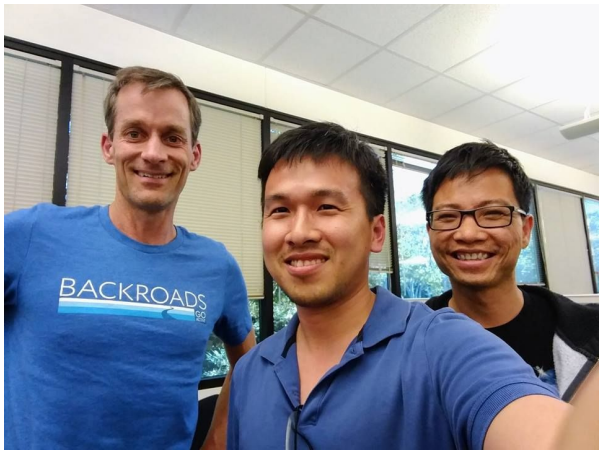
VietAI Guest Lecture

About me

- Research Scientist at Google Brain; PhD from Stanford



Chris Manning & Kyunghuyn Cho
(4-hr tutorial rehearsal at ACL'16)



Jeff Dean & Quoc Le
(Google office on July 4th, 2017)

Vietnamese @ Google Brain
Farewell Hung Bui
DeepMind → VinAI, March 2019

Why Transformer?

VietAI

Why Transformer?

VietAI

Best NLP Model Ever? Google BERT Sets New Standards in 11 Language Tasks

Synced in SyncedReview Follow
Oct 16, 2018 · 3 min read ★

Google's BERT

Google AI

Transformer

Why Transformer?

VietAI

Best NLP Model Ever? Google BERT Sets New Standards in 11 Language Tasks

Synced in SyncedReview Follow
Oct 16, 2018 · 3 min read ★

Transformer

- Google's BERT
- OpenAI's GPT2

Google AI

Outline

- Transformer
  - Self-attention
  - Architecture
- Question Answering (BiDAF, QANet)
- Beyond

Let’s translate “Apple” into your language

Apple is loved by many people; it is good for your **health**

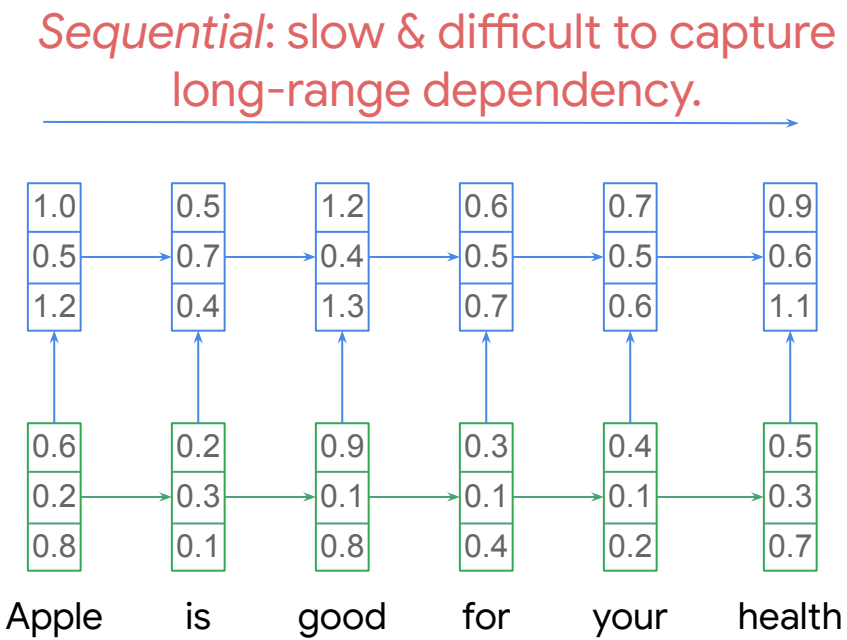
Apple is loved by many people; it is good in its **design**

Needs to capture long-range dependency

Apple is loved by many people; it is good for your **health**

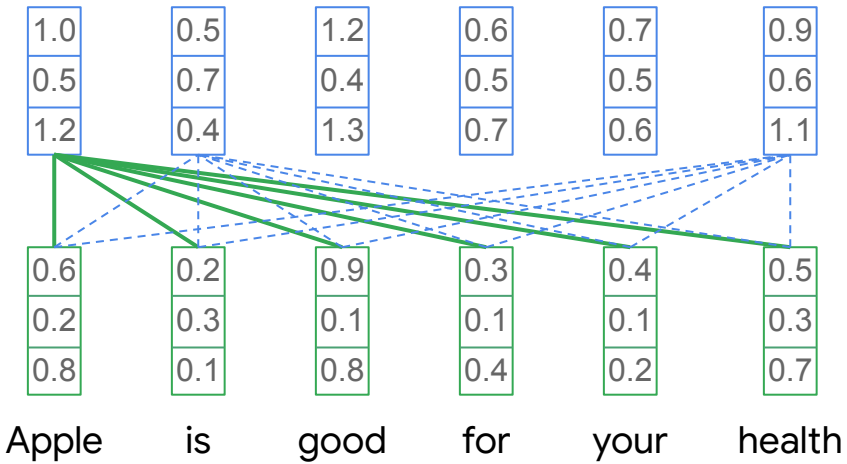
Apple is loved by many people; it is good in its **design**

RNNs/LSTMs


Sequential: slow & difficult to capture long-range dependency.
1.0 0.5 1.2
0.5 0.7 0.4
1.2 1.3 0.7
0.6 0.5 0.6
0.7 0.6 1.1
0.9
0.6 0.2 0.8
0.2 0.3 0.1
0.9 0.1 0.8
0.3 0.1 0.4
0.4 0.1 0.2
0.5 0.3 0.7
Apple
is
good
for
your
health
'Apple' is not aware of 'health'

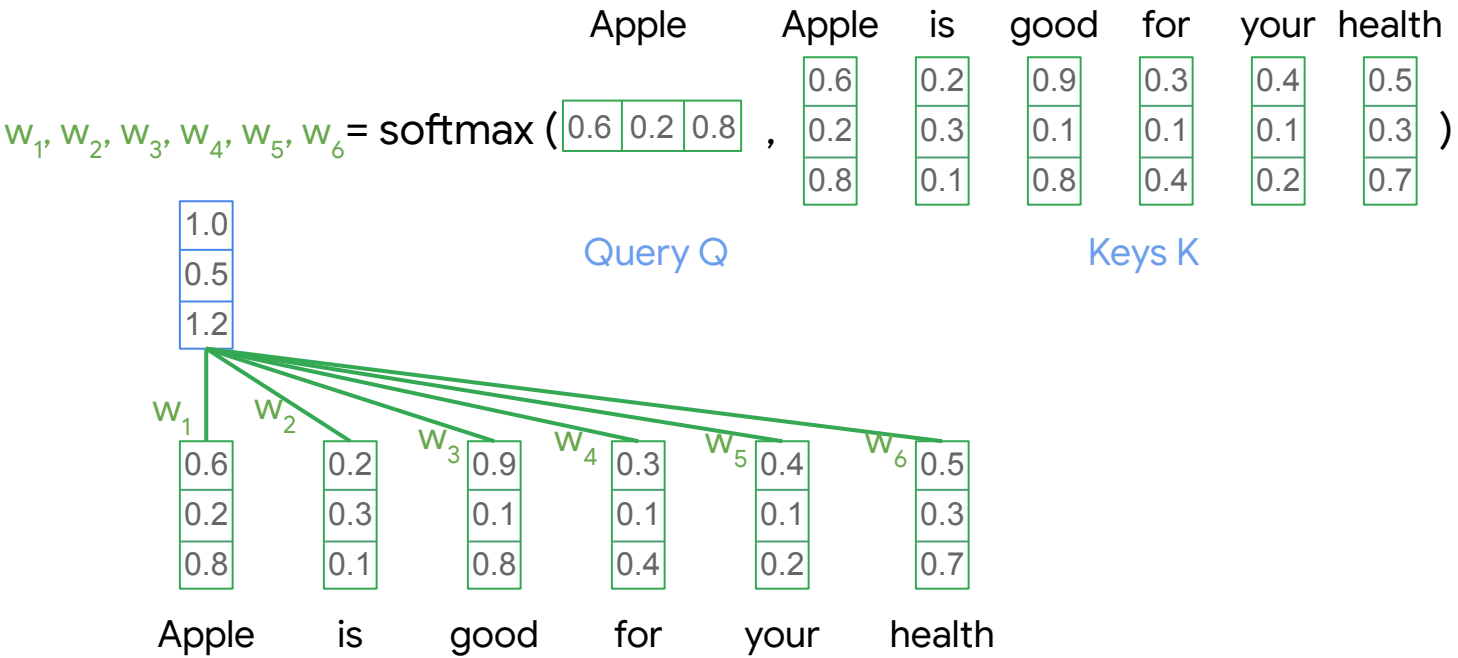
Self-attention

All words connect to one another

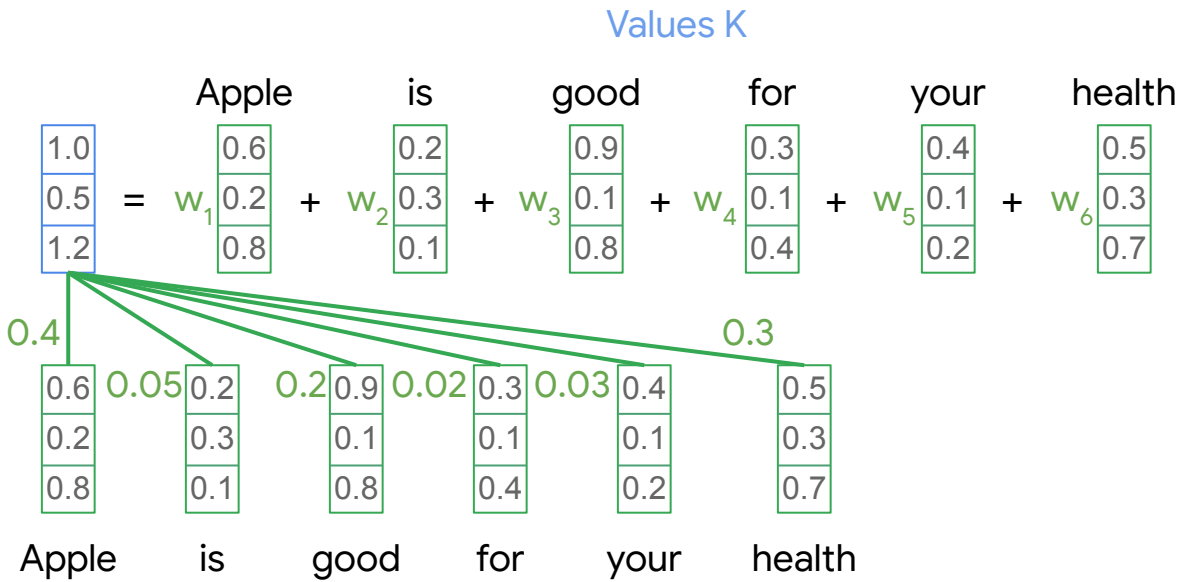


'Apple' is aware of 'health'

(Simplified) **Self-attention**



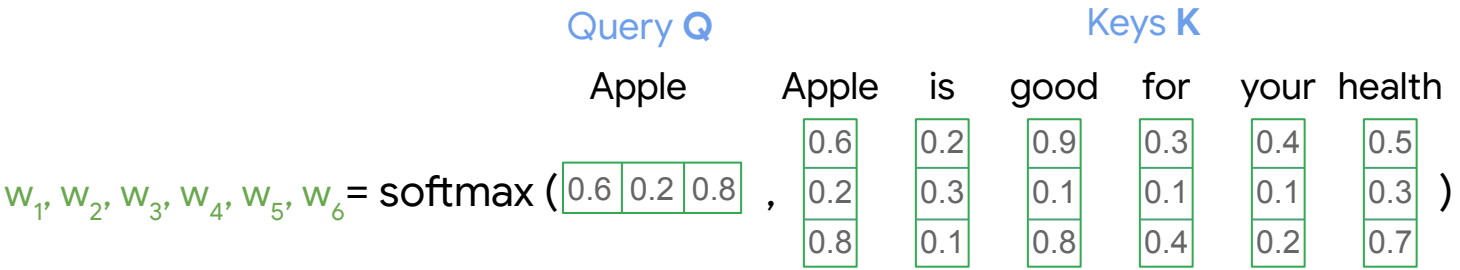
(Simplified) **Self-attention**

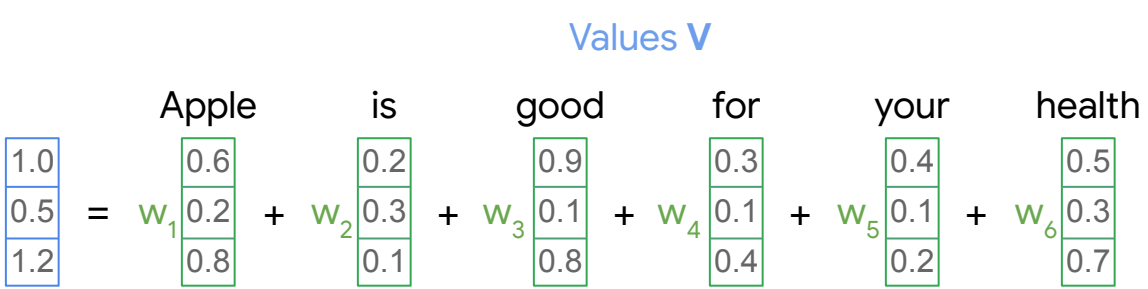


Self-attention

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

Scaled factor = sqrt(hidden size)





Multi-headed Self-attention

VietAI

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

One parameter set per head i

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Query **Q** | Keys **K**

	Apple	Apple	is	good	for	your	health
$w_1, w_2, w_3, w_4, w_5, w_6$ = softmax (	0.6	0.6	0.2	0.9	0.3	0.4	0.5
	0.2	0.2	0.3	0.1	0.1	0.1	0.3
	0.8	0.8	0.1	0.8	0.4	0.2	0.7

Values **V**

		Apple		is		good		for		your		health
1.0	=	w_1 0.6	+	w_2 0.2	+	w_3 0.9	+	w_4 0.3	+	w_5 0.4	+	w_6 0.5
0.5		0.2		0.3		0.1		0.1		0.1		0.3
1.2		0.8		0.1		0.8		0.4		0.2		0.7

Tensor2Tensor

VietAI

tensorflow / tensor2tensor

Watch 427 | Star 8,142 | Fork 2,081

Tensor2Tensor

pypi package 1.13.4 | issues 406 open | contributions welcome | chat on gitter | License Apache 2.0 | build error | Run on FloydHub

Tensor2Tensor, or T2T for short, is a library of deep learning models and datasets designed to make deep learning more accessible and accelerate ML research. T2T is actively used and maintained by researchers and engineers within the Google Brain team and a community of users. We're eager to collaborate with you too, so feel free to open an issue on GitHub or send along a pull request (see our contribution doc). You can chat with us on Gitter and join the T2T Google Group.

Quick Start

This iPython notebook explains T2T and runs in your browser using a free VM from Google, no installation needed. Alternatively, here is a one-command version that installs T2T, downloads MNIST, trains a model and evaluates it:

```
pip install tensor2tensor && t2t-trainer \
  --generate_data \
  --data_dir=~/t2t_data \
  --output_dir=~/t2t_train/mnist \
  --problem=image_mnist \
  --model=shake_shake \
  --hparams_set=shake_shake_quick \
  --train_steps=1000 \
  --eval_steps=100
```

https://github.com/tensorflow/tensor2tensor

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

https://arxiv.org/abs/1706.03762

Tensor2Tensor: *Multi-headed Self-attention*

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

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

```
def scaled_dot_product_attention_simple(q, k, v, bias, name=None):
  """Scaled dot-product attention. One head. One spatial dimension.

  Args:
    q: a Tensor with shape [batch, length_q, depth_k]
    k: a Tensor with shape [batch, length_kv, depth_k]
    v: a Tensor with shape [batch, length_kv, depth_v]
    bias: optional Tensor broadcastable to [batch, length_q, length_kv]
    name: an optional string
```

https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/layers/common_attention.py#L4390
https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/layers/common_attention.py#L5985

Tensor2Tensor: *Multi-headed Self-attention*

VietAI

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

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

```
scalar = tf.rsqrt(tf.to_float(common_layers.shape_list(q)[2]))
logits = tf.matmul(q * scalar, k, transpose_b=True)
if bias is not None:
  logits += bias
weights = tf.nn.softmax(logits, name="attention_weights")
if common_layers.should_generate_summaries():
  tf.summary.image(
      "attention", tf.expand_dims(tf.pow(weights, 0.2), 3), max_outputs=1)
return tf.matmul(weights, v)
```

https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/layers/common_attention.py#L4390
https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/layers/common_attention.py#L5985

Tensor2Tensor: *Multi-headed Self-attention*

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

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

```
scalar = tf.rsqrt(tf.to_float(common_layers.shape_list(q)[2]))
logits = tf.matmul(q * scalar, k, transpose_b=True)
if bias is not None:
  logits += bias
weights = tf.nn.softmax(logits, name="attention_weights")
if common_layers.should_generate_summaries():
  tf.summary.image(
      "attention", tf.expand_dims(tf.pow(weights, 0.2), 3), max_outputs=1)
return tf.matmul(weights, v)
```

https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/layers/common_attention.py#L4390
https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/layers/common_attention.py#L5985

Tensor2Tensor: *Multi-headed Self-attention*

VietAI

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

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

```
scalar = tf.rsqrt(tf.to_float(common_layers.shape_list(q)[2]))
logits = tf.matmul(q * scalar, k, transpose_b=True)
if bias is not None:
  logits += bias
weights = tf.nn.softmax(logits, name="attention_weights")
if common_layers.should_generate_summaries():
  tf.summary.image(
      "attention", tf.expand_dims(tf.pow(weights, 0.2), 3), max_outputs=1)
return tf.matmul(weights, v)
```

https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/layers/common_attention.py#L4390
https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/layers/common_attention.py#L5985

Tensor2Tensor: *Multi-headed Self-attention*

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

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

```
scalar = tf.rsqrt(tf.to_float(common_layers.shape_list(q)[2]))
logits = tf.matmul(q * scalar, k, transpose_b=True)
if bias is not None:
  logits += bias
weights = tf.nn.softmax(logits, name="attention_weights")
if common_layers.should_generate_summaries():
  tf.summary.image(
      "attention", tf.expand_dims(tf.pow(weights, 0.2), 3), max_outputs=1)
return tf.matmul(weights, v)
```

https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/layers/common_attention.py#L4390
https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/layers/common_attention.py#L5985

Tensor2Tensor: *Multi-headed Self-attention*

VietAI

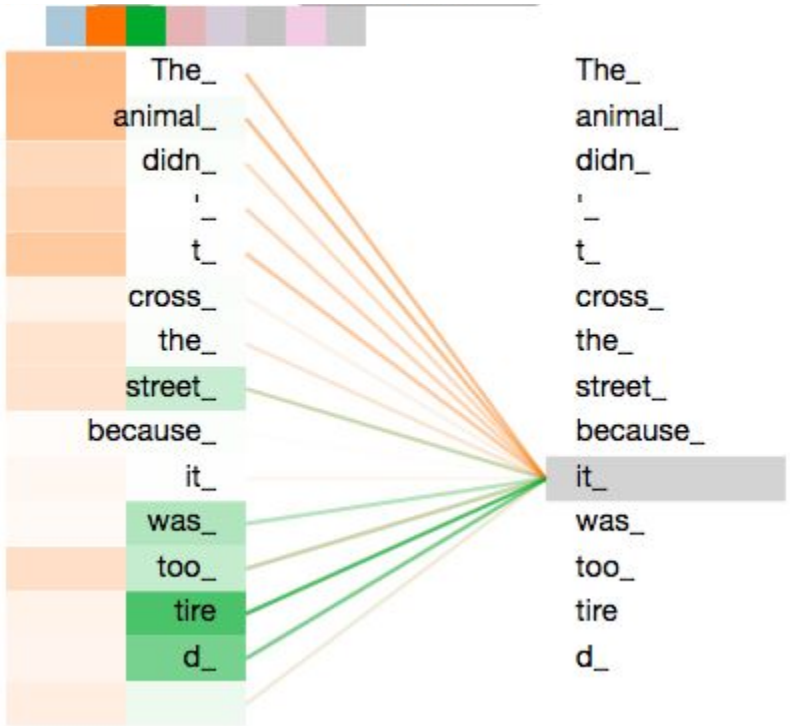
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

```
scalar = tf.rsqrt(tf.to_float(common_layers.shape_list(q)[2]))
logits = tf.matmul(q * scalar, k, transpose_b=True)
if bias is not None:
  logits += bias
weights = tf.nn.softmax(logits, name="attention_weights")
if common_layers.should_generate_summaries():
  tf.summary.image(
      "attention", tf.expand_dims(tf.pow(weights, 0.2), 3), max_outputs=1)
return tf.matmul(weights, v)
```

https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/layers/common_attention.py#L4390
https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/layers/common_attention.py#L5985

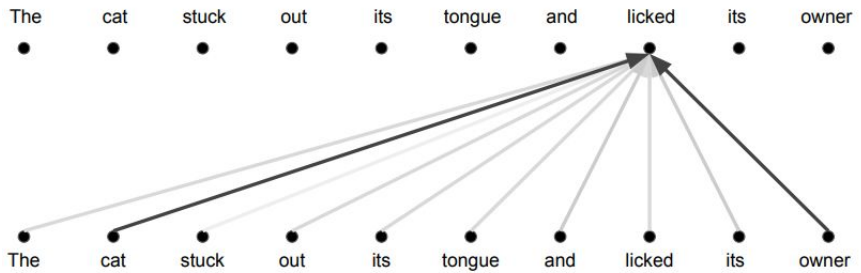
Visualization of multi-head attention



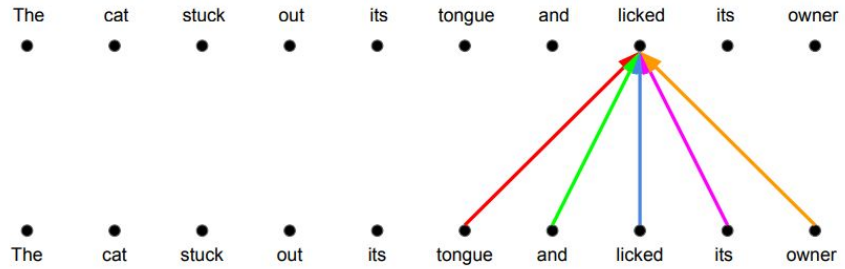
The Illustrated Transformer. https://jalammar.github.io/illustrated-transformer

Attention vs. Convolution

Attention: a weighted average



Convolution:
Different linear transformations by relative position.



Content from http://web.stanford.edu/class/cs224n/slides/cs224n-2019-lecture13-contextual-representations.pdf

Attention is cheap!

	FLOPs	
Self-Attention	$O(\text{length}^2 \cdot \text{dim})$	$= 4 \cdot 10^9$
RNN (LSTM)	$O(\text{length} \cdot \text{dim}^2)$	$= 16 \cdot 10^9$
Convolution	$O(\text{length} \cdot \text{dim}^2 \cdot \text{kernel\_width})$	$= 6 \cdot 10^9$

length=1000 dim=1000 kernel_width=3

Content from http://web.stanford.edu/class/cs224n/slides/cs224n-2019-lecture13-contextual-representations.pdf

Outline

- Transformer
  - Self-attention
  - Architecture
- Question Answering (BiDAF, QANet)
- Beyond

Self-attention as a layer

N vectors in → N vectors out

Transformer

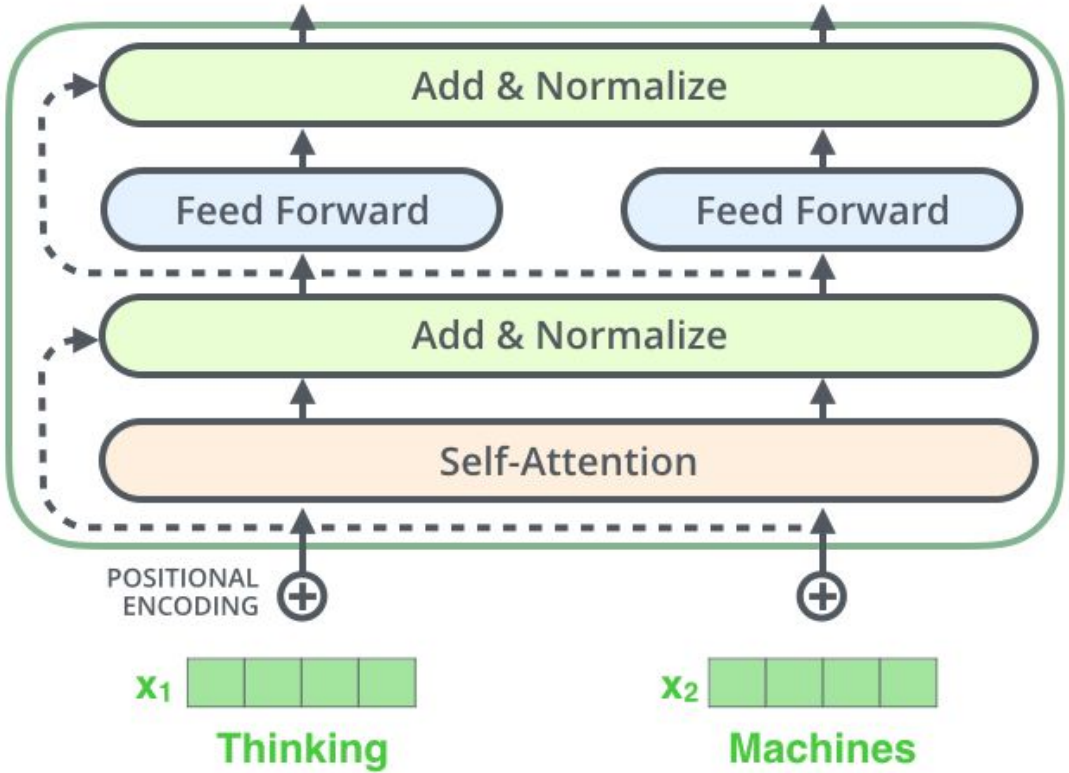


Image from https://jalammar.github.io/illustrated-transformer

Transformer

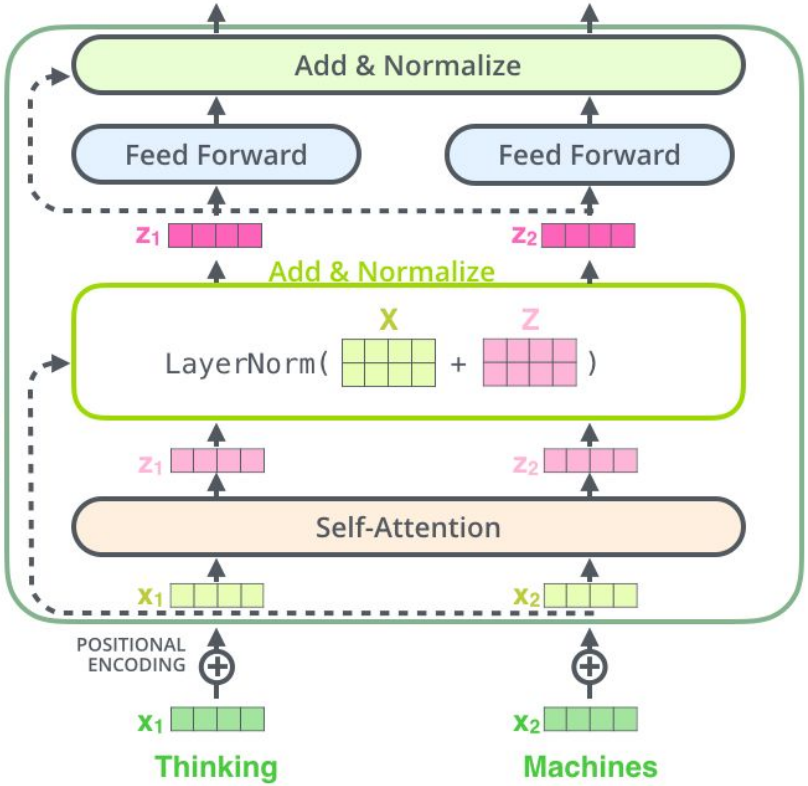
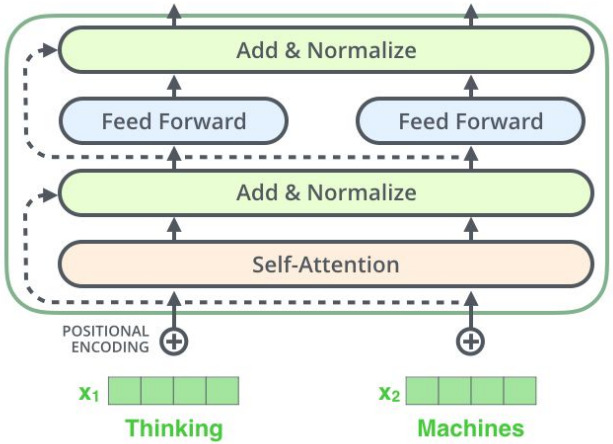


Image from https://jalammar.github.io/illustrated-transformer

Transformer

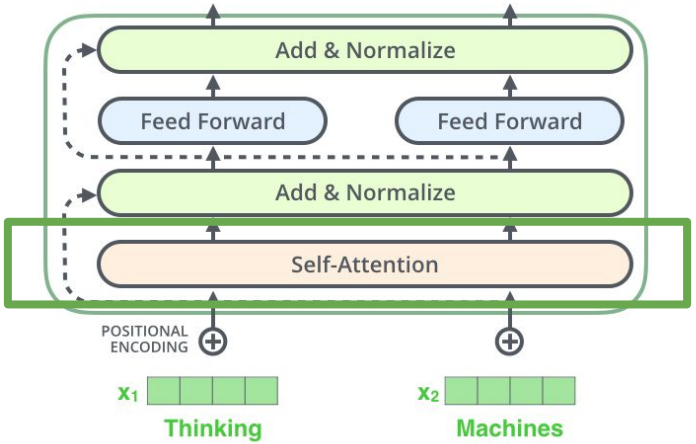
```
for layer in range(hparams.num_encoder_layers or hparams.num_hidden_layers):
  with tf.variable_scope("layer_%d" % layer):
    with tf.variable_scope("self_attention"):
      y = common_attention.multihead_attention(
          common_layers.layer_preprocess(x, hparams),
          training=(hparams.get("mode", tf.estimator.ModeKeys.TRAIN)
                    == tf.estimator.ModeKeys.TRAIN))
      x = common_layers.layer_postprocess(x, y, hparams)
    with tf.variable_scope("ffn"):
      y = transformer_ffn_layer(
          common_layers.layer_preprocess(x, hparams),
          hparams,
          pad_remover,
          conv_padding="SAME",
          nonpadding_mask=nonpadding,
          losses=losses)
      x = common_layers.layer_postprocess(x, y, hparams)
```



https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/transformer.py#L185
https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/layers/transformer_layers.py#L128

Transformer

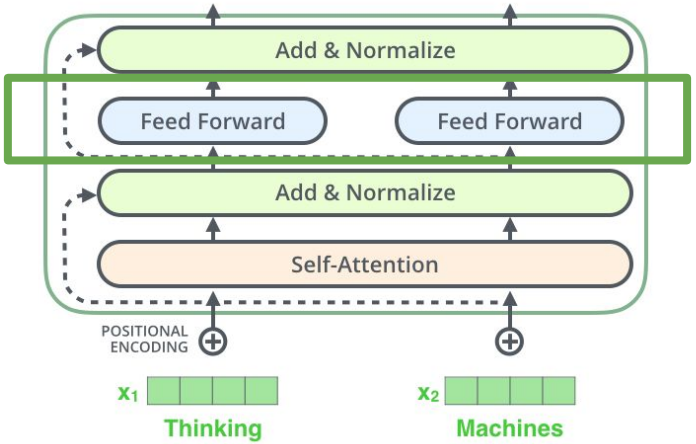
```
for layer in range(hparams.num_encoder_layers or hparams.num_hidden_layers):
  with tf.variable_scope("layer_%d" % layer):
    with tf.variable_scope("self_attention"):
      y = common_attention.multihead_attention(
          common_layers.layer_preprocess(x, hparams),
          training=(hparams.get("mode", tf.estimator.ModeKeys.TRAIN)
                    == tf.estimator.ModeKeys.TRAIN))
      x = common_layers.layer_postprocess(x, y, hparams)
    with tf.variable_scope("ffn"):
      y = transformer_ffn_layer(
          common_layers.layer_preprocess(x, hparams),
          hparams,
          pad_remover,
          conv_padding="SAME",
          nonpadding_mask=nonpadding,
          losses=losses)
      x = common_layers.layer_postprocess(x, y, hparams)
```



https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/transformer.py#L185
https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/layers/transformer_layers.py#L128

Transformer

```
for layer in range(hparams.num_encoder_layers or hparams.num_hidden_layers):
  with tf.variable_scope("layer_%d" % layer):
    with tf.variable_scope("self_attention"):
      y = common_attention.multihead_attention(
          common_layers.layer_preprocess(x, hparams),
          training=(hparams.get("mode", tf.estimator.ModeKeys.TRAIN)
                    == tf.estimator.ModeKeys.TRAIN))
      x = common_layers.layer_postprocess(x, y, hparams)
    with tf.variable_scope("ffn"):
      y = transformer_ffn_layer(
          common_layers.layer_preprocess(x, hparams),
          hparams,
          pad_remover,
          conv_padding="SAME",
          nonpadding_mask=nonpadding,
          losses=losses)
      x = common_layers.layer_postprocess(x, y, hparams)
```

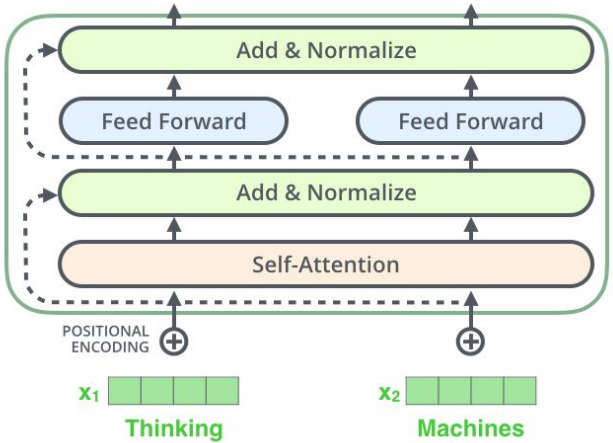


https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/transformer.py#L185
https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/layers/transformer_layers.py#L128

Transformer

```
for layer in range(hparams.num_encoder_layers or hparams.num_hidden_layers):
  with tf.variable_scope("layer_%d" % layer):
    with tf.variable_scope("self_attention"):
      y = common_attention.multihead_attention(
          common_layers.layer_preprocess(x, hparams),
          training=(hparams.get("mode", tf.estimator.ModeKeys.TRAIN)
                    == tf.estimator.ModeKeys.TRAIN))
      x = common_layers.layer_postprocess(x, y, hparams)
    with tf.variable_scope("ffn"):
      y = transformer_ffn_layer(
          common_layers.layer_preprocess(x, hparams),
          hparams,
          pad_remover,
          conv_padding="SAME",
          nonpadding_mask=nonpadding,
          losses=losses)
      x = common_layers.layer_postprocess(x, y, hparams)
```

Pre-process: x = LayerNorm(x)



Post-process: x = dropout(y) + x

https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/transformer.py#L185
https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/layers/transformer_layers.py#L128

Transformer for Translation

3 types of attentions:

- Self-attention Encoder
- (*Masked*) Self-attention Decoder
- Attention Decoder ⟶ Encoder

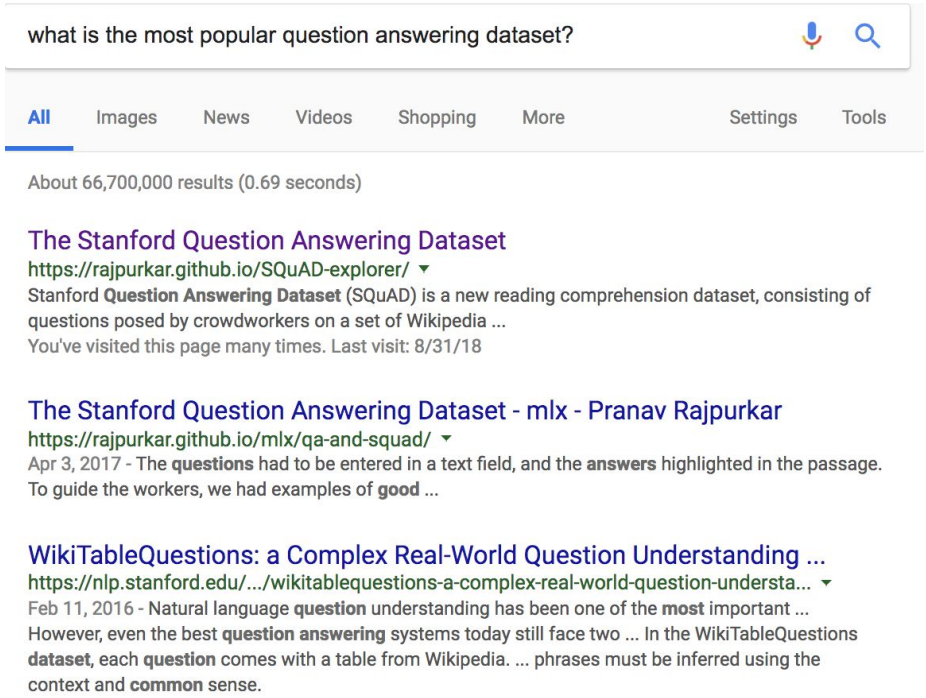
Parallelism:

- Training: Encoder & Decoder
- Inference: Encoder

Outline

- Transformer
- Question Answering (BiDAF, QANet)
- Beyond

Question answering is Google's core task!



Have to read to find answer

General QA systems

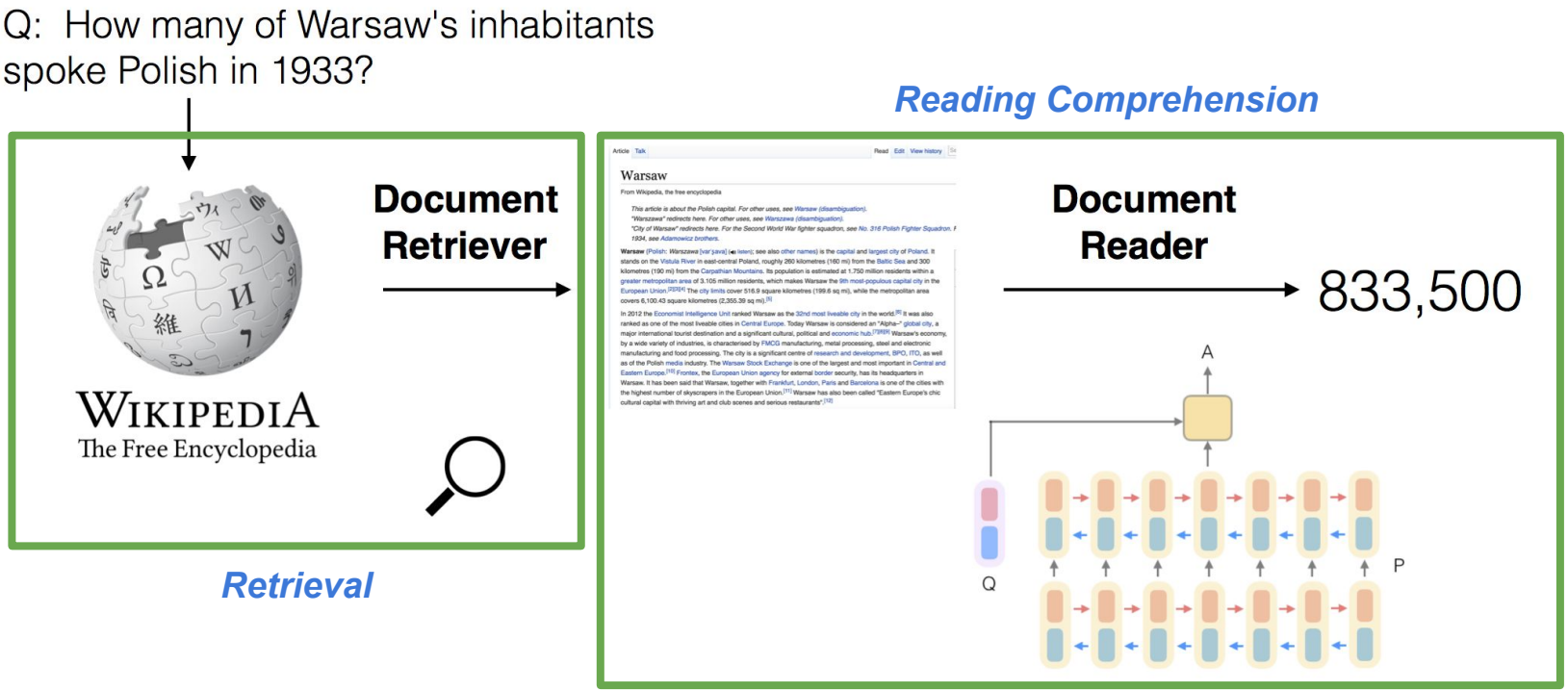
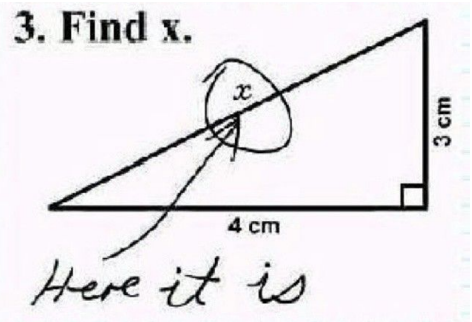
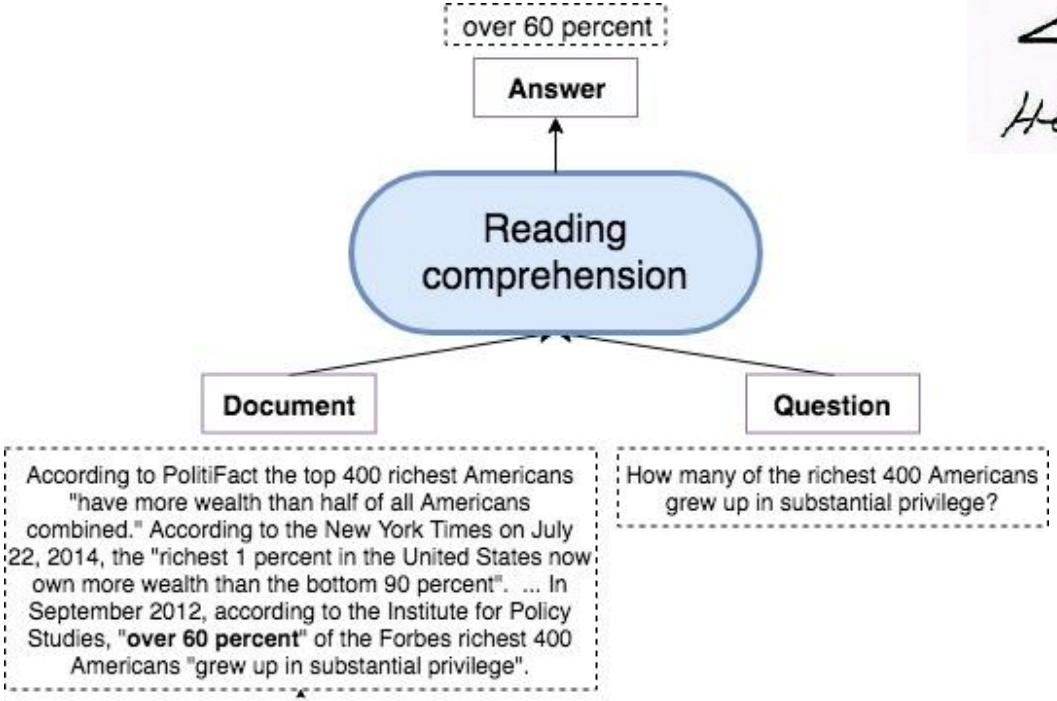


Image from https://github.com/facebookresearch/DrQA

Reading Comprehension

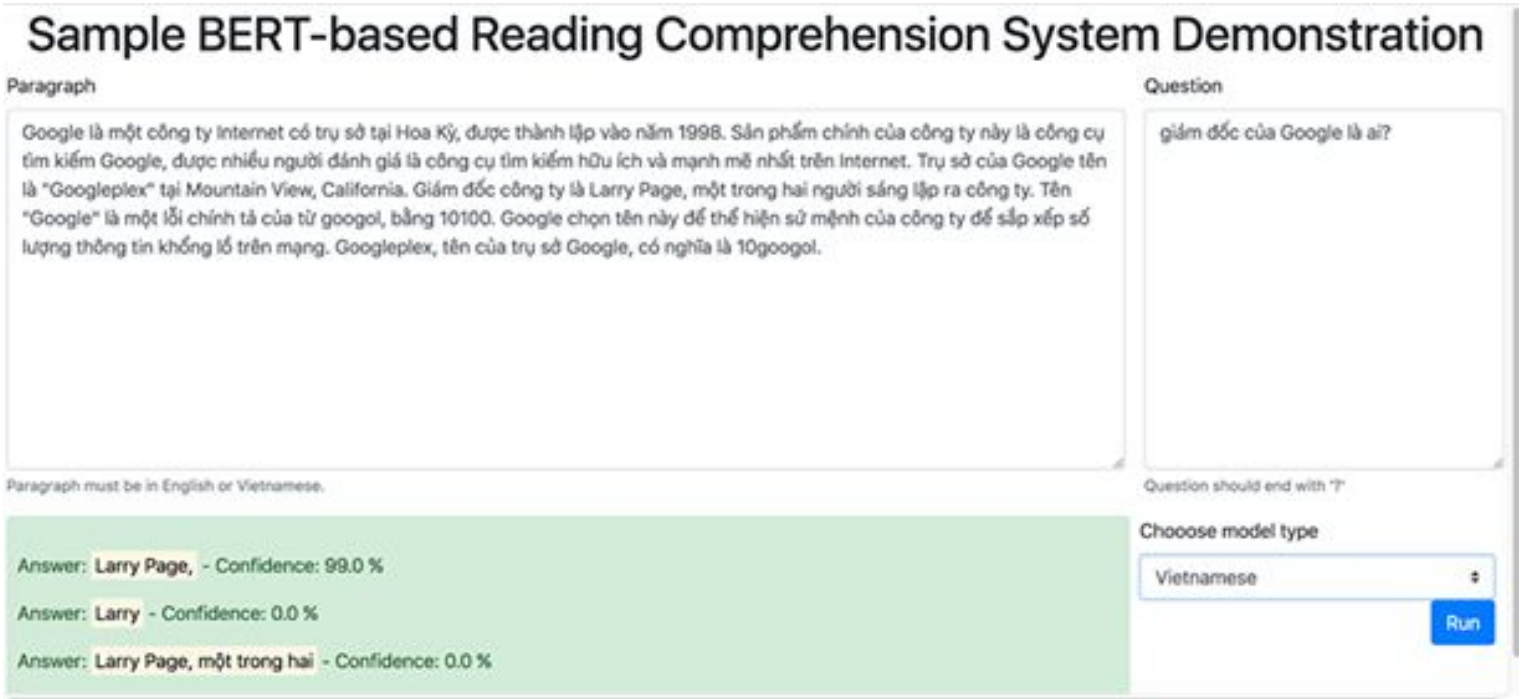




Stanford Question Answering Dataset (**SQuAD 1.1**) https://rajpurkar.github.io/SQuAD-explorer/

- **100K+ (question, document, answer)** over 500+ Wikipedia articles

Vietnamese Reading Comprehension System



Nguyen Viet Nam -- UIT -- https://github.com/phateopera/UITHelper_QAS

SQuAD: 100+ teams since Aug'16

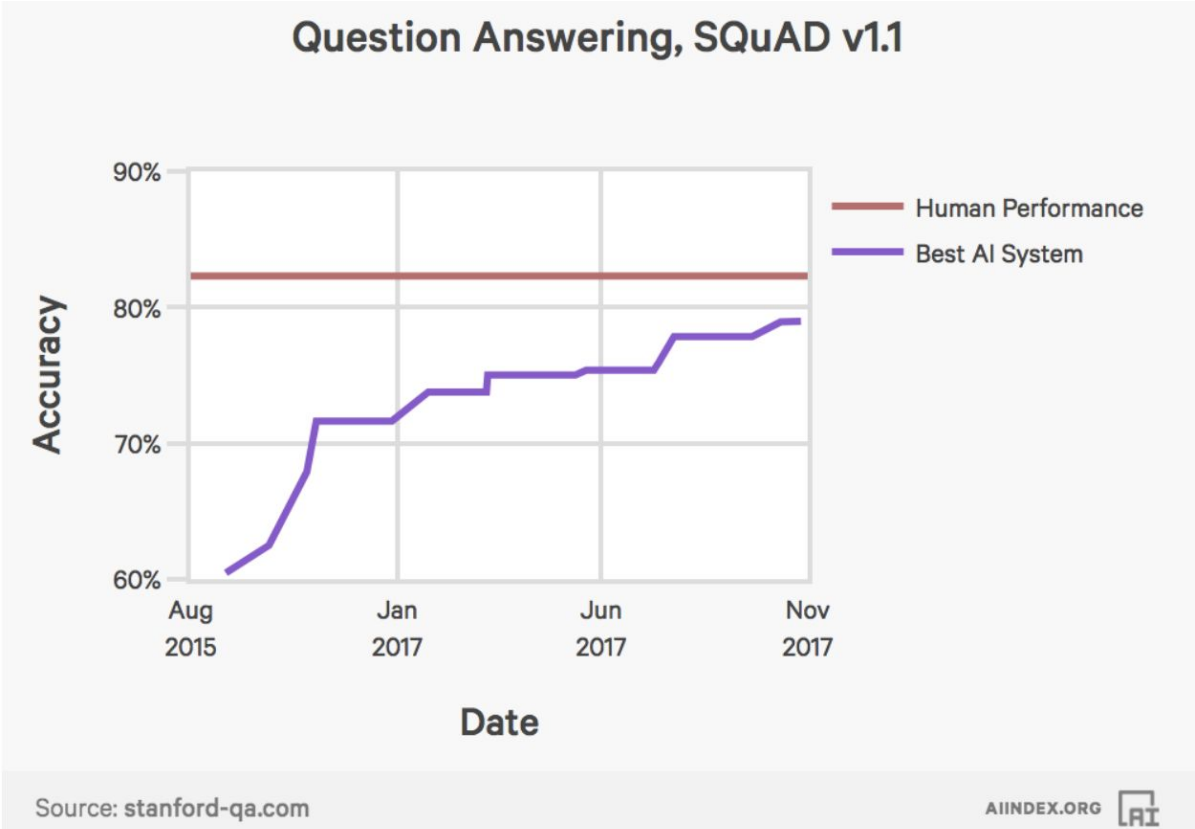
VietAI

Industry

Academia

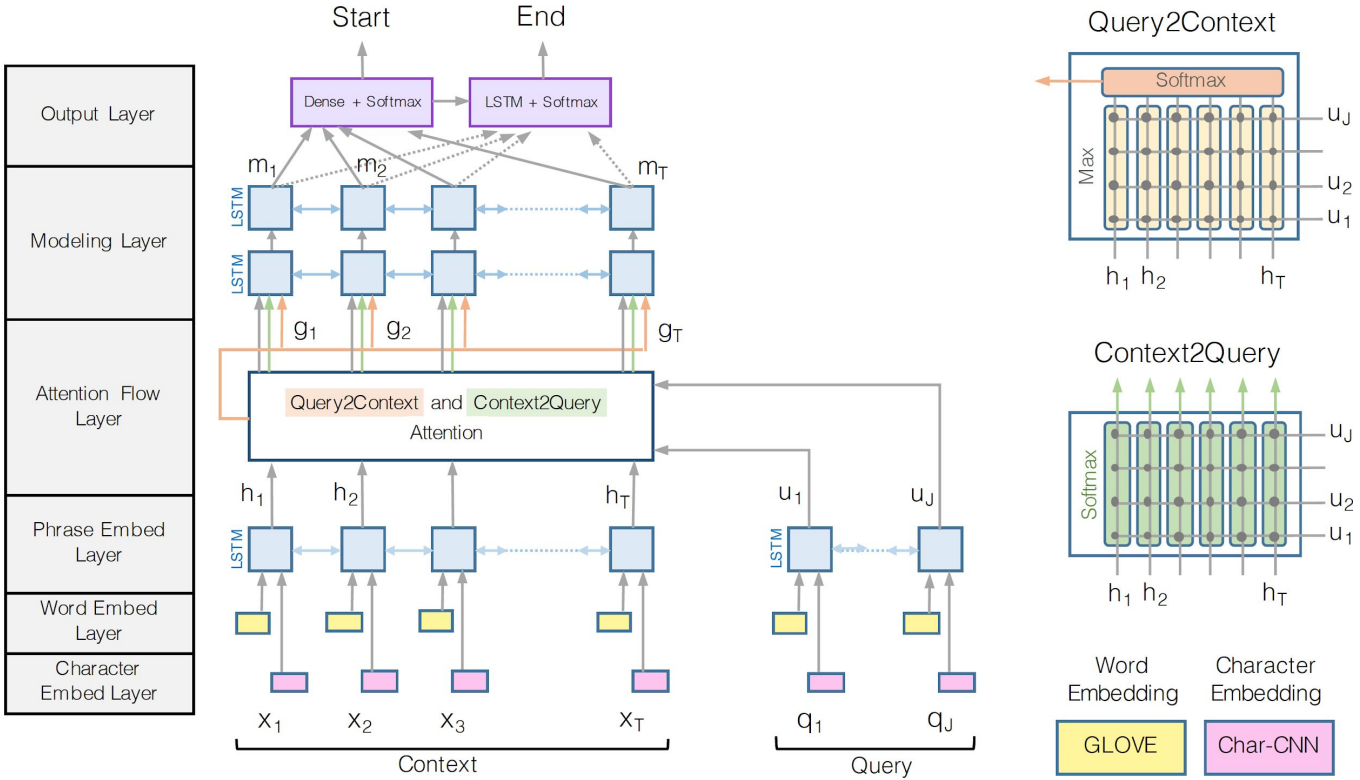


Until 2017


Question Answering, SQuAD v1.1
90%
80%
70%
60%
Accuracy
Human Performance
Best AI System
Aug 2015
Jan 2017
Jun 2017
Nov 2017
Date
Source: stanford-qa.com
AIINDEX.ORG

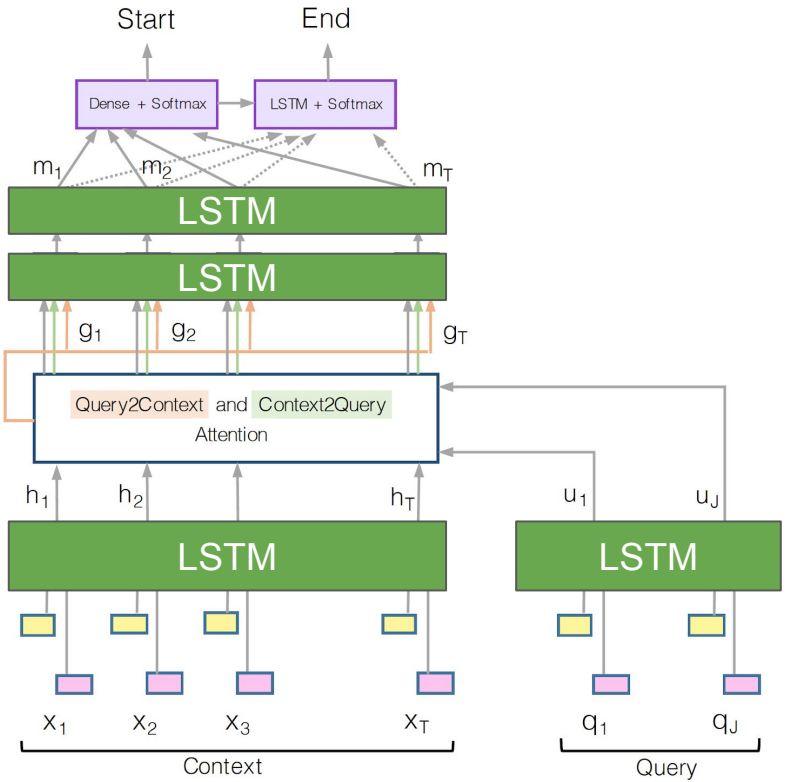
BiDAF -- LSTM

(Seo et al., ICLR'17)
https://allenai.github.io/bi-att-flow/



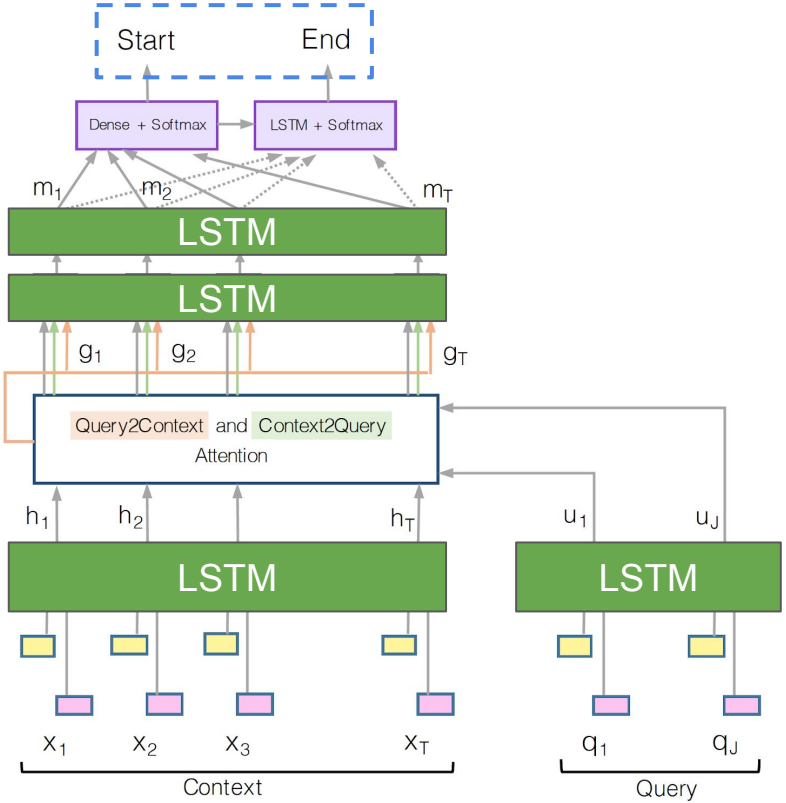
BiDAF -- LSTM

(Seo et al., ICLR'17)
https://allenai.github.io/bi-att-flow/



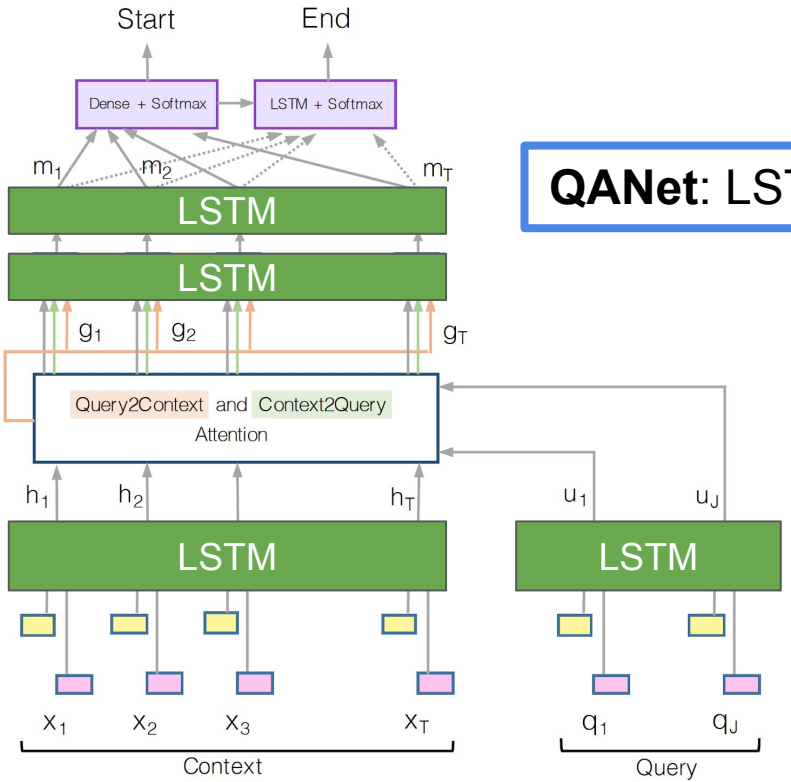
BiDAF -- LSTM

Predict start / end positions instead of words



BiDAF -- LSTM

(Seo et al., ICLR'17)
https://allenai.github.io/bi-att-flow/



QANet: LSTM is slow, let's replace it!

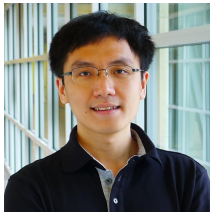
SQuAD1.1 Leaderboard

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 Jul 11, 2018	QANet (ensemble) *Google Brain & CMU*	**84.454**	**90.490**
2 Jul 08, 2018	r-net (ensemble) *Microsoft Research Asia*	84.003	90.147
3 Jun 20, 2018	MARS (ensemble) *YUANFUDAO research NLP*	83.982	89.796
4 Mar 19, 2018	QANet (ensemble) *Google Brain & CMU*	83.877	89.737
5 Jun 21, 2018	MARS (single model) *YUANFUDAO research NLP*	83.122	89.224
6 Mar 06, 2018	QANet (ensemble) *Google Brain & CMU*	82.744	89.045
7 May 09, 2018	MARS (single model) *YUANFUDAO research NLP*	82.587	88.880
7 Feb 19, 2018	Reinforced Mnemonic Reader + A2D (ensemble model) *Microsoft Research Asia & NUDT*	82.849	88.764
7 Jun 20, 2018	QANet (single) *Google Brain & CMU*	82.471	89.306
7 Jan 22, 2018	Hybrid AoA Reader (ensemble) *Joint Laboratory of HIT and iFLYTEK Research*	82.482	89.281
7 Jan 03, 2018	r-net+ (ensemble) *Microsoft Research Asia*	82.650	88.493
7 Jan 05, 2018	SLQA+ (ensemble) *Alibaba iDST NLP*	82.440	88.607

#1 Mar-Aug 2018

VietAI

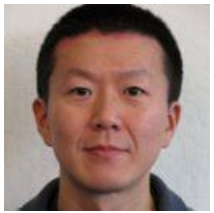


Adams Wei Yu

David Dohan

Thang Luong

Rui Zhao



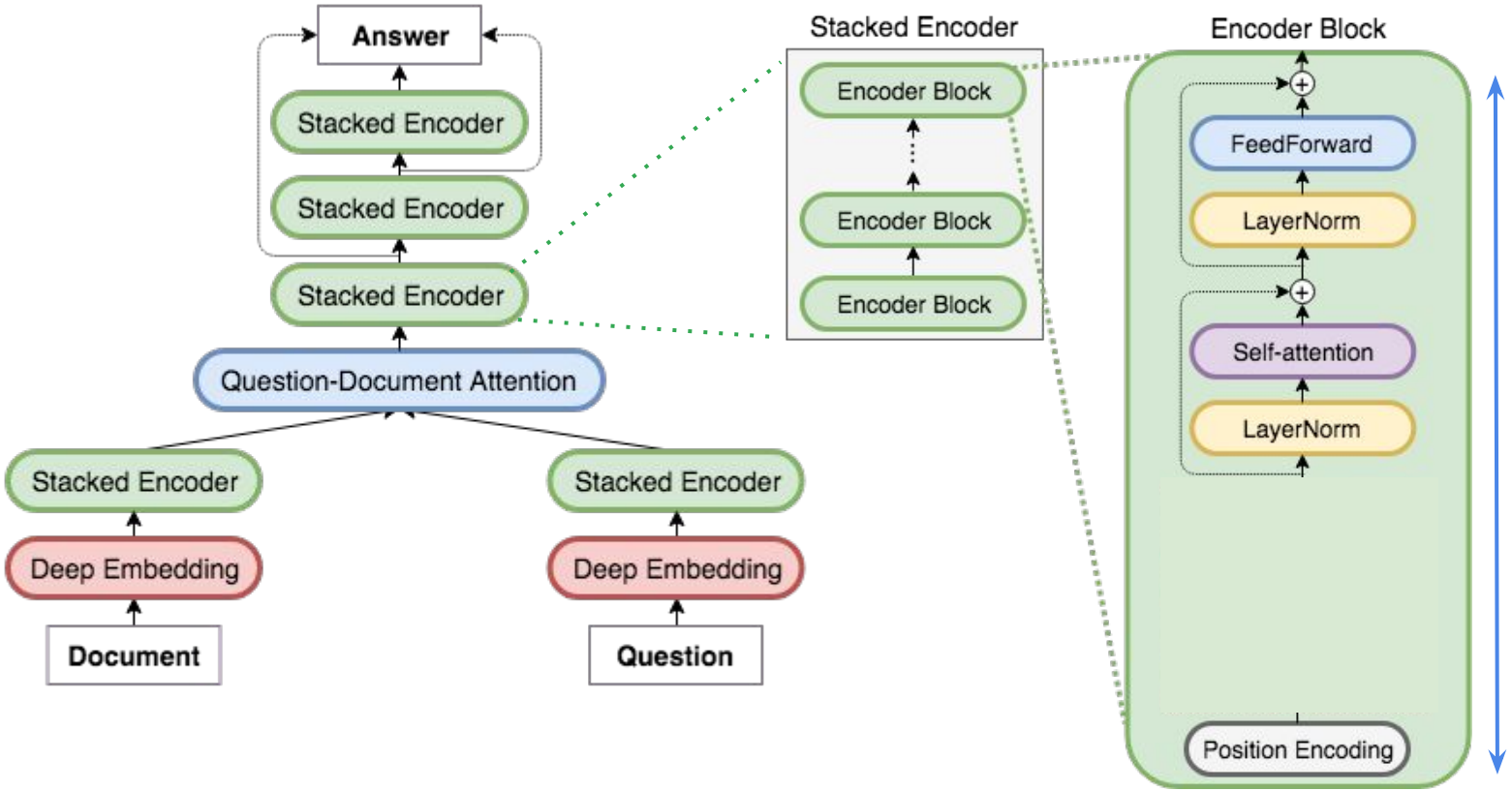
Kai Chen

Mohammad Norouzi

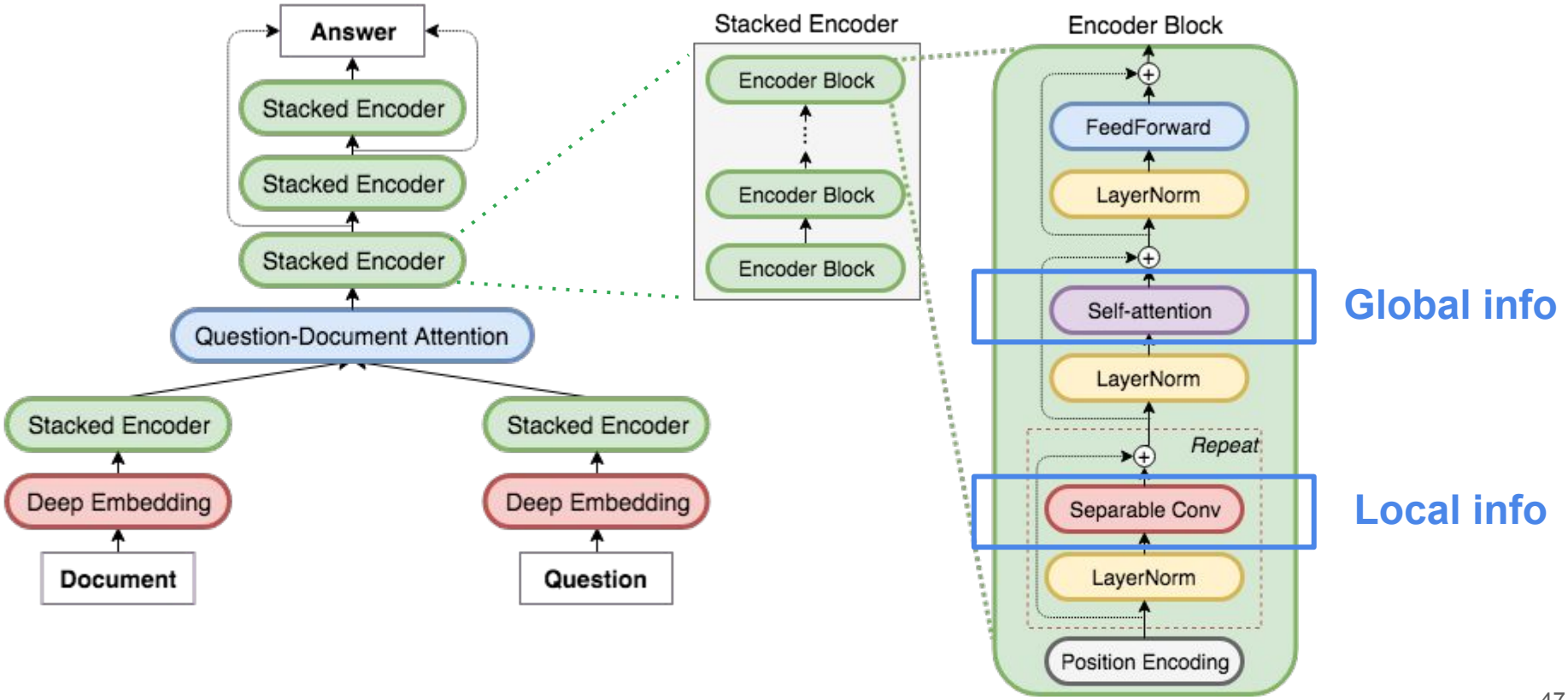
Quoc Le

Combining **Local Convolution** & **Global Self-Attention** for Reading Comprehension
ICLR'18, https://arxiv.org/abs/1804.09541

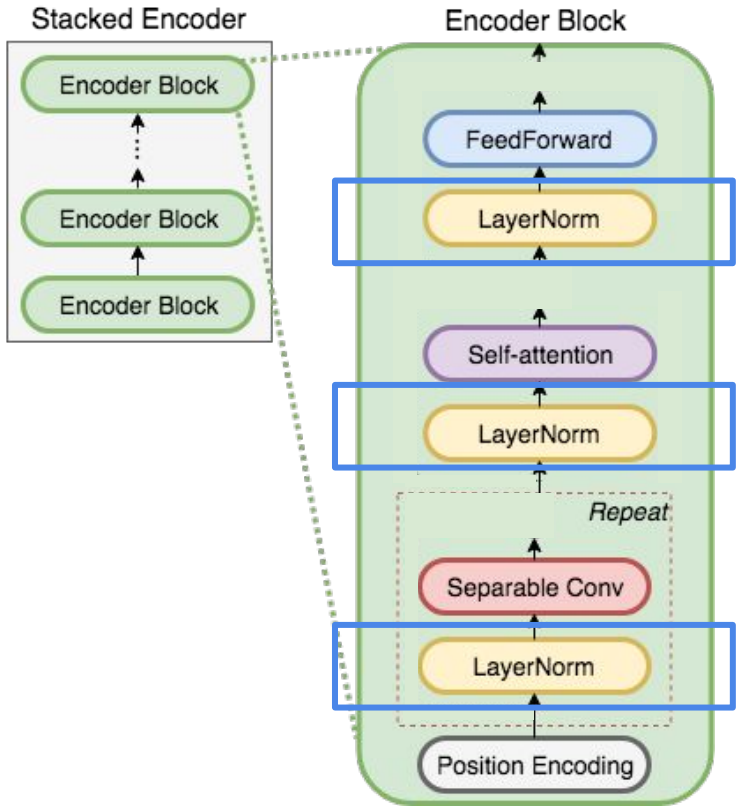
QANet - Inspired by Transformer



QANet - Inspired by Transformer



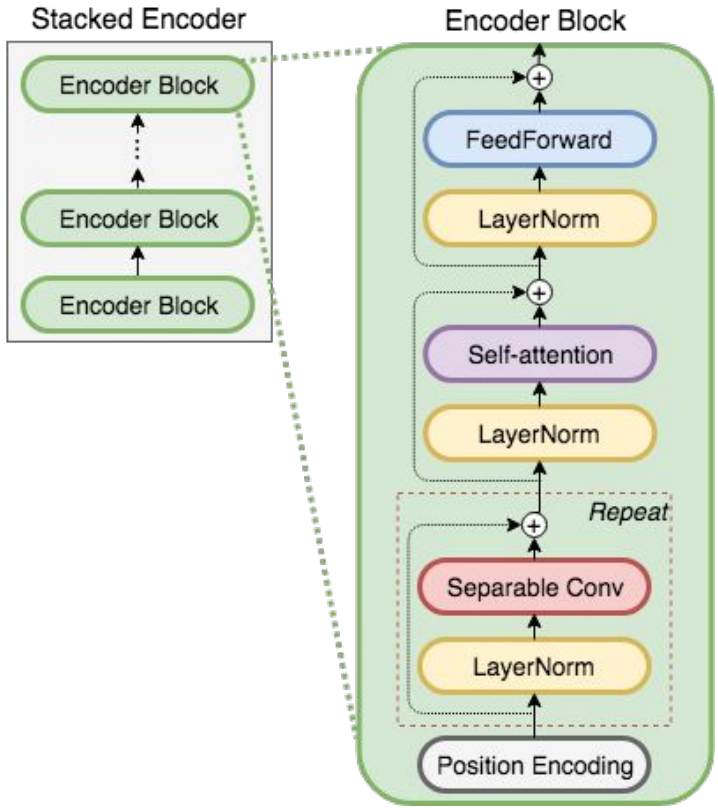
Lots of Regularizations



- **LayerNorm** everywhere

Layer Normalization (Ba et al., arXiv'16)
https://arxiv.org/pdf/1607.06450.pdf

Lots of Regularizations

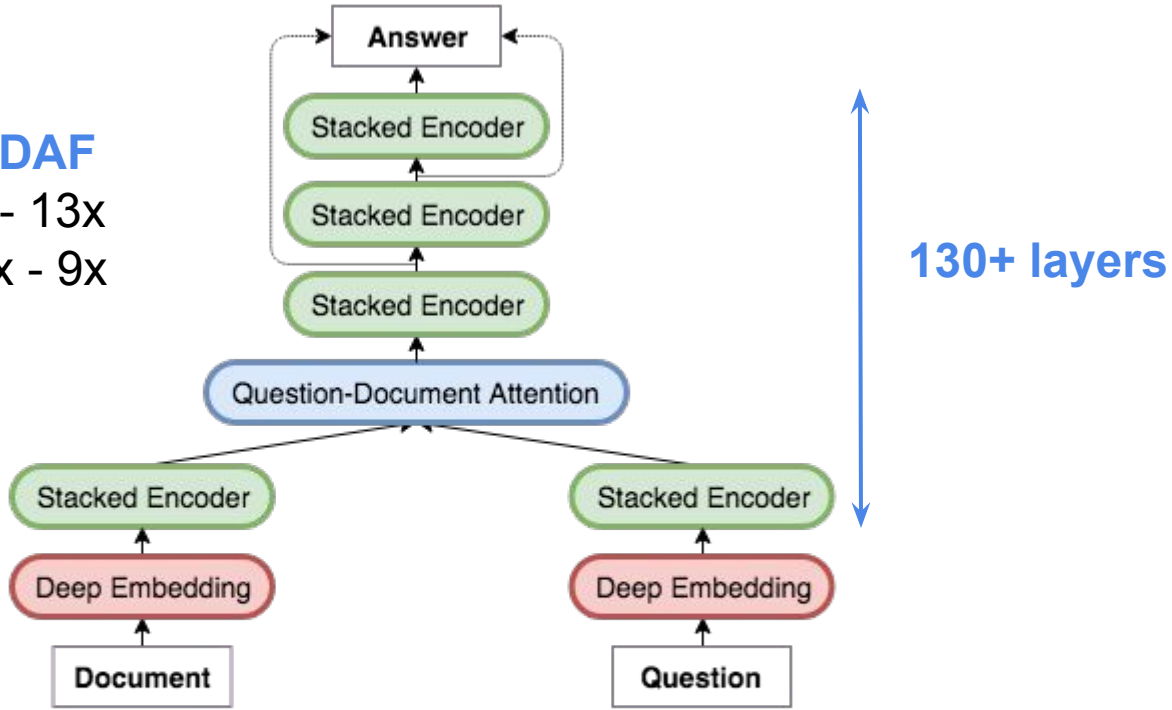


- **LayerNorm** everywhere
- **Residual Connections** everywhere

ResNet (He et al., CVPR'16)
https://arxiv.org/pdf/1512.03385.pdf

QANet: fast & accurate

- **Compared to BiDAF**
  - Training: 3x - 13x
  - Inference: 4x - 9x



Previously: 29 conv layers (Conneau et al., 2017) https://arxiv.org/abs/1606.01781

QANet was #1 (Mar-Aug 2018)

Jeff Dean 
@JeffDean

Following

You have questions? QANet, a new stacked architecture designed to do well on question answering tasks, has answers.

Thang Luong @lmthang
Happy to announce our QANet models, #1 on @stanfordnlp question answering dataset (SQuAD). 3 ideas: deep & fast arch (130+ layers), data augmentation, transfer learning. Joint work /w @AdamsYu @dmdohan @oahziur, Quoc Le, et al. See our …

Show this thread

6:09 PM - 25 Apr 2018

Outline

- Transformer
- Question Answering (BiDAF, QANet)
- Beyond

BERT: fully surpassing human (Oct 2018)

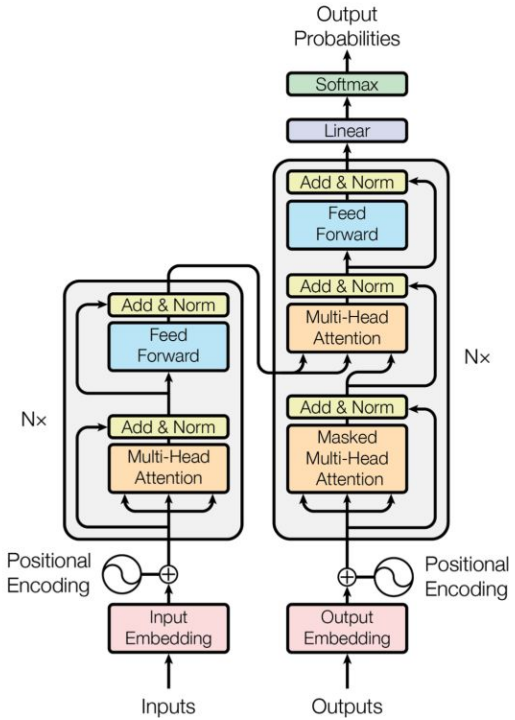
Thang Luong
@lmthang

A new era of NLP has just begun a few days ago: large pretraining models (Transformer 24 layers, 1024 dim, 16 heads) + massive compute is all you need. BERT from @GoogleAI: SOTA results on everything arxiv.org/abs/1810.04805. Results on SQuAD are just mind-blowing. Fun time ahead!

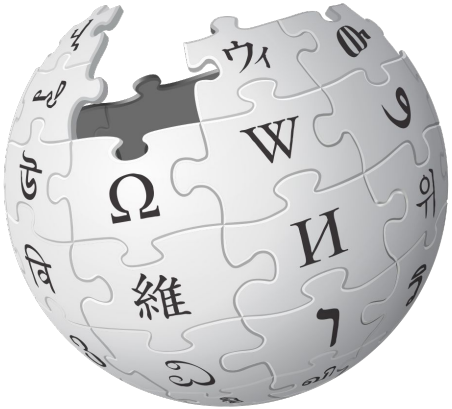
SQuAD1.1 Leaderboard

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 Oct 05, 2018	BERT (ensemble) *Google A.I.*	**87.433**	**93.160**



A successful transfer learning story



WIKIPEDIA
The Free Encyclopedia

The first recorded travels by Europeans to China and back date from this time. The most famous traveler of the period was the Venetian Marco Polo, whose account of his trip to "Cambaluc," the capital of the Great Khan, and of life there astounded the people of Europe. The account of his travels, Il milione (or, The Million, known in English as the Travels of Marco Polo), appeared about the year 1299. Some argue over the accuracy of Marco Polo's accounts due to the lack of mentioning the Great Wall of China, tea houses, which would have been a prominent sight since Europeans had yet to adopt a tea culture, as well the practice of foot binding by the women in capital of the Great Khan. Some suggest that Marco Polo acquired much of his knowledge through contact with Persian traders since many of the places he named were in Persian.

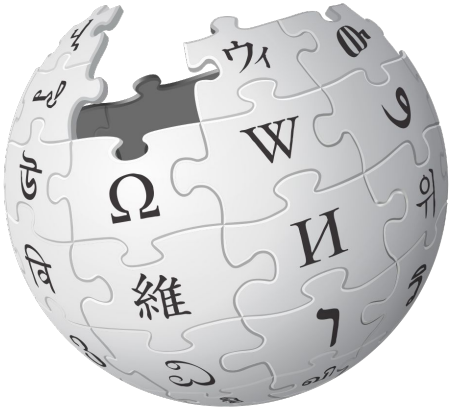
How did some suspect that Polo learned about China instead of by actually visiting it?

Answer: through contact with Persian traders

Pretraining: large, unlabeled data

Transfer Learning: small, labeled data

BERT → SQuAD



WIKIPEDIA
The Free Encyclopedia

The first recorded travels by Europeans to China and back date from this time. The most famous traveler of the period was the Venetian Marco Polo, whose account of his trip to "Cambaluc," the capital of the Great Khan, and of life there astounded the people of Europe. The account of his travels, Il milione (or, The Million, known in English as the Travels of Marco Polo), appeared about the year 1299. Some argue over the accuracy of Marco Polo's accounts due to the lack of mentioning the Great Wall of China, tea houses, which would have been a prominent sight since Europeans had yet to adopt a tea culture, as well the practice of foot binding by the women in capital of the Great Khan. Some suggest that Marco Polo acquired much of his knowledge through contact with Persian traders since many of the places he named were in Persian.

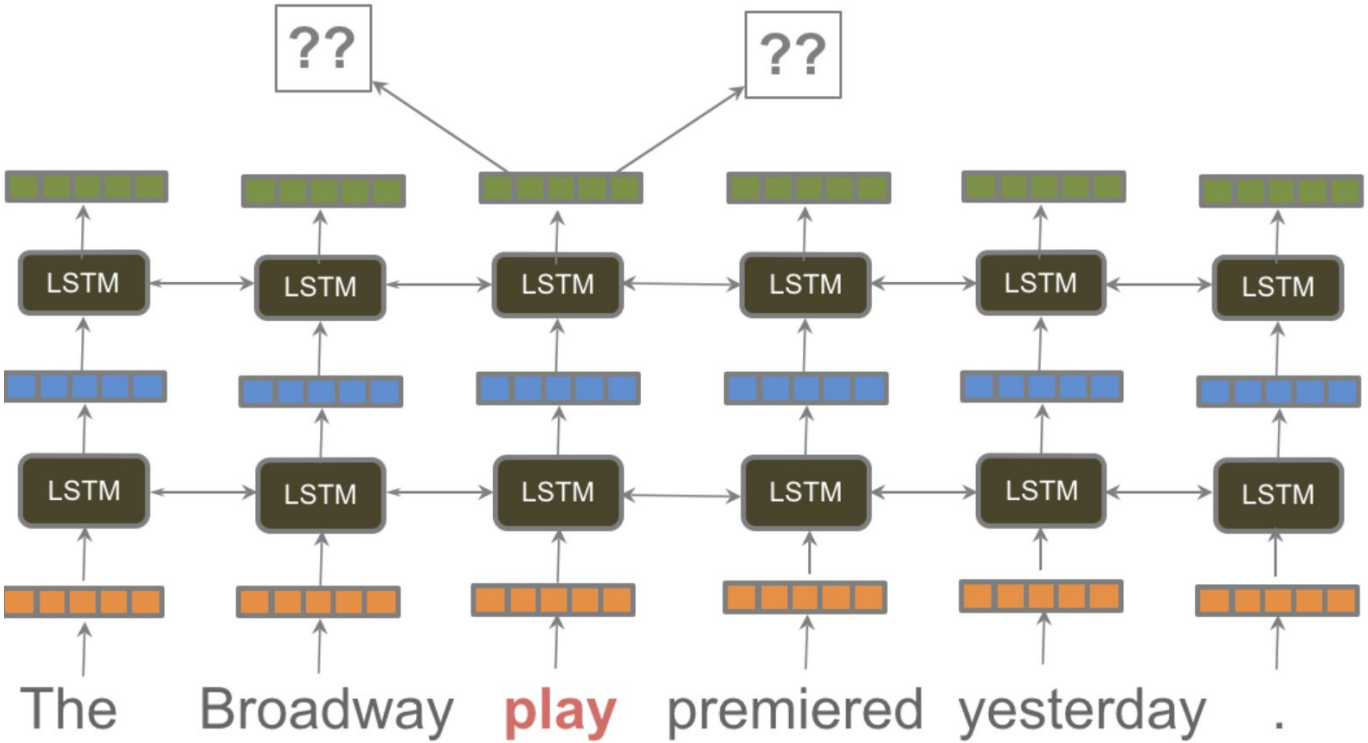
How did some suspect that Polo learned about China instead of by actually visiting it?

Answer: through contact with Persian traders

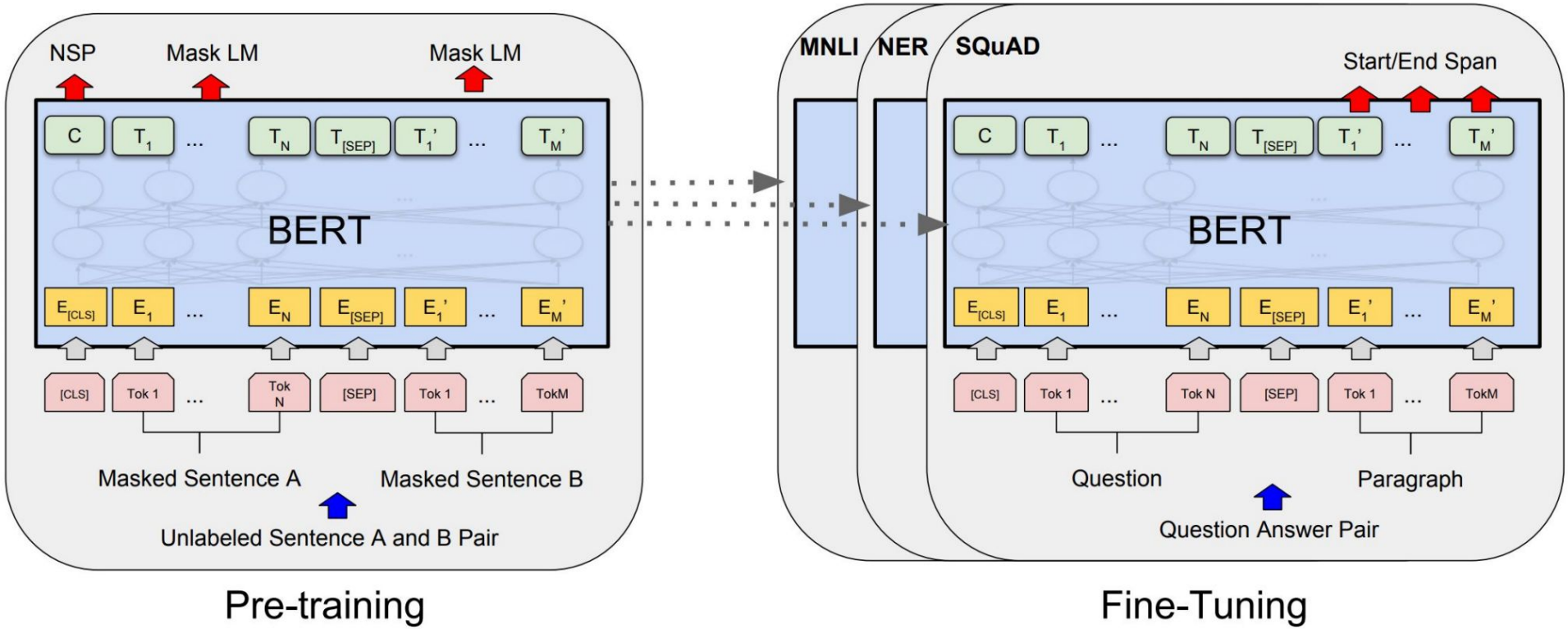
BERT pretrained on
English Wikipedia (~4M articles)

SQuAD: 500 articles
Stanford Question Answering Dataset

Unsupervised Learning for Text



BERT for SQuAD



Content from https://nlp.stanford.edu/seminar/details/jdevlin.pdf

BERT for SQuAD

1.1

What was another term used for the oil crisis?
Ground Truth Answers: first oil shock | shock | shock | first oil shock | shock
Prediction: shock

The 1973 oil crisis began in October 1973 when the members of the Organization of Arab Petroleum Exporting Countries (OAPEC, consisting of the Arab members of OPEC plus Egypt and Syria) proclaimed an oil embargo. By the end of the embargo in March 1974, the price of oil had risen from US$3 per barrel to nearly $12 globally; US prices were significantly higher. The embargo caused an oil crisis, or "shock", with many short- and long-term effects on global politics and the global economy. It was later called the "first oil shock", followed by the 1979 oil crisis, termed the "second oil shock."

2.0

What action did the US begin that started the second oil shock?
Ground Truth Answers: <No Answer>
Prediction: <No Answer>

The 1973 oil crisis began in October 1973 when the members of the Organization of Arab Petroleum Exporting Countries (OAPEC, consisting of the Arab members of OPEC plus Egypt and Syria) proclaimed an oil embargo. By the end of the embargo in March 1974, the price of oil had risen from US$3 per barrel to nearly $12 globally; US prices were significantly higher. The embargo caused an oil crisis, or "shock", with many short- and long-term effects on global politics and the global economy. It was later called the "first oil shock", followed by the 1979 oil crisis, termed the "second oil shock."

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar et al. '16)	82.304	91.221
1 Oct 05, 2018	BERT (ensemble) *Google AI Language* https://arxiv.org/abs/1810.04805	**87.433**	**93.160**
2 Oct 05, 2018	BERT (single model) *Google AI Language* https://arxiv.org/abs/1810.04805	85.083	91.835
2 Sep 26, 2018	nlnet (ensemble) *Microsoft Research Asia*	85.954	91.677
5 Sep 09, 2018	nlnet (single model) *Microsoft Research Asia*	83.468	90.133
3 Jul 11, 2018	QANet (ensemble) *Google Brain & CMU*	84.454	90.490

Content from https://nlp.stanford.edu/seminar/details/jdevlin.pdf

Final: Let's work on something for Vietnamese!

VietAI

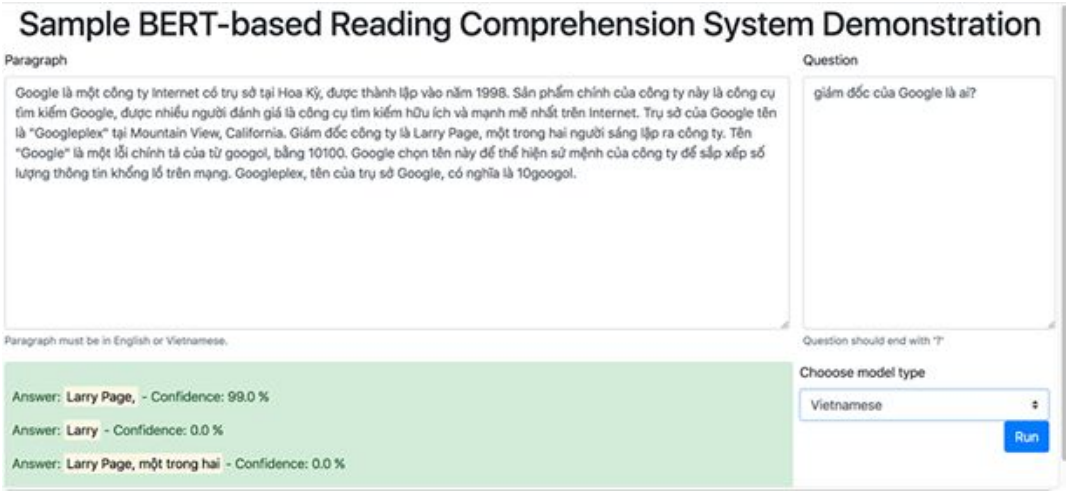
Binhvq News Corpus

Thông tin cơ bản:

- Trích xuất từ khoảng 14.896.998 bài báo trên internet bao gồm các báo:

```
2Sao, ANTG, ANTT, ANTĐ, ATGT, BVPL, BizLIVE, Biên Phòng, Bnews, Báo Văn hóa, E
Bộ GTVT, Bộ KHĐT, Bộ Ngoại Giao, Bộ Nội Vụ, Bộ Tài Chính, Bộ VHTTDL, CAND, CAE
Cartimes, Chính Phủ, Công Luận, Công Lý, Công Thương, DNVN, Doanh Nghiệp, Dân
Em Đẹp, GD&TĐ, GTVT, Gia Đình Mới, Gia Đình VN, Giao Thông, Giáo Dục VN, GĐ&XH
Hà Tĩnh, Hải Quan, ICTNews, Infonet, KTNT, KTĐT, Khỏe 365, Khỏe Plus, Khỏe Plu
Kiểm Sát, Kiểm sát, Kỷ Nguyên Số, Lao Động, LĐTĐ, MT&CS, Mặt Trận, Một Thế Gio
Nghe Nhìn VN, Nghệ An, Ngày Nay, Người Làm Báo, Người Tiêu Dùng, Người Đô Thị,
Nhân Dân, Nông Nghiệp, NĐ&ĐS, PC World, PL&XH, PLO, PNNews, PNSK, PetroTimes,
Pháp Luật Plus, Pháp Luật VN, Phụ Nữ VN, Quốc Hội, Quốc Hội TV, QĐND, SGGP, SG
Seatimes, Sài Gòn Tiếp Thị, TBDN, TBKTSG, TG&VN, TGTT, TH&PL, TNMT, TTOL, TTXV
Thanh Niên, Thanh Tra, TheLEADER, Thương Gia, Thế Giới Trẻ, Thế Giới Xe, Tin N
Tin Tức TTXVN, Tiền Phong, TuanVietNam, Tuyên Giáo, Tuổi Trẻ TĐ, Tài Chính, Tạ
Tạp chí Xây dựng Đảng, Tạp chí cộng sản, Tổ Quốc, VEF, VNCA, VNEWS, VOV, VTC,
Vietnam Finance, VietnamNet, VietnamPlus, VnEconomy, VnMedia, Văn Hiến, Văn Hó
Xây Dựng Đảng, Zing, Ôtô – xe máy, Ôtô Xe Máy, ĐCSVN, ĐS&PL, ĐTCK, Đại Đoàn Kế
Đất Việt, Đấu Thầu, Đầu Tư, Đời Sống Plus
```

https://github.com/binhvq/news-corpus



Thank you!

External Slides for BERT

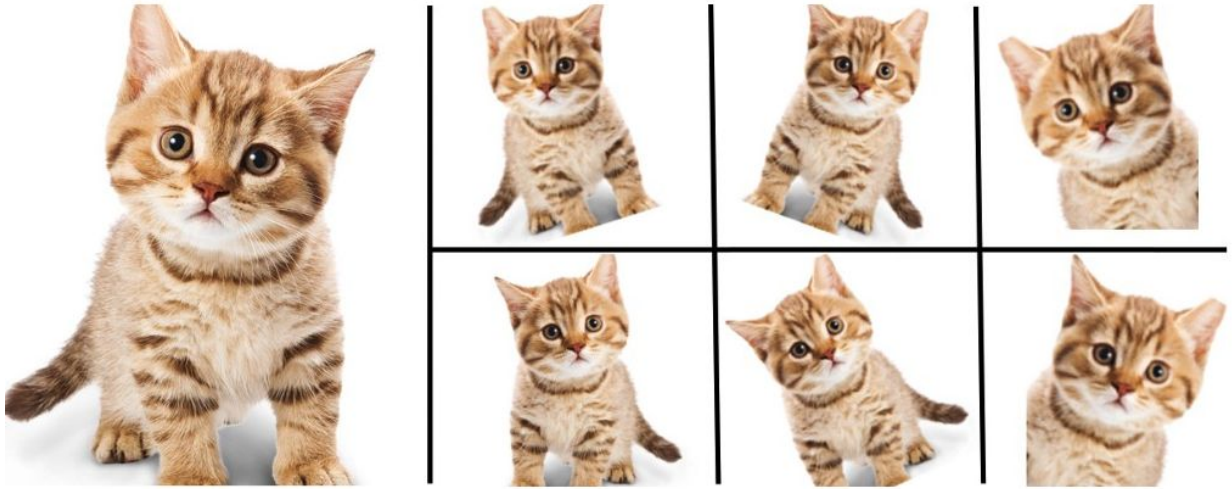
Jacob Devlin's slides:
https://nlp.stanford.edu/seminar/details/jdevlin.pdf

ICML slides 40-51:
https://media.neurips.cc/Conferences/NIPS2018/Slides/graves-deeplearning2.pdf

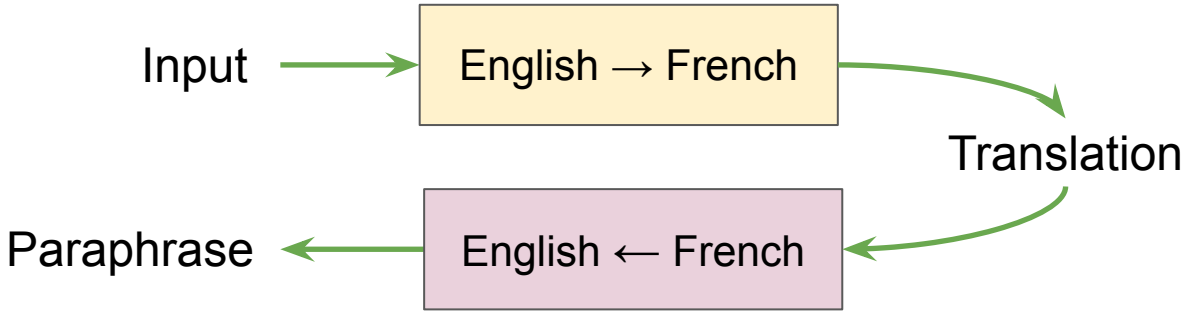
The Illustrated BERT.
http://jalammar.github.io/illustrated-bert/

QANet Idea #2: Data Augmentation for Text

Data augmentation: popular in vision & speech, but not NLP

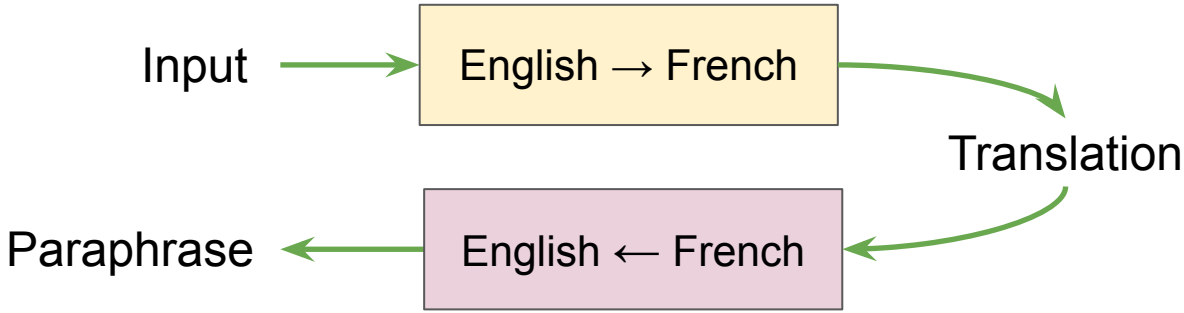


More data with NMT back-translation



More data with NMT back-translation

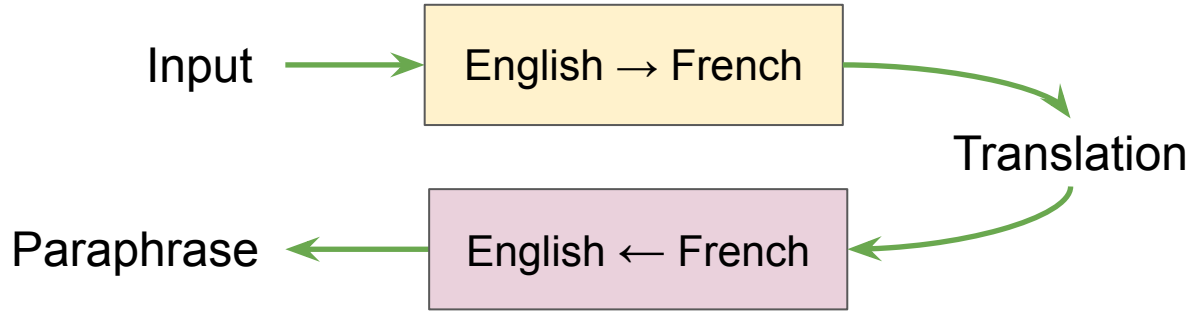
Previously, tea had been used primarily for Buddhist monks to stay awake during meditation.



Autrefois, le thé avait été utilisé surtout pour les moines bouddhistes pour rester éveillé pendant la méditation.

More data with NMT back-translation

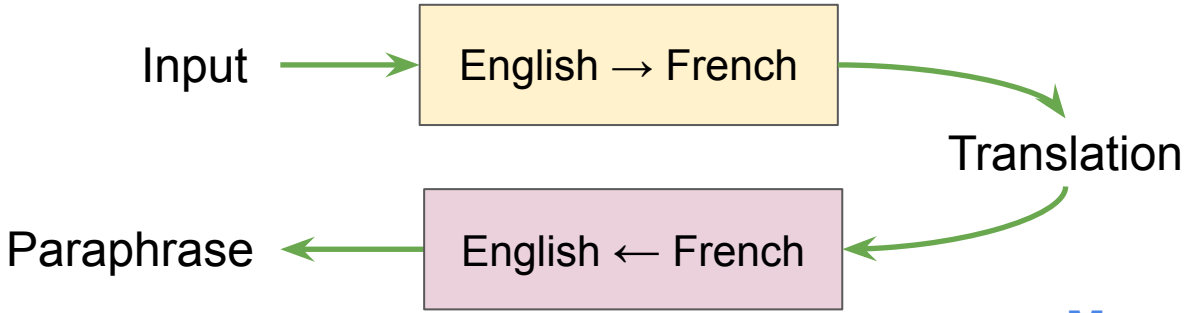
Previously, tea had been used primarily for Buddhist monks to stay awake during meditation.



In the past, tea was used mostly for Buddhist monks to stay awake during the meditation.

More data with NMT back-translation

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



In the past, tea was used mostly for Buddhist monks to stay awake during the meditation.

- **More data**
  - (Input, *label*)
  - (Paraphrase, *label*)

Applicable to virtually any NLP tasks!

QANet 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

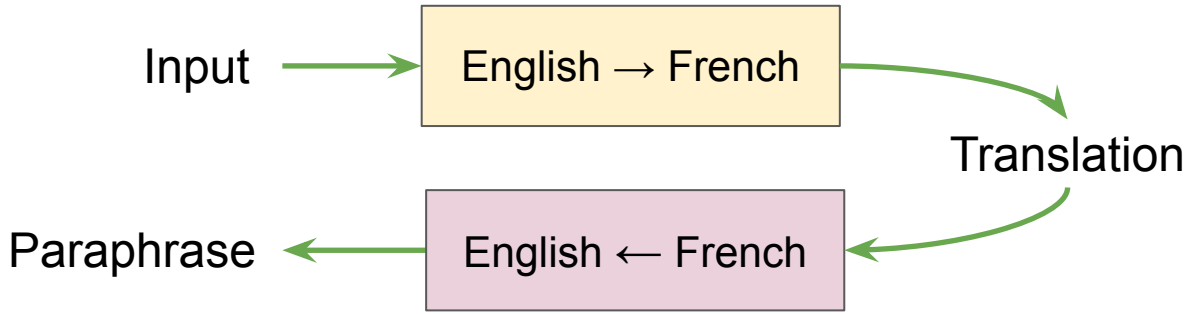
QANet augmentation

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-**fr**-en	… **Priests and retired brothers** reside at Fatima House (former public centre), the Holy Cross House and the Columbayn Hall near the Cave. ...	Priests and retired brothers

QANet augmentation



Use 2 language pairs: *English-French, English-German*. **3x data**.

Improvement: +1.1 F1

Failure cases

Question

Where by mass is oxygen a major part?

Context

… Oxygen constitutes 49.2 % of the Earth ' s crust by mass and is the major component of the **world ' s oceans** (**88.8 %** by mass) . Oxygen gas is the second most common component of the Earth ' s atmosphere , taking up 20.8 % of its volume and 23.1 % of its mass (some 1015 tonnes) ...

Ground truths

"world's oceans", "the world's oceans", “oceans"

Ensemble Answers

model 0: 88.8%
model 1: 88.8%
model 2: world's oceans
model 3: 88.8%
model 4: the world's oceans
model 5: 88.8%
model 6: 88.8%
model 7: 88.8%
model 8: world's oceans
model 9: world's oceans
model 10: 88.8%
model 11: 88.8%
model 12: world's oceans
model 13: world's oceans

Failure cases

Question

Who else did Tesla make the acquaintance of in 1886?

Context

… In late 1886 Tesla met Alfred S. Brown , a Western Union superintendent , and New York attorney **Charles F. Peck** … Together in April 1887 they formed the Tesla Electric Company with an agreement that profits from generated patents would go ⅓ to Tesla , ⅓ to Peck and Brown , and ⅓ to fund development .

Ground truths

'Charles F. Peck', 'New York attorney Charles F. Peck'

Ensemble Answers

model 0: Alfred S. Brown
model 1: Alfred S. Brown
model 2: Alfred S. Brown
model 3: Alfred S. Brown
model 4: Alfred S. Brown
model 5: Alfred S. Brown
model 6: Alfred S. Brown
model 7: Alfred S. Brown
model 8: Alfred S. Brown
model 9: Alfred S. Brown
model 10: Alfred S. Brown
model 11: Alfred S. Brown, a Western Union superintendent, and New York attorney Charles F. Peck
model 12: Alfred S. Brown
model 13: Peck and Brown

Học AI thế nào?

A typical day for researchers on Google's Brain Team

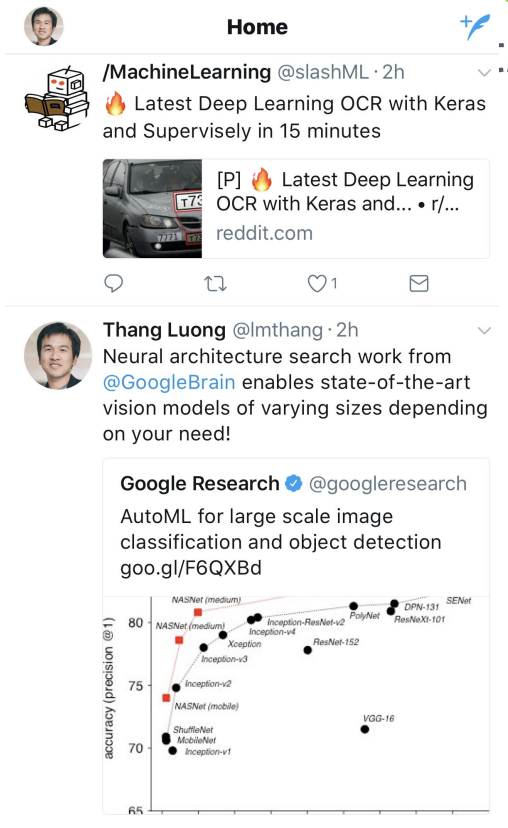
Posted Sep 15, 2017 by John Mannes (@JohnMannes)

- Everyone universally spends a lot of time reading papers on arXiv that are relevant to their research and collaborating with colleagues. Sara Hooker, a resident on the team, likes to chat with colleagues over breakfast — and lunch and dinner — to keep up to date on what other researchers are doing to solve similar problems.
- Chat with colleagues: *very important!*

https://techcrunch.com/2017/09/15/a-typical-day-for-researchers-on-googles-brain-team/

Cập nhật thông tin Deep Learning

- Twitter: most deep learning researchers.



Cập nhật thông tin Deep Learning

- Twitter: most deep learning researchers.
- Newsletters
  - Import AI: https://jack-clark.net/
  - The Wild Week in AI https://www.getrevue.co/profile/wildml

Import AI: #66: Better synthetic images heralds a Fake News future, GraphCore shows why AI chips are about to get very powerful, simulated rooms for better reinforcement learning

Welcome to Import AI, subscribe here.

NVIDIA achieves remarkable quality on synthetic images:
...AKA: The Fake Photo News Tidalwave Cometh
...NVIDIA researchers have released 'Progressive Growing of GANS for Improved QUality, Stability, and Variation' – a training methodology for creating better synthetic images with generative adversarial networks

The Wild Week in AI

October 2 - Issue #63 - View online

The Wild Week in AI is a weekly AI & Deep Learning newsletter curated by @dennybritz.

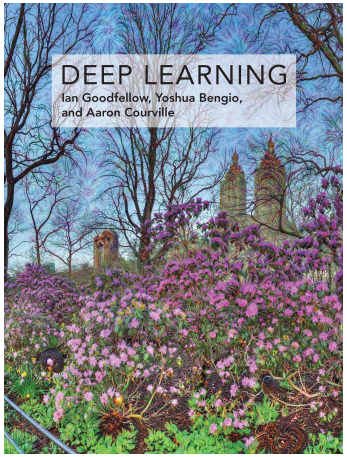
News

Theano Development Ends

GROUPS.GOOGLE.COM – Share

The MILA group will stop the development of Theano, one of the first and most popular Deep Learning frameworks. Many people have started their Deep Learning journey with Theano, which has been responsible for early breakthroughs and served as inspiration for frameworks like Tensorflow. Theano will continue to be available, but will not be maintained.

Tài liệu



- Deep learning book

- Online courses:
  - Coursera & Udacity
  - Stanford CS224N (language), CS231N (vision)
- Github
  - Learn from others & create your own

Thank you!