Advanced DS Overview: *Attention is all you need*

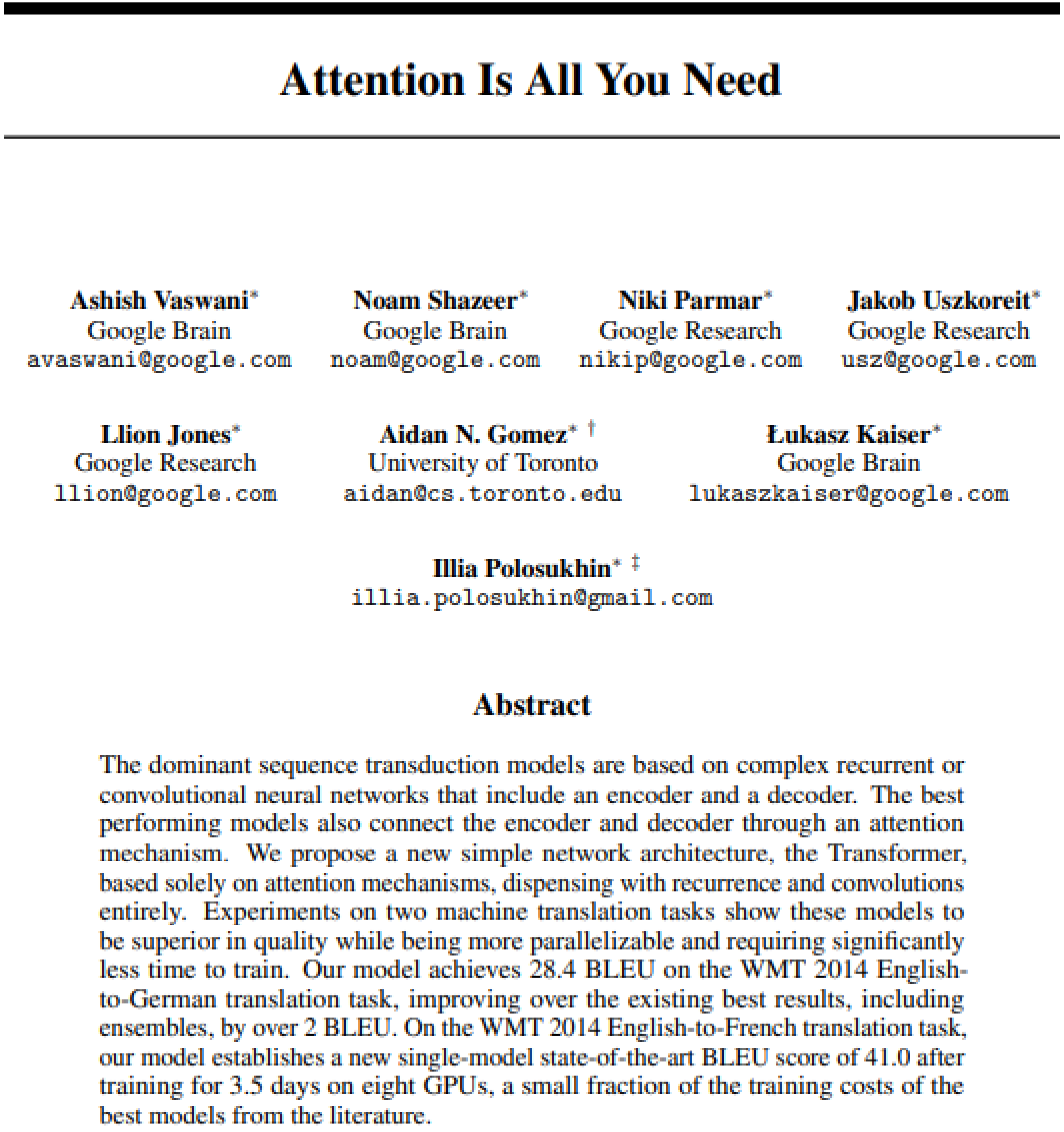
Data Science & Machine Learning Team

01/29/2021

Andrew Wheeler, PhD

andrew.wheeler@hms.com

Paper



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

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

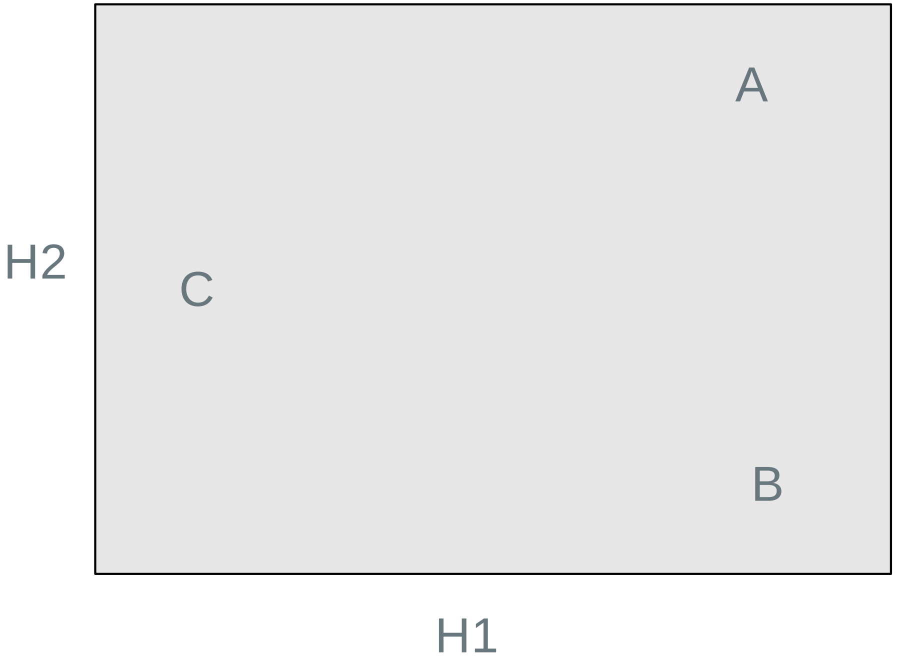
Step Through

- Very simple example problem: translate a,b,c to -> alpha, beta, gamma

- Data -> Encode -> Decode

Encode Matrix

Input	H1	H2
A	1.0	1.0
B	1.0	0.0
C	0.0	0.5



Step Through

- Very simple example problem: translate a,b,c to -> alpha, beta, gamma

- Data -> Encode -> Decode

Encode Matrix

Input	H1	H2
A	1.0	1.0
B	1.0	0.0
C	0.0	0.5

Notes:

Encode matrix is pretty much a synonymous with *Embedding Matrix*

Size of encode is [# of Inputs]*[# of Hidden Dimensions]
- e.g. [5000 ICD codes]*[100 Hidden] = 500,000

# of Hidden Dimensions is hyper-parameter you can choose

Step Through

- Very simple example problem: translate a,b,c to -> alpha, beta, gamma

- Data -> Encode -> Decode

Encode Matrix

Input	H1	H2
A	1.0	1.0
B	1.0	0.1
C	0.1	0.5

Decode Matrix

Input	alpha	beta	gamma
H1	0.6	0.9	-0.05
H2	0.6	-0.05	0.9

Step Through

Softmax([1.2, 0.85, 0.85]) -> [0.42 , 0.29, 0.29]

- Input [A] -> [1.0, 1.0] -> alpha = 1*0.6 + 1*0.6 = 1.2
- beta = 1*0.9 + -0.05*1 = 0.85
- gamma = 1*-0.05 + 1*0.9 = 0.85

Encode Matrix

Input	H1	H2
A	1.0	1.0
B	1.0	0.1
C	0.1	0.5

Decode Matrix

Input	alpha	beta	gamma
H1	0.6	0.9	-0.05
H2	0.6	-0.05	0.9

Step Through

- Matrix algebra: softmax(Encode[input]*Decode)
- Brief code in python

Encode Matrix

Input	H1	H2
A	1.0	1.0
B	1.0	0.1
C	0.1	0.5

Decode Matrix

Input	alpha	Beta	gamma
H1	0.6	0.9	-0.05
H2	0.6	-0.05	0.9

Attention is all you need

- Instead of doing RNN to account for temporal portions of the sequence, we use an attention layer:
  - Encode[input] * Decode
  - Pretend we have multiple input tokens: [a,a,c,a]
  - Encode[[a,a,c,a]]*Decode =

Long Output

Input	alpha	beta	gamma
a	1.20	0.85	0.85
a	1.20	0.85	0.85
c	0.36	0.07	0.45
a	1.20	0.85	0.85

Attention is all you need

- Attention layer is per the input, so here have attention layer of length 4 (can pad with 0s)
- This example, earlier inputs have more weight

Long Output

	a	a	c	a
Attn	0.60	0.25	0.10	0.05

*

Input	alpha	beta	gamma
a	1.20	0.85	0.85
a	1.20	0.85	0.85
c	0.36	0.07	0.45
a	1.20	0.85	0.85

=

alpha	beta	gamma
1.16	0.77	0.81

Attention is all you need

- Very simplified, paper has much more detail into task of sequence translation
  - Much of architecture is more 1 to 1 translation in paper, and use prior words to predict future
    - Eg input [a,b,c] -> output [alpha, beta, gamma], splits into for RNN:
      - [a] -> [alpha]
      - [a, b, prior1] -> [beta]
      - [a, b, c, prior1, prior2] -> [gamma]

Can’t be run in parallel

    - For Attention all you need, it is just:
      - [a,0,0], [a,b,0], [a,b,c] -> [alpha], [beta], [gamma]
      - With additional attention weights for each position

Can be run in parallel

- Attention is relevant for ICD codes when order matters (e.g. primary more important than secondary). [Did not cover positional encodings]

Advanced DS Overview: *Attention is all you need*

Data Science & Machine Learning Team

07/31/2020

Andrew Wheeler, PhD

andrew.wheeler@hms.com