1. <https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/>
2. <http://jalammar.github.io/illustrated-bert/>
3. <https://jalammar.github.io/illustrated-transformer/>
4. <https://towardsdatascience.com/nlp-extract-contextualized-embeddings-from-bert-keras-tf-67ef29f60a7b>
5. [Annotated Transformer](https://nlp.seas.harvard.edu/2018/04/03/attention.html): a blogpost by Professor Sasha Rush describing the transformer architecture by implementing it from the paper in PyTorch.

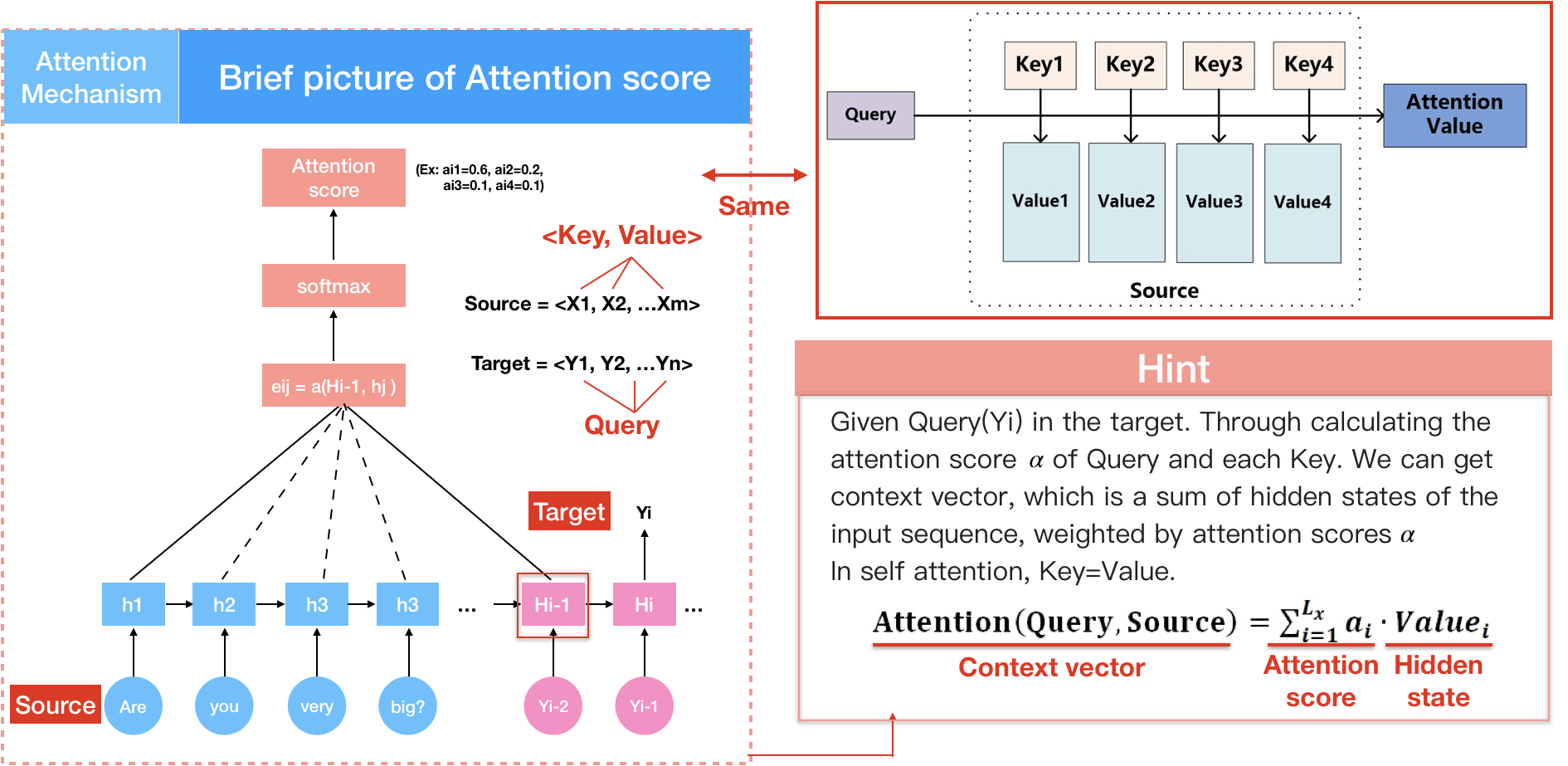
[**http://peterbloem.nl/blog/transformers**](http://peterbloem.nl/blog/transformers)

<https://colab.research.google.com/github/tensorflow/tensor2tensor/blob/master/tensor2tensor/notebooks/hello_t2t.ipynb>

<https://mlexplained.com/2017/12/29/attention-is-all-you-need-explained/>

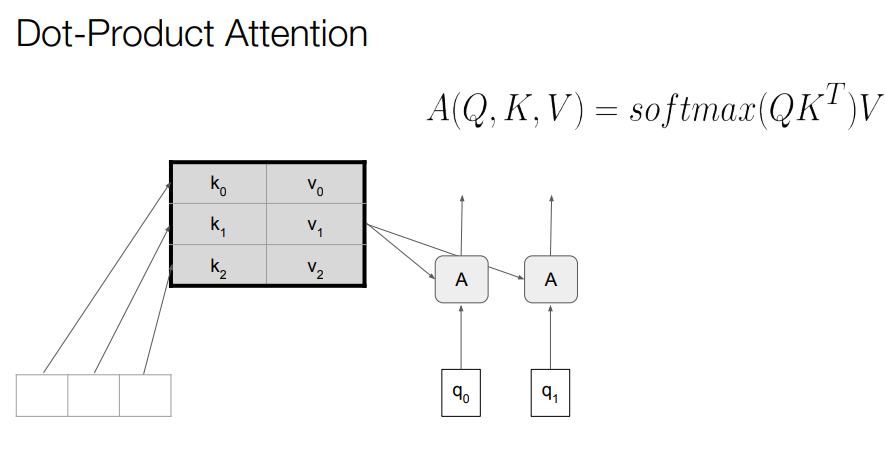
<https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html>

[**https://medium.com/@bgg/seq2seq-pay-attention-to-self-attention-part-2-cf81bf32c73d**](https://medium.com/@bgg/seq2seq-pay-attention-to-self-attention-part-2-cf81bf32c73d)



<https://www.youtube.com/watch?v=rBCqOTEfxvg>

<https://nlp.stanford.edu/seminar/details/lkaiser.pdf>

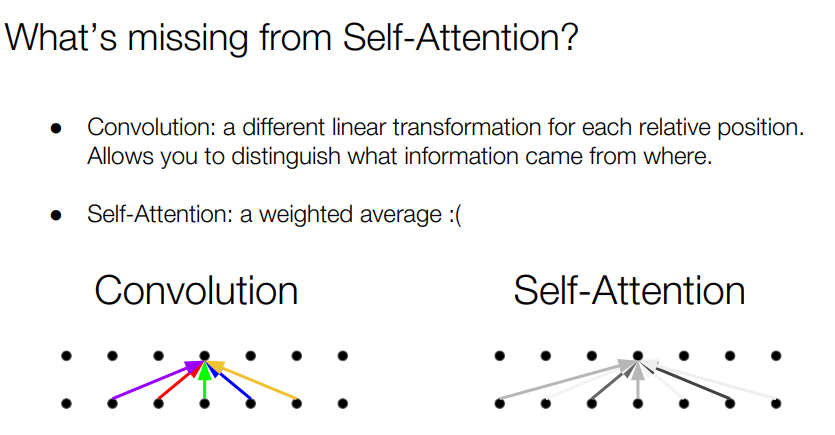


so what is exactly this attention so we 15:52 define attention there is a query that 15:54 will be a vector there is a and then15:58there is this key and value match matrix16:00which is your memory so you can think16:02like this is the current word what I'm16:04operating on and this is all the past16:07all the words I've generated before and16:09keys and values can be the same thing16:11but they don't have to does that's why16:13we write it like this so what you want16:16to do in attention is take the query16:19

find the most similar key as I told you16:23it's it's a similarity thing and then16:26get the values that correspond to these16:29similar keys but of course it's soft at16:32or needs to be differentiable so how to16:37do this efficiently well you take the16:39query16:40multiply it by the transposed keys so16:43

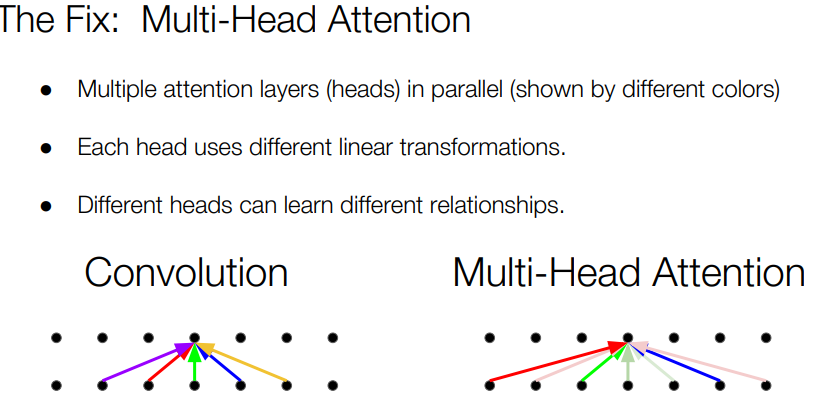
that's a matrix multiply16:45then you take a softmax which which is16:47like this exponentiation and16:49normalization and that gives you a mask16:54that gives you a probability16:55distribution over keys which is peaked16:59at the ones that are similar to the17:01query and then this mask you make a17:05matrix multiplied with the values which17:08is the same as summing over values17:10

multiplied by this mask and there you17:13are17:14you4re what you want you've done to17:17matrix multiply and one softmax17:19operation which is very fast actually17:23you need to normalize it a little bit to



so so so we did it and there are so one19:03problem with attention since it's just a19:04similaritymeasure it works as this was19:08as if this was just a set of words it19:11has no ideathat this word comes after19:14this because it just retrieves the most19:15similar ones but it'sreally important19:18the order of words is not arbitrary you19:21cannot just reorder and hope that it19:23will translate well or or generate the19:25right next word do you need to add some19:27timing signal and you need to add some a19:29little bit of the positional things so19:34so we have the multi hat attention with19:36positional signals I will not go into19:40the details of how this is done so there

19:42will be a multiple attention heads



<https://nlp.seas.harvard.edu/2018/04/03/attention.html>

**How transformers reduce sequectial compution:**

The goal of reducing sequential computation also forms the foundation of the Extended Neural GPU [16], ByteNet [18] and ConvS2S [9], all of which use convolutional neural networks as basic building block, computing hidden representations in parallel for all input and output positions. I***n these models, the number of operations required to relate signals from two arbitrary input or output positions grows in the distance between positions, linearly for ConvS2S and logarithmically for ByteNet***. **This makes it more difficult to learn dependencies between distant positions** [12]. **In the Transformer this is reduced to a constant number of operations, albeit at the cost of reduced effective resolution due to averaging attention-weighted positions, an effect we counteract with Multi-Head Attention as described in section 3.2**.

Why multi head in attention?