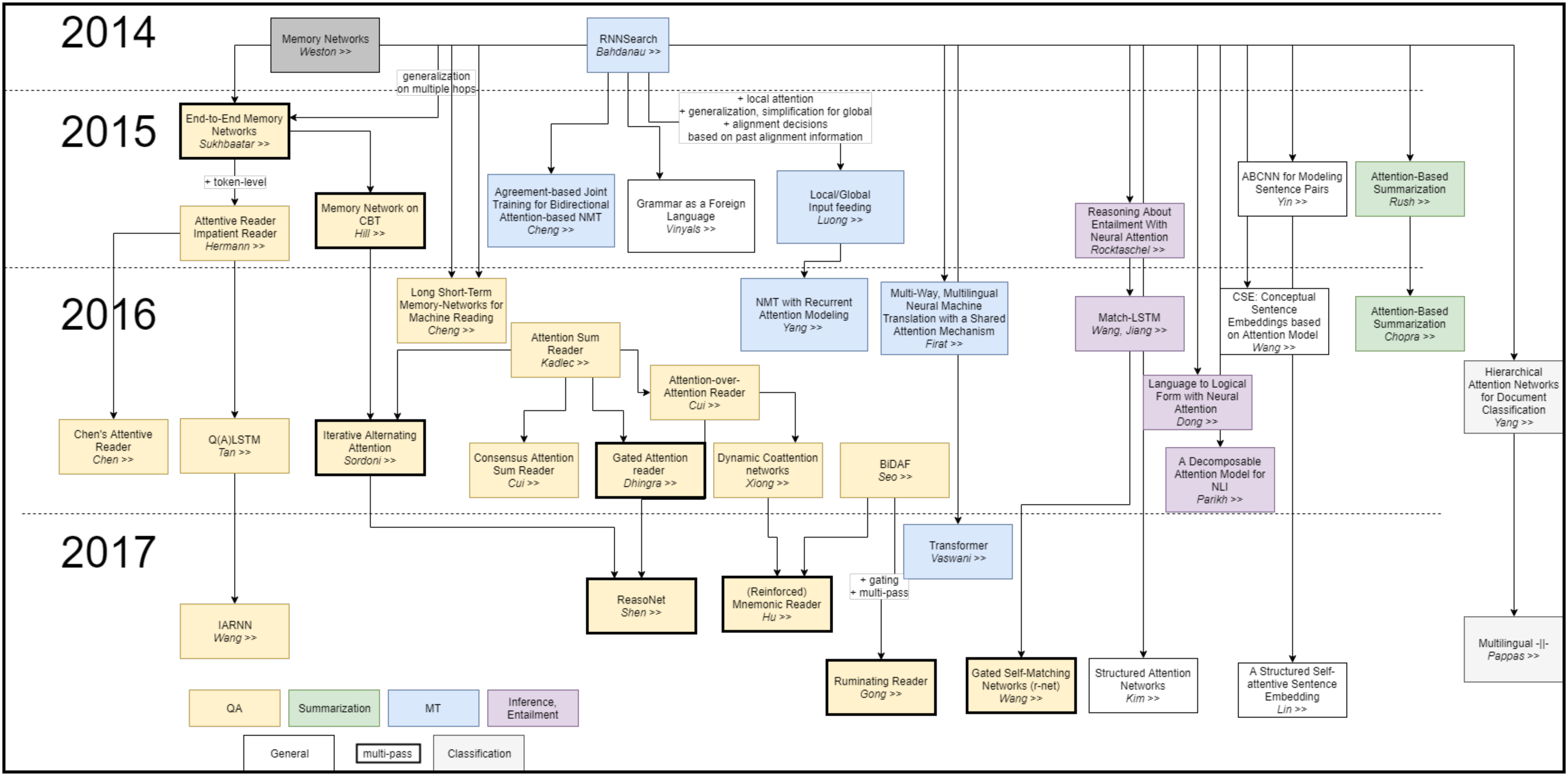
You Just Want Attent!on

Charlie Puth (Actually: Omar Attia)
CS886: Deep Learning and NLP
Winter 2020

Charlie Puth
Vs.
Google Research

- Charlie Puth released his hit "You Just Want Attention" in April 2017 (1B+ views)
- Google Research did not publish "Attention Is All You Need" until June 2017 (5K+ citations)
- Coincidence??



2014

Memory Networks
Weston >>

RNNSearch
Bahdanau >>

generalization
on multiple hops

+ local attention
+ generalization, simplification for global
+ alignment decisions
based on past alignment information

2015

End-to-End Memory Networks
Sukhbaatar >>

+ token-level

Attentive Reader
Impatient Reader
Hermann >>

Memory Network on CBT
Hill >>

Agreement-based Joint Training for Bidirectional Attention-based NMT
Cheng >>

Grammar as a Foreign Language
Vinyals >>

Local/Global
Input feeding
Luong >>

Reasoning About Entailment With Neural Attention
Rocktaschel >>

ABCNN for Modeling Sentence Pairs
Yin >>

Attention-Based Summarization
Rush >>

2016

Long Short-Term Memory-Networks for Machine Reading
Cheng >>

Attention Sum Reader
Kadlec >>

Attention-over-Attention Reader
Cui >>

NMT with Recurrent Attention Modeling
Yang >>

Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism
Firat >>

Match-LSTM
Wang, Jiang >>

CSE: Conceptual Sentence Embeddings based on Attention Model
Wang >>

Attention-Based Summarization
Chopra >>

Hierarchical Attention Networks for Document Classification
Yang >>

Language to Logical Form with Neural Attention
Dong >>

Chen's Attentive Reader
Chen >>

Q(A)LSTM
Tan >>

Iterative Alternating Attention
Sordoni >>

Consensus Attention Sum Reader
Cui >>

Gated Attention reader
Dhingra >>

Dynamic Coattention networks
Xiong >>

BiDAF
Seo >>

A Decomposable Attention Model for NLI
Parikh >>

2017

Transformer
Vaswani >>

IARNN
Wang >>

ReasoNet
Shen >>

(Reinforced) Mnemonic Reader
Hu >>

+ gating
+ multi-pass

Multilingual -||-
Pappas >>

Ruminating Reader
Gong >>

Gated Self-Matching Networks (r-net)
Wang >>

Structured Attention Networks
Kim >>

A Structured Self-attentive Sentence Embedding
Lin >>

QA

Summarization

MT

Inference, Entailment

General

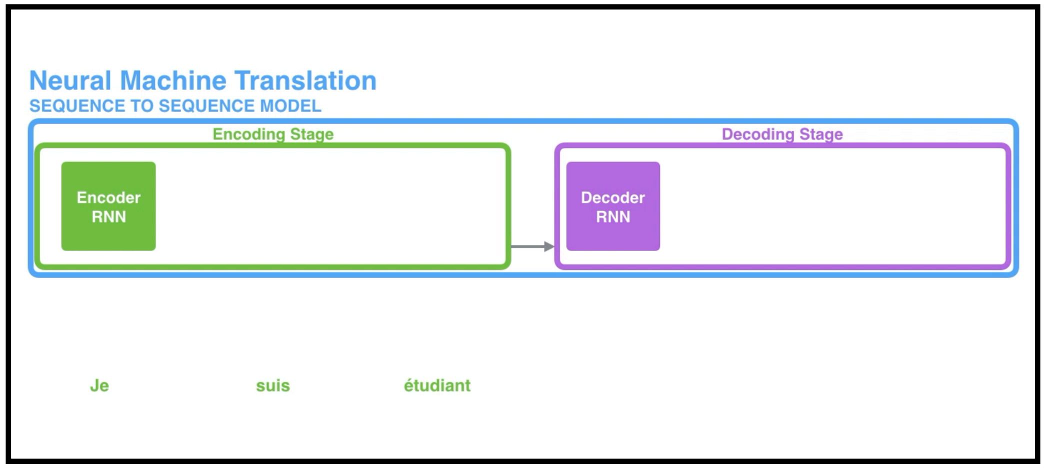
multi-pass

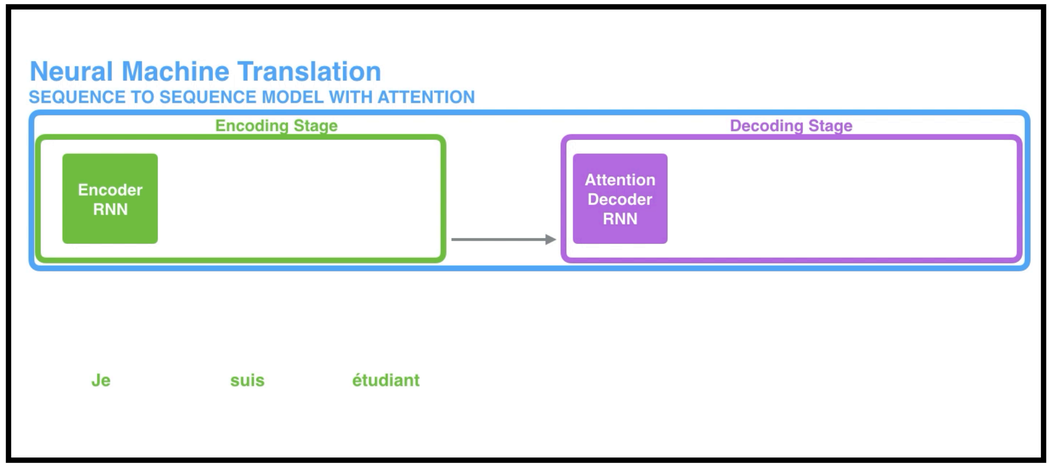
Classification

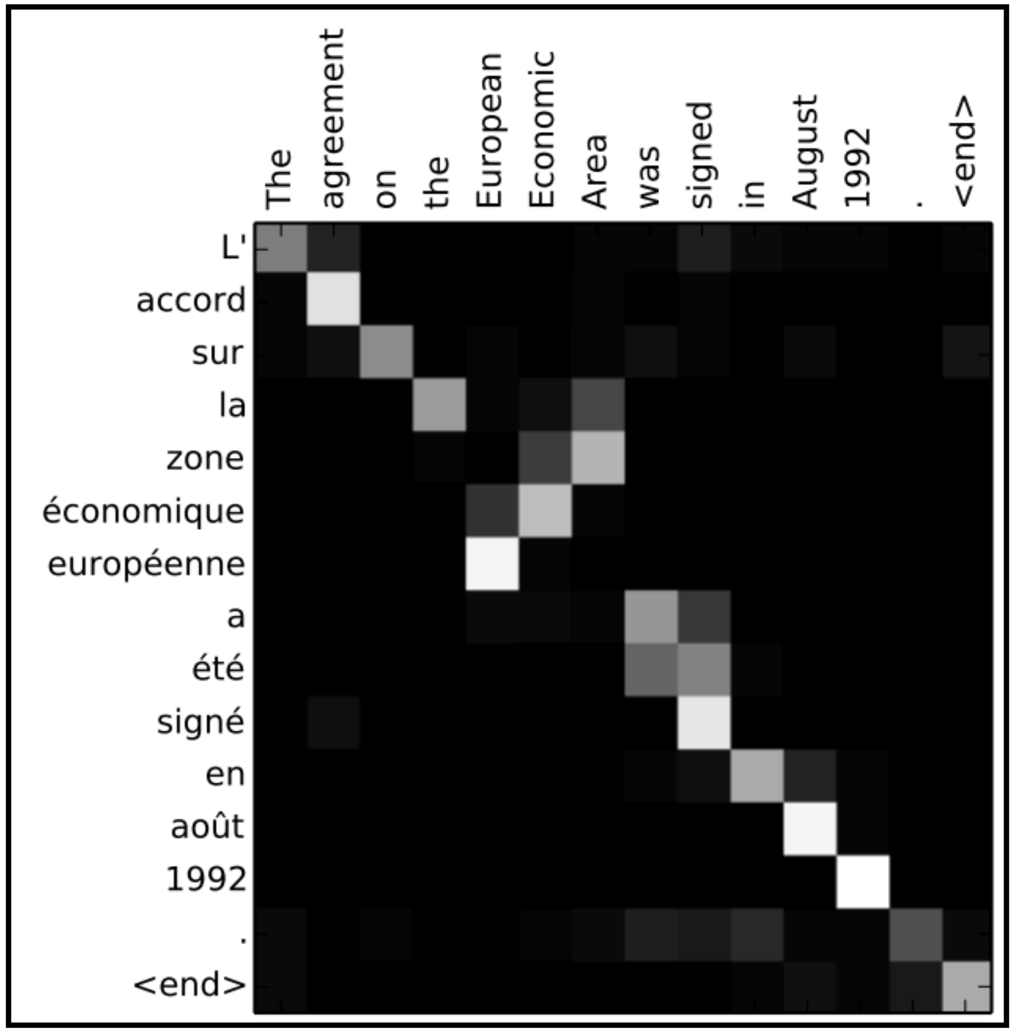
Agenda:

- Motivation
- What Is Attention?
- How to Compute Attention?
- Attention Functions
- The Cost of Attention
- Attention in NLP
- BERT Attention Analysis

Motivation







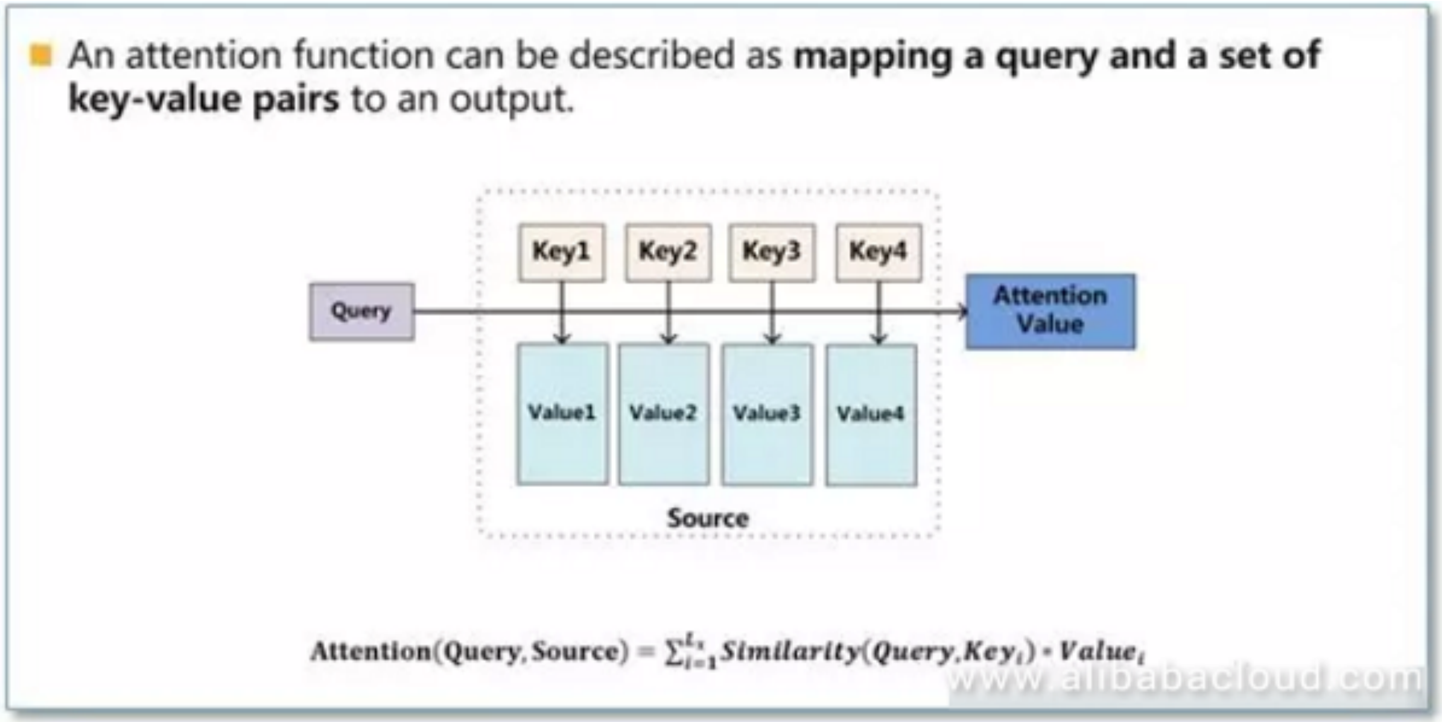
- First proposed in Bahdanau et al., 2014 as an alignment mechanism.

- Basically, Attention allows the model to focus on the relevant parts of the input sequence as needed (Think of how humans pay attention to only most relevant parts of a scene)

What Is Attention?

- Attention can be described as mapping a query (Q) and a set of key-value pairs (<K, V>) to an output (Z), where the query, keys, values, and output are all vectors.

- The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function (f) of the query with the corresponding key.



An attention function can be described as **mapping a query and a set of key-value pairs** to an output.

$$\text{Attention}(\text{Query}, \text{Source}) = \sum_{i=1}^{L_x} \mathit{Similarity}(\mathit{Query}, \mathit{Key}_i) * \mathit{Value}_i$$

The K, V, Q notation is inspired by the information retrieval literature.

How to Compute Attention?

- Calculating Attention:

  1. Similarity between the query and each key to obtain scalar scores.

  2. Use a Softmax function to normalize these scores into weights.

  3. Apply weighted sum of the values.

- Major components in attention mechanisms:

  1. Define <K, V> and Q

  2. Choose a similarity function

In a lot of NLP work, the key and value are often the same, therefore key=value=word-embeddings

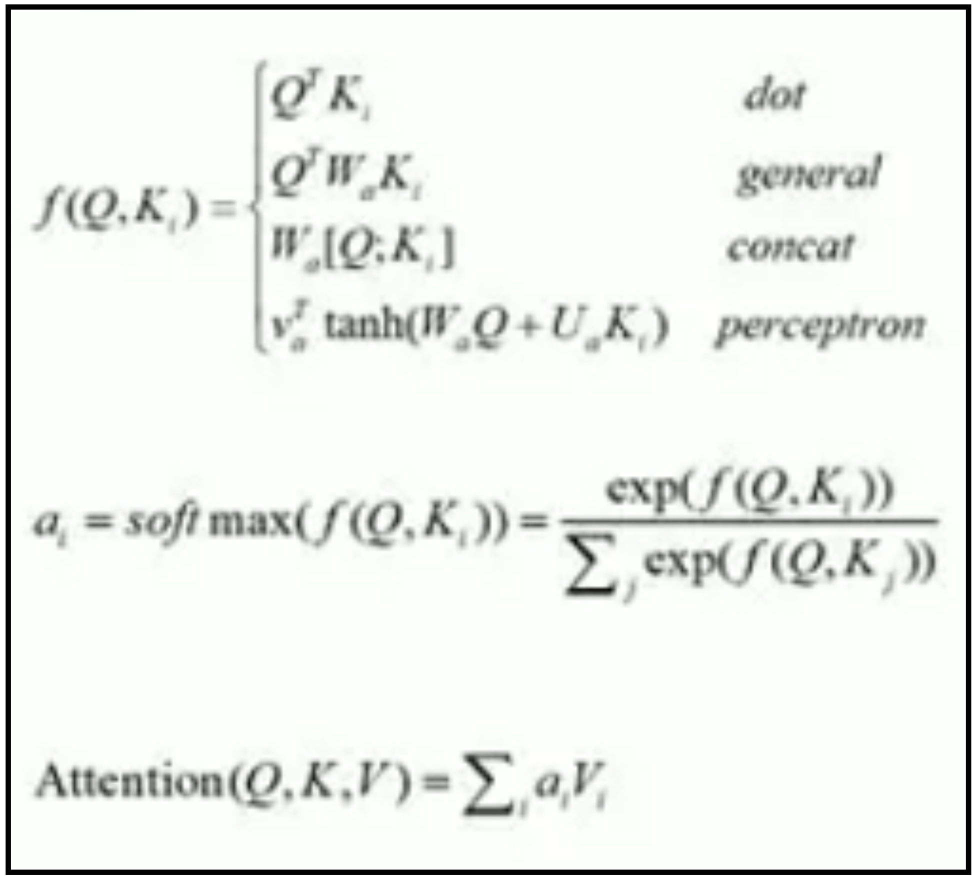
$$Attention(Q, K_i, V_i) = \frac{e^{Similarity(Q,K_i)}}{\sum e^{Similarity(Q,K_i)}} V_i$$

Attention Functions

- Additive Attention:
- The attention scores are computed using a Perceptron.
- Example:
  LSTM + Attention

- Multiplicative Attention:
- The attention scores are computed using (scaled/general) dot products.
- Example:
  Transformer

- Additive attention outperforms Multiplicative attention without scaling for larger dimensions, because the dot products grow large in magnitude, pushing the softmax function into regions where it has extremely small gradients.



$$f(Q,K_i)=\begin{cases} Q^T K_i & dot \\ Q^T W_a K_i & general \\ W_a[Q;K_i] & concat \\ v_a^T \tanh(W_a Q + U_a K_i) & perceptron \end{cases}$$

$$a_i = softmax(f(Q,K_i)) = \frac{\exp(f(Q,K_i))}{\sum_j \exp(f(Q,K_j))}$$

$$\text{Attention}(Q,K,V) = \sum_i a_i V_i$$

The Cost of Attention

- Human attention is something that’s supposed to **save** computational resources. By focusing on one thing, we can neglect many other things.

- However, we need to calculate an attention value for *each combination* of input and output objects (for long sequences, it could become prohibitively expensive).

- Essentially looks at everything in detail before deciding what to attend to.

- Intuitively that’s equivalent outputting a translated word, and then going back through *all* of your internal memory of the text to decide which word to produce next. That seems like a waste, and not at all what humans are doing.

- In fact, it’s more akin to memory access (Information Retrieval), not attention.

Agenda:

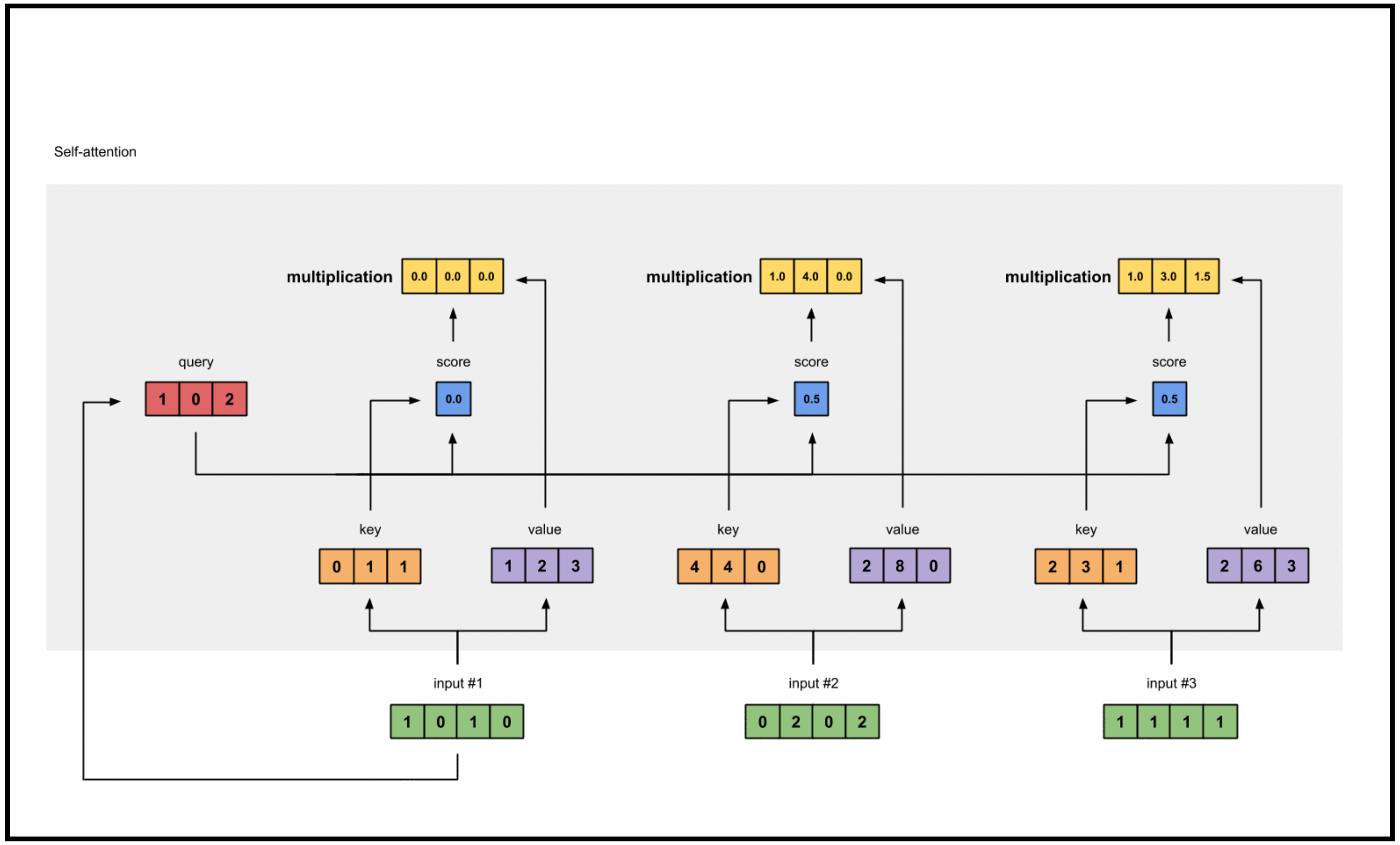
- ~~Motivation~~
- ~~What Is Attention?~~
- ~~How to Compute Attention?~~
- ~~The Cost of Attention~~
- ~~Attention Functions~~
- Attention in NLP:
    - Self Attention
    - Multi-Head Attention
    - Hierarchical Attention
    - Co-Attention
    - Two-Way Attention
    - Key-Value-Predict Attention
- BERT Attention Analysis

Self Attention

- Application:
  Representation Learning + Machine Translation (Transformer, BERT, GPT)

- Intuition: When there is no external information available, relate different positions of the
  same sequence to compute its internal representation (i.e. Q = K = V)

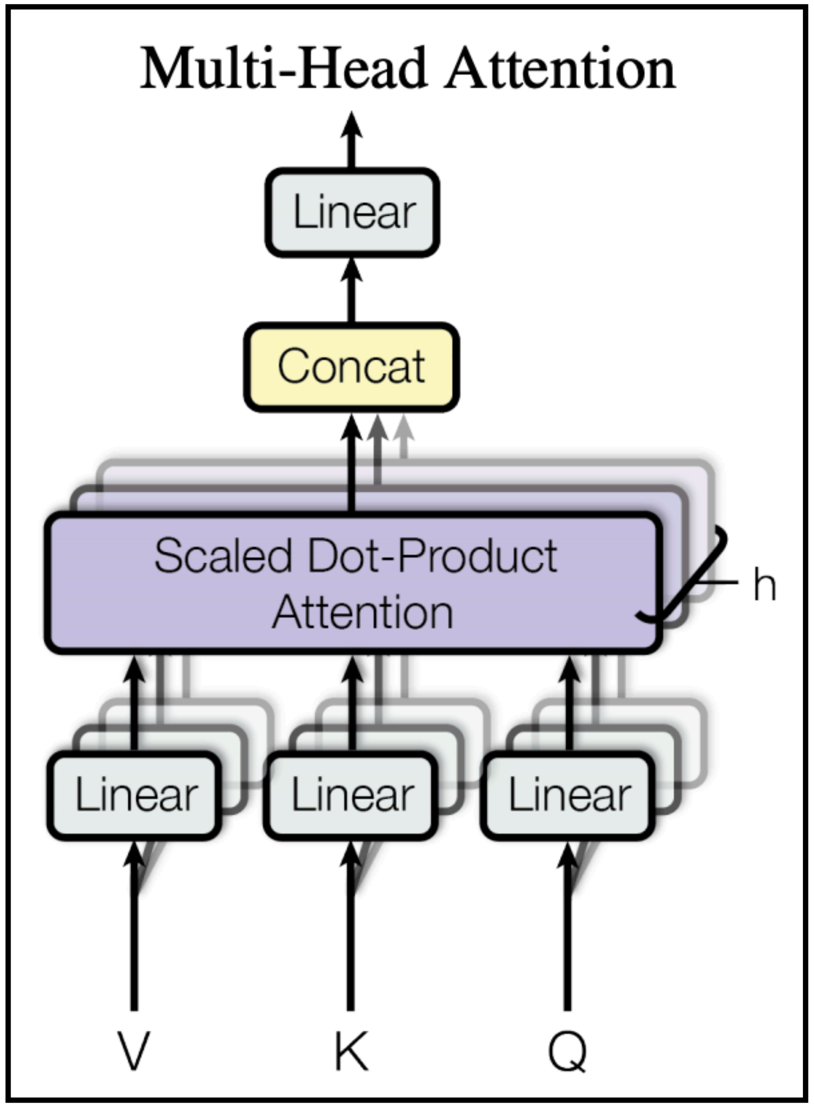
- Seems to learn sophisticated syntactic patterns! (More on that later)



Multi-Head Attention

- Application:
  Representation Learning + Machine Translation (Transformer, BERT, GPT)

- Intuition : The more the merrier!Allow the model to jointly attend to information from different representation subspaces at different positions.



$$\begin{aligned}\mathrm{MultiHead}(Q,K,V) &= \mathrm{Concat}(\mathrm{head_1},...,\mathrm{head_h})W^O\\ \text{where}\ \mathrm{head_i} &= \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)\end{aligned}$$

Hierarchical Attention

- Application: Document Classification
- Intuition: Employ two levels of attention: one at the word level and one at the sentence level, to capture and utilize the hierarchical nature of documents (words —> sentences —> document)

Word Level:

$$
\begin{aligned}
x_{it} &= W_e w_{it}, t \in [1, T], \\
\overrightarrow{h}_{it} &= \overrightarrow{\text{GRU}}(x_{it}), t \in [1, T], \\
\overleftarrow{h}_{it} &= \overleftarrow{\text{GRU}}(x_{it}), t \in [T, 1].
\end{aligned}
$$

1

$$
\begin{aligned}
u_{it} &= \tanh(W_w h_{it} + b_w) \\
\alpha_{it} &= \frac{\exp(u_{it}^\top u_w)}{\sum_t \exp(u_{it}^\top u_w)} \\
s_i &= \sum_t \alpha_{it} h_{it}.
\end{aligned}
$$

2

Sentence Level:

$$
\begin{aligned}
\overrightarrow{h}_i &= \overrightarrow{\text{GRU}}(s_i), i \in [1, L], \\
\overleftarrow{h}_i &= \overleftarrow{\text{GRU}}(s_i), t \in [L, 1].
\end{aligned}
$$

1

$$
\begin{aligned}
u_i &= \tanh(W_s h_i + b_s), \\
\alpha_i &= \frac{\exp(u_i^\top u_s)}{\sum_i \exp(u_i^\top u_s)}, \\
v &= \sum_i \alpha_i h_i,
\end{aligned}
$$

2

Classification:

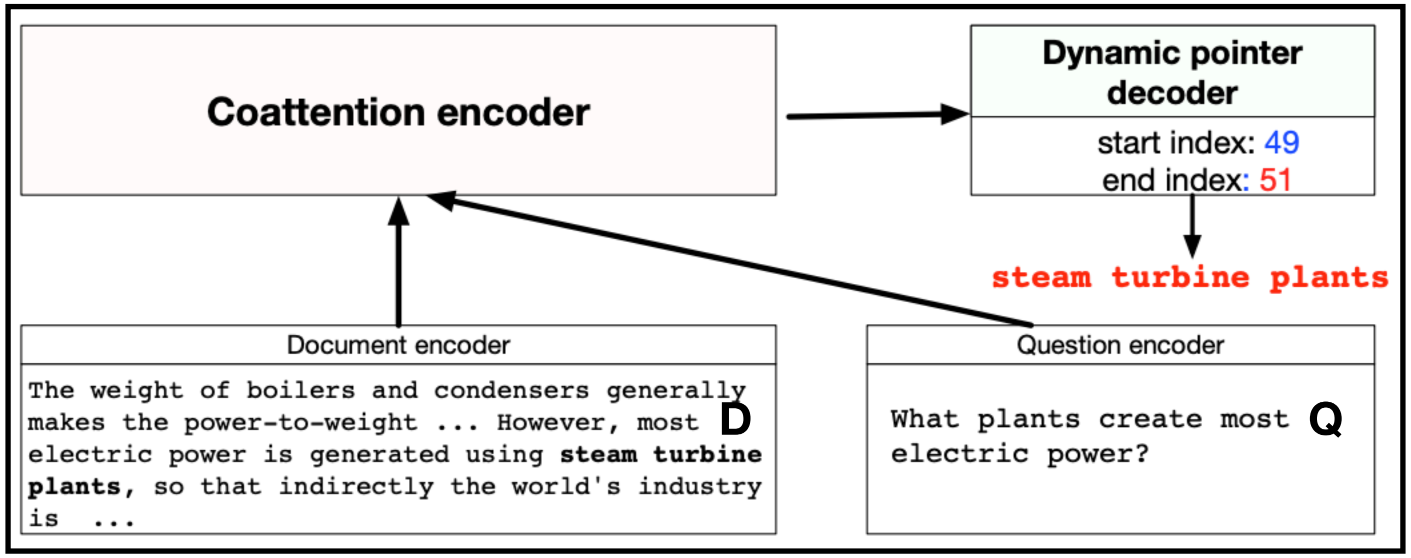
$$p = \text{softmax}(W_c v + b_c).$$

Co-Attention

- Application:
  Question Answering

- Intuition: Co-attention is computed as an alignment matrix on *all* pairs of document and query words.

- Extra: Use an iterative procedure to select an answer span by alternating between predicting the start point and predicting the end point. This allows for recovering from initial local maxima corresponding to incorrect answer spans.



The weight of boilers and condensers generally makes the power-to-weight ... However, most electric power is generated using **steam turbine plants**, so that indirectly the world's industry is ...

What plants create most electric power?

$$\boxed{D = [d_1 \;\ldots\; d_m \; d_\varnothing]}$$

$$\boxed{Q' = [q_1 \;\ldots\; q_n \; q_\varnothing]}$$

$$\boxed{Q = \tanh\left(W^{(Q)}Q' + b^{(Q)}\right) \in \mathbb{R}^{\ell \times (n+1)}.}$$

$$\boxed{L = D^\top Q \in \mathbb{R}^{(m+1)\times(n+1)}}$$

$$\boxed{A^D = \mathrm{softmax}\left(L^\top\right) \in \mathbb{R}^{(n+1)\times(m+1)}}$$

$$\boxed{A^Q = \mathrm{softmax}\left(L\right) \in \mathbb{R}^{(m+1)\times(n+1)}}$$

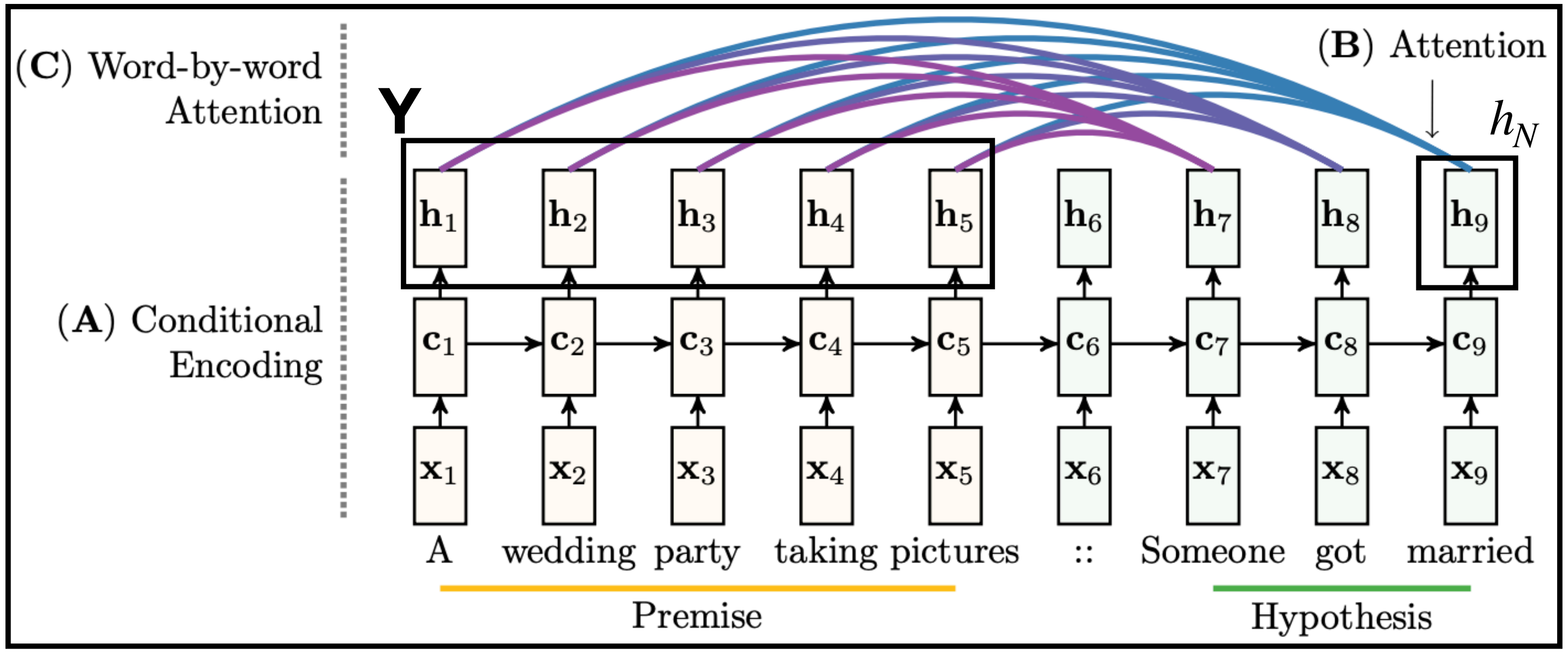
$$\boxed{C^Q = DA^Q \in \mathbb{R}^{\ell \times (n+1)}.}$$

$$\boxed{C^D = \left[Q; C^Q\right] A^D \in \mathbb{R}^{2\ell \times (m+1)}.}$$

Two-Way Attention

- Application: Textual Entailment

- Intuition: Use the same model to attend over the premise conditioned on the hypothesis, as well as to attend over the hypothesis conditioned on the premise, by simply swapping the two sequences. This produces two sentence-pair representations that we concatenate for classification.



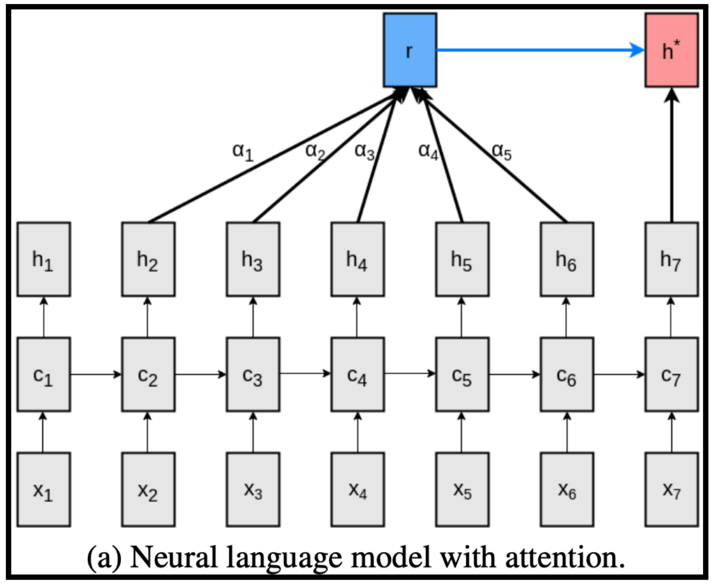
(C) Word-by-word Attention
(B) Attention
h_N
Y
$\mathbf{h}_1$ $\mathbf{h}_2$ $\mathbf{h}_3$ $\mathbf{h}_4$ $\mathbf{h}_5$ $\mathbf{h}_6$ $\mathbf{h}_7$ $\mathbf{h}_8$ $\mathbf{h}_9$
(A) Conditional Encoding
$\mathbf{c}_1$ $\mathbf{c}_2$ $\mathbf{c}_3$ $\mathbf{c}_4$ $\mathbf{c}_5$ $\mathbf{c}_6$ $\mathbf{c}_7$ $\mathbf{c}_8$ $\mathbf{c}_9$
$\mathbf{x}_1$ $\mathbf{x}_2$ $\mathbf{x}_3$ $\mathbf{x}_4$ $\mathbf{x}_5$ $\mathbf{x}_6$ $\mathbf{x}_7$ $\mathbf{x}_8$ $\mathbf{x}_9$
A wedding party taking pictures :: Someone got married
Premise
Hypothesis

$$
\begin{aligned}
\mathbf{M} &= \tanh(\mathbf{W}^y\mathbf{Y} + \mathbf{W}^h\mathbf{h}_N \otimes \mathbf{e}_L) & \mathbf{M} &\in \mathbb{R}^{k\times L}\\
\alpha &= \mathrm{softmax}(\mathbf{w}^T\mathbf{M}) & \alpha &\in \mathbb{R}^{L}\\
\mathbf{r} &= \mathbf{Y}\alpha^T & \mathbf{r} &\in \mathbb{R}^{k}
\end{aligned}
$$

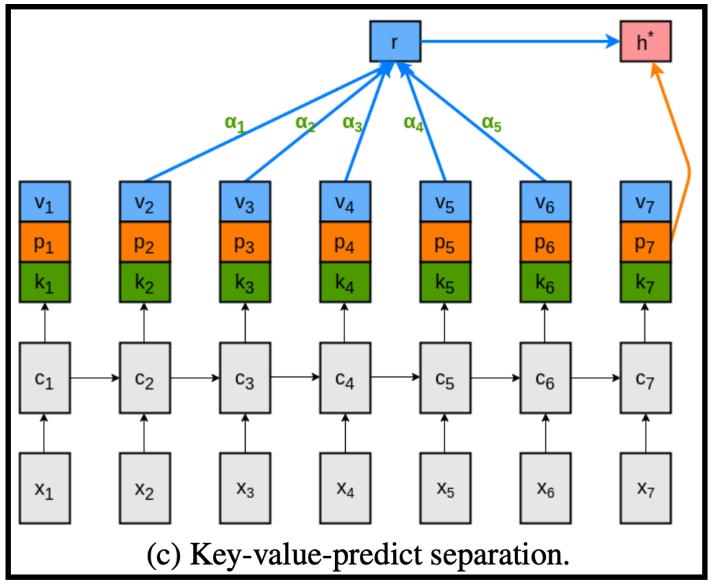
$$
\begin{aligned}
\mathbf{M}_t &= \tanh(\mathbf{W}^y\mathbf{Y} + (\mathbf{W}^h\mathbf{h}_t + \mathbf{W}^r\mathbf{r}_{t-1}) \otimes \mathbf{e}_L) & \mathbf{M}_t &\in \mathbb{R}^{k\times L}\\
\alpha_t &= \mathrm{softmax}(\mathbf{w}^T\mathbf{M}_t) & \alpha_t &\in \mathbb{R}^{L}\\
\mathbf{r}_t &= \mathbf{Y}\alpha_t^T + \tanh(\mathbf{W}^t\mathbf{r}_{t-1}) & \mathbf{r}_t &\in \mathbb{R}^{k}
\end{aligned}
$$

Key-Value-Predict Attention

- Application: Language Modelling
- Intuition: Using K = V for all applications can make training difficult (K=V simultaneously store information for predicting the next word, computing the attention, and encode content relevant to future steps)



(a) Neural language model with attention.



(c) Key-value-predict separation.

$$\boxed{\begin{bmatrix}\boldsymbol{k}_t\\ \boldsymbol{v}_t\\ \boldsymbol{p}_t\end{bmatrix} = \boldsymbol{h}_t}$$

$$\boxed{\begin{aligned}\boldsymbol{M}_t &= \tanh(\boldsymbol{W}^Y[\boldsymbol{k}_{t-L} \;\cdots\; \boldsymbol{k}_{t-1}] + (\boldsymbol{W}^h\boldsymbol{k}_t)\boldsymbol{1}^T)\\ \boldsymbol{\alpha}_t &= \mathrm{softmax}(\boldsymbol{w}^T\boldsymbol{M}_t)\\ \boldsymbol{r}_t &= [\boldsymbol{v}_{t-L} \;\cdots\; \boldsymbol{v}_{t-1}]\boldsymbol{\alpha}^T\end{aligned}}$$

$$\boxed{\boldsymbol{h}_t^* = \tanh(\boldsymbol{W}^r\boldsymbol{r}_t + \boldsymbol{W}^x\boldsymbol{p}_t)}$$

BERT Attention Analysis

- The massive success of pre-trained attention-based models (even though these models are trained in a self-supervised fashion on unlabeled data, without explicit supervision for syntax or coreference) begs the question: What specific linguistic features do they learn?

- Methodology: Collect various statistics (e.g. average entropy, average attention weight per token) and study attention maps of BERT on different datasets.

- Some Findings:

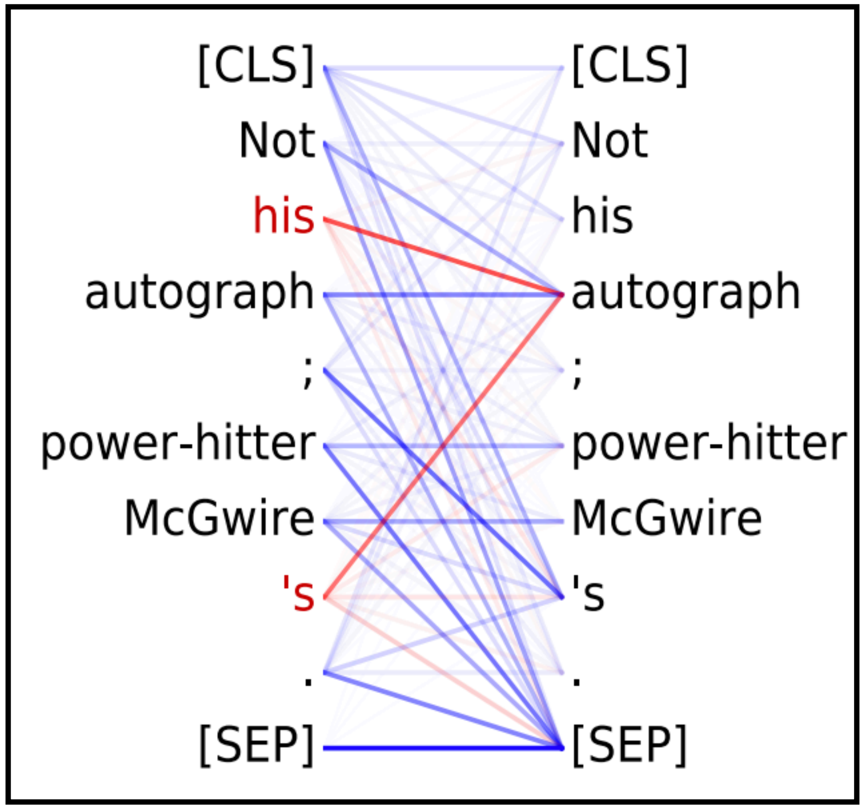
  - Most heads put little attention on the current token. However, there are heads that specialize in attending heavily on the next or previous token, especially in earlier layers of the network.

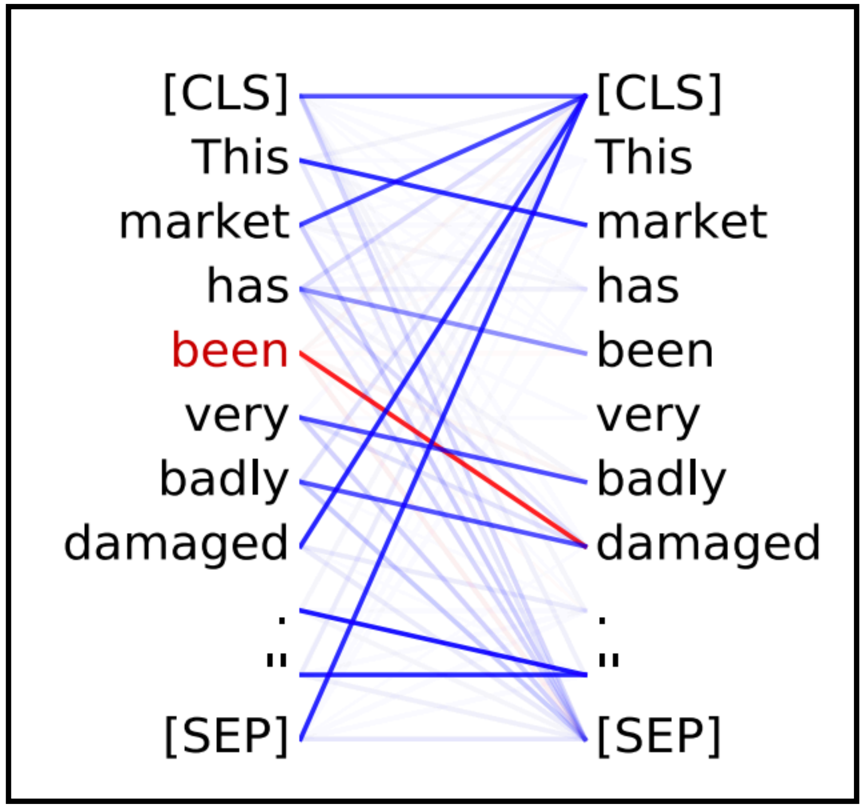
  - Over half of BERT's attention in layers 6-10 focuses on the delimiter token [SEP], which could be used by the model as a sort of “no-op”.

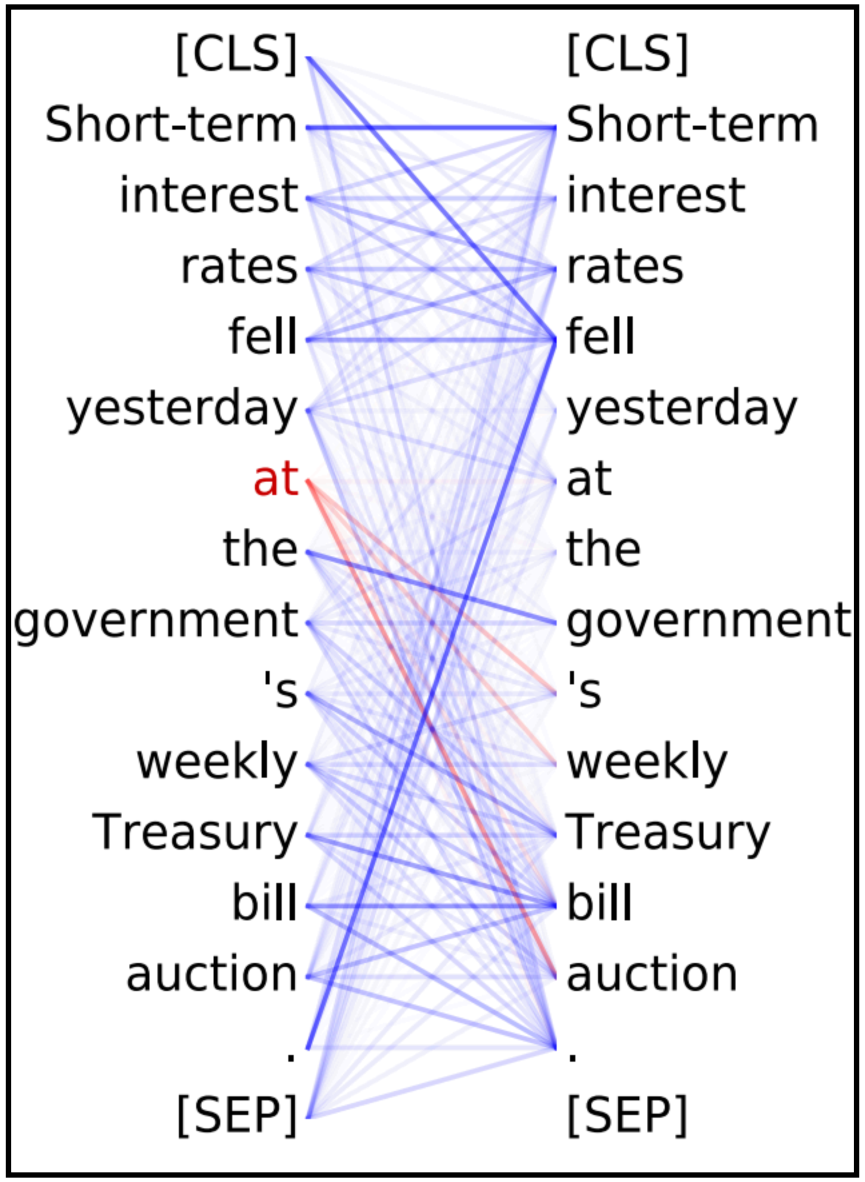
  - Some attention heads, especially in lower layers, have very broad attention (at most 10% of their attention mass on any single word). The output of these heads is roughly a bag-of-vectors representation of the sentence.

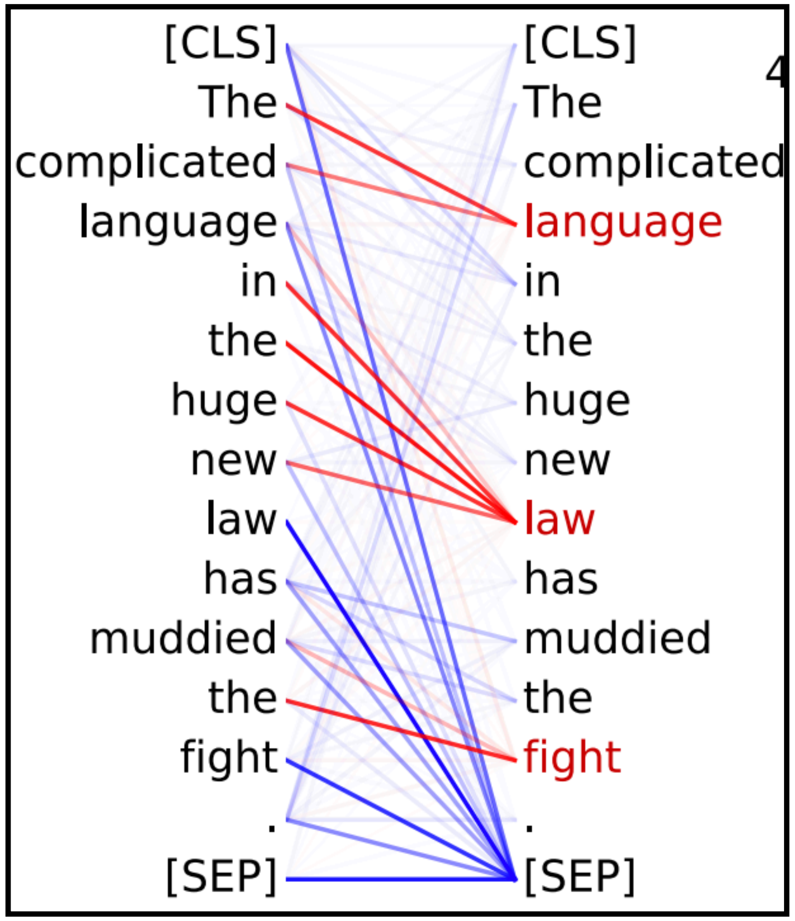
  - Particular heads specialize to specific aspects of syntax. For example, there are heads that find direct objects of verbs, determiners of nouns, objects of prepositions, and objects of possessive pronouns with >75% accuracy.

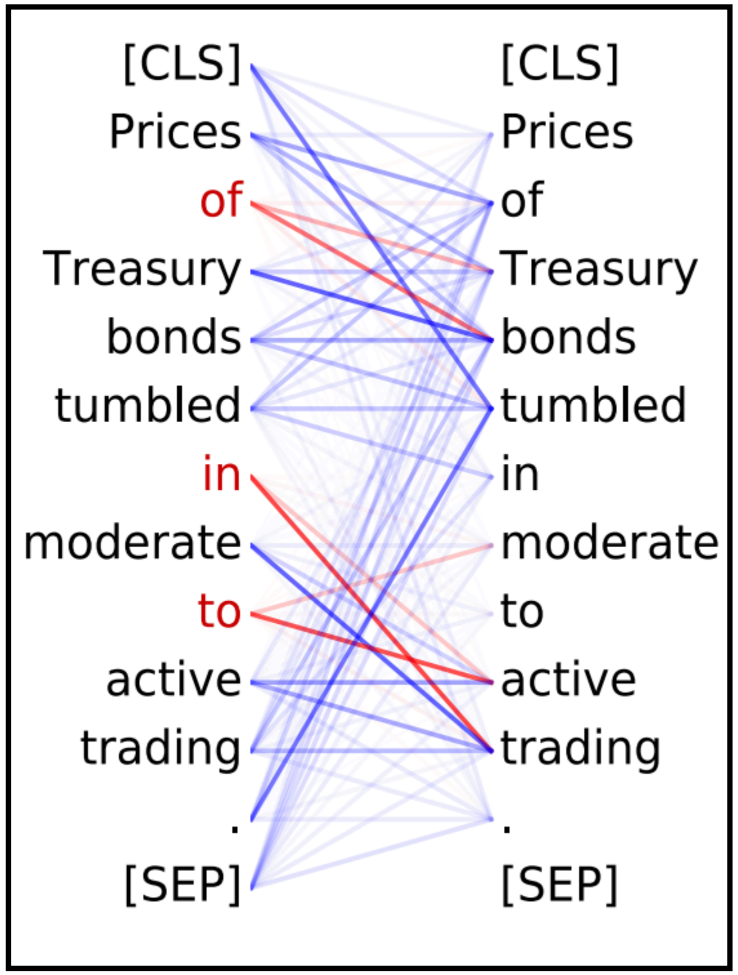
Examples:


[CLS]
Not
his
autograph
;
power-hitter
McGwire
's
.
[SEP]
[CLS]
Not
his
autograph
;
power-hitter
McGwire
's
.
[SEP]


[CLS]
This
market
has
been
very
badly
damaged
.
"
[SEP]
[CLS]
This
market
has
been
very
badly
damaged
.
"
[SEP]


[CLS]
Short-term
interest
rates
fell
yesterday
at
the
government
's
weekly
Treasury
bill
auction
.
[SEP]
[CLS]
Short-term
interest
rates
fell
yesterday
at
the
government
's
weekly
Treasury
bill
auction
.
[SEP]


[CLS]
The
complicated
language
in
the
huge
new
law
has
muddied
the
fight
.
[SEP]
[CLS]
The
complicated
language
in
the
huge
new
law
has
muddied
the
fight
.
[SEP]


[CLS]
Prices
of
Treasury
bonds
tumbled
in
moderate
to
active
trading
.
[SEP]
[CLS]
Prices
of
Treasury
bonds
tumbled
in
moderate
to
active
trading
.
[SEP]

References:

- Attention in NLP: https://medium.com/@joealato/attention-in-nlp-734c6fa9d983
- Attention Is All You Need: https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf
- Attention and its Different Forms: https://towardsdatascience.com/attention-and-its-different-forms-7fc3674d14dc
- What Does BERT Look At? An Analysis of BERT's Attention: https://arxiv.org/pdf/1906.04341.pdf
- Visualizing Attention for Seq2Seq models: https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/
- Attention and Memory in Deep Learning and NLP: http://www.wildml.com/2016/01/attention-and-memory-in-deep-learning-and-nlp/
- An Introductory Survey on Attention Mechanisms in NLP Problems: https://arxiv.org/pdf/1811.05544.pdf
- Hierarchical Attention Networks for Document Classification: http://www.cs.cmu.edu/~./hovy/papers/16HLT-hierarchical-attention-networks.pdf
- REASONING ABOUT ENTAILMENT WITH NEURAL ATTENTION: https://arxiv.org/pdf/1509.06664.pdf
- DYNAMIC COATTENTION NETWORKS FOR QUESTION ANSWERING: https://arxiv.org/pdf/1611.01604.pdf
- Neural Machine Translation by Jointly Learning to Align and Translate: https://arxiv.org/abs/1409.0473
- FRUSTRATINGLY SHORT ATTENTION SPANS IN NEURAL LANGUAGE MODELING: https://arxiv.org/pdf/1702.04521.pdf
- Illustrated: Self-Attention: https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a