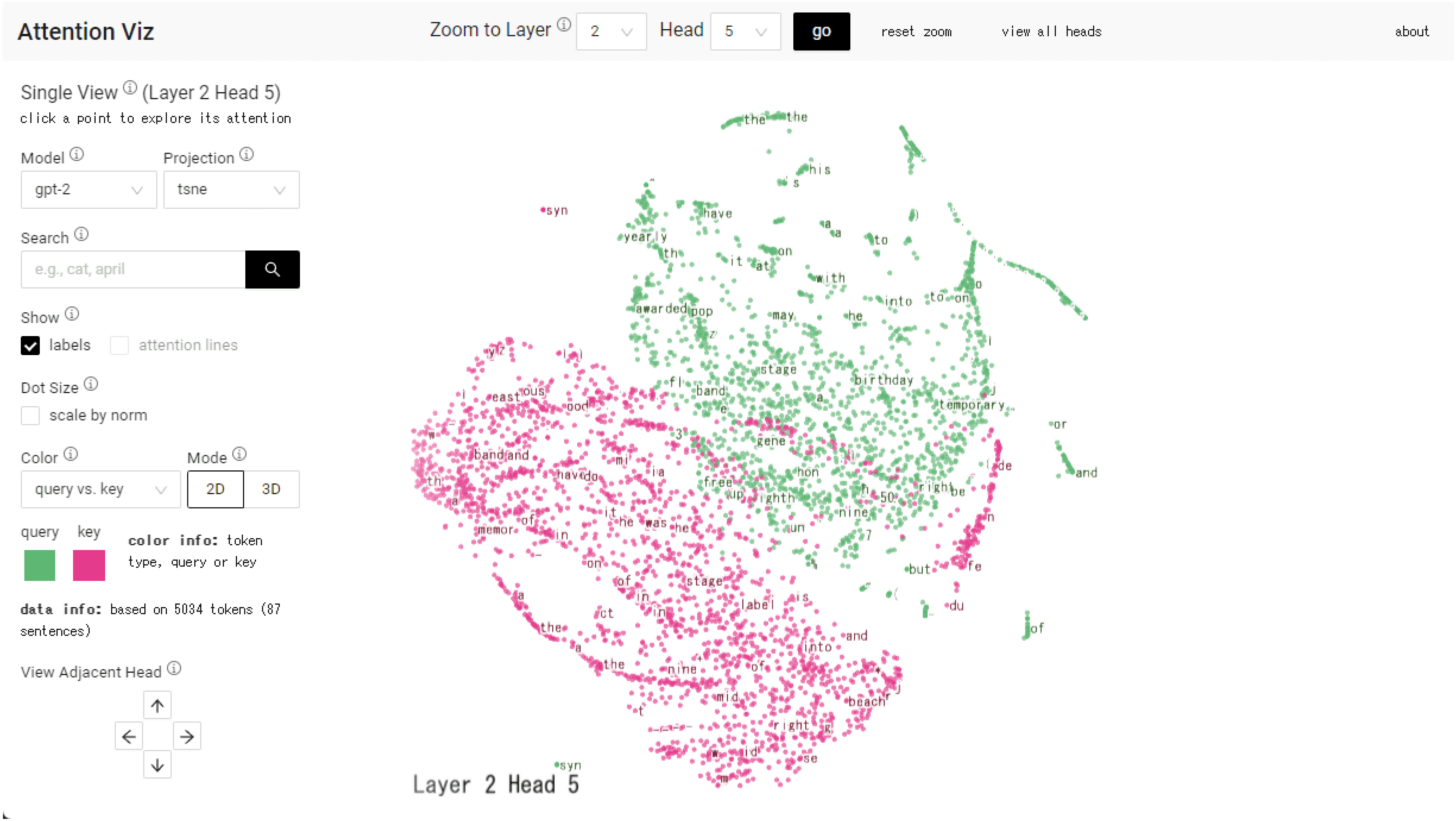
AttentionViz: A Global View of Transformer Attention

Najmieh Sadat Safarabadi

Spring 2023

- AttentionViz, our interactive visualization tool, allows users to explore transformer self-attention at scale by creating a joint embedding space for queries and keys.
- In language transformers, these visualizations reveal striking visual traces that can be linked to attention patterns.
- Each point in the scatterplot represents the query or key version of a word, as denoted by point color. Users can explore individual attention heads (left) or zoom out for a “global” view of attention (right)


Attention Viz
Zoom to Layer 2 Head 5 go
reset zoom
view all heads
about
Single View (Layer 2 Head 5)
click a point to explore its attention
Model
gpt-2
Projection
tsne
Search
e.g., cat, april
Show
labels
attention lines
Dot Size
scale by norm
Color
query vs. key
Mode
2D
3D
query
key
color info: token type, query or key
data info: based on 5034 tokens (87 sentences)
View Adjacent Head
Layer 2 Head 5

- such as attention heads that group image patches by hue and brightness. Here the Border colours denotes query embeddings of a patch (green) or key embeddings (pink).
- Unlike previous attention visualization techniques, this approach enables **the analysis of global patterns across multiple input sequences**.
- This allows these models to learn rich, contextual relationships between elements of a sequence.

- Although attention patterns have been intensively studied, previous techniques generally visualize information related to just a single input sequence (e.g., one sentence or image) at a time.
- Typical approaches create bipartite graph or heatmap representations of attention weights for a given input sequence.


(a) Heatmap of Transformer for example-1


(b) Heatmap of GTrans for example-1


(c) Heatmap of Transformer for example-2


(d) Heatmap of GTrans for example-2

Figure

Caption

Heatmap of the attention layer in Transformer and GTrans for the input of example-1 and example-2. The Transformer always focuses on several fixed positions, and GTrans will pay attention to different parts in each decoding step

- a kind of “attention atlas” that can provide researchers with a rich and detailed view of how a transformer’s various attention heads operate.
- The primary new technique is visualizing a joint embedding of the query and key vectors used by transformers, which creates a **visual signature** for an individual attention head.

One of the Goals

- **AttentionViz** affords exploration through multiple levels of detail providing both a global view to see all **attention heads at once and the ability to zoom in on details in a single attention head or input sequence**
- We find several identifiable "visual traces" linked to attention patterns in BERT, detect novel hue/frequency behavior in ViT's visual attention mechanism, and **uncover potentially anomalous behavior in GPT-2**

- This supports the wider applicability of our approach in visualizing other embeddings at scale.
- A visualization technique for exploring attention trends in transformer models based on **joint query-key embeddings**.

Attention in Transformer Models

- Sequence modelling is a process of representing the input sequence (e.g., words) in a continuous representation and recovering another sequence which semantically maps to the input sequence.
- An example application of such modelling is language modelling and translation tasks.

- The encoder-decoder model provides a pattern for using recurrent neural networks to address challenging sequence-to-sequence prediction problems such as machine translation.
- Attention is an extension to the encoder-decoder model that improves the performance of the approach on longer sequences.

Transformer Architecture

- While self-attention layer is the central mechanism of the Transformer architecture, it is not the whole picture. Transformer architecture is a composite of following parts:

1. Tokenizers convert text to tokens and tokens are mapped to embeddings
2. Positional encodings inject input word-position information
3. Self-attention layer contextually encodes the input sequence information
4. Feed forward layer which operates bit like a static key-value memory. FF layer is similar to self-attention except it does not use softmax and one of the input sequences is a constant.
5. Cross-attention decodes output sequence of different inputs and modalities.

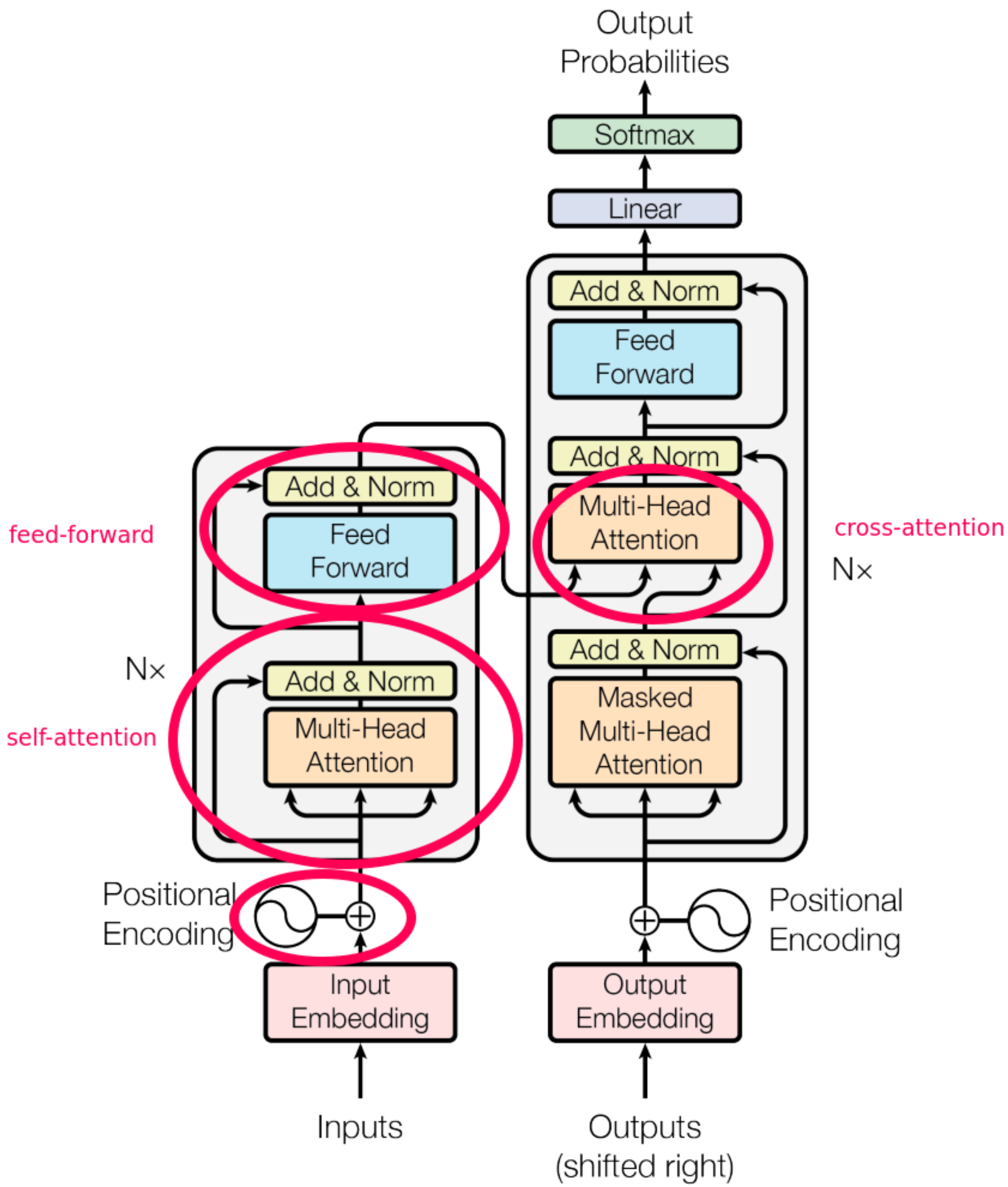


Figure 1: The Transformer - model architecture.

Self-Attention

- Self-Attention is a form of attention in which queries, keys, and values are sampled from the same original word sequence which is input to a transformer model.
- This allows transformer to build semantic word associations and be able infer other words while given a specific word as a query from the sequence.

- attention depends on three terms: *query*, *key* and *value*.
- A query — as the name suggests, is a search word with an intention to find related words in a sequence.
- It is where one wants to draw the attention of the transformer. For example, in the word sequence used earlier, if we choose "*Author*" as a query then we would essentially be looking for all the other words in the sequence that have strong relationship with "*Author*".

- ***softmax* generates a probability distribution from the similarity scores**.
- This **similarity weight matrix** when multiplied with the embedded word vectors **V** would work **as a mask and highlight only those words which have highest similarity** with the respective query vector.

Multi-Head Attention

- A single self-attention mechanism provides a way to model the word associations between **an input and an output sequence**.
- However, it becomes beneficial to use multiple attention modules (called *heads*) in a transformer architecture.
- This means having **multiple layers of the attention matrix bundle** (**Q**,**K**,**V**) along with the respective training weights (**WQ**,**WK**,**WV**)
- This allows better handling of large text and gives **different attention outputs** for different heads.

Multi Head Attention

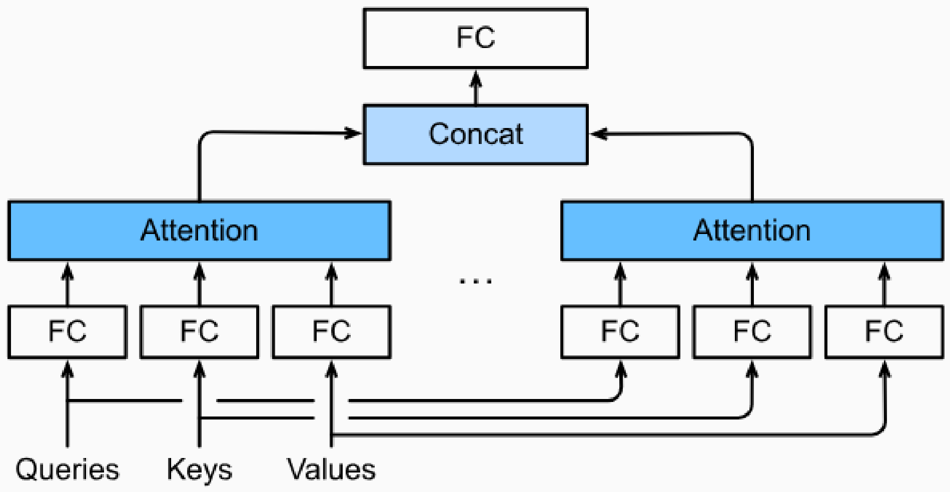
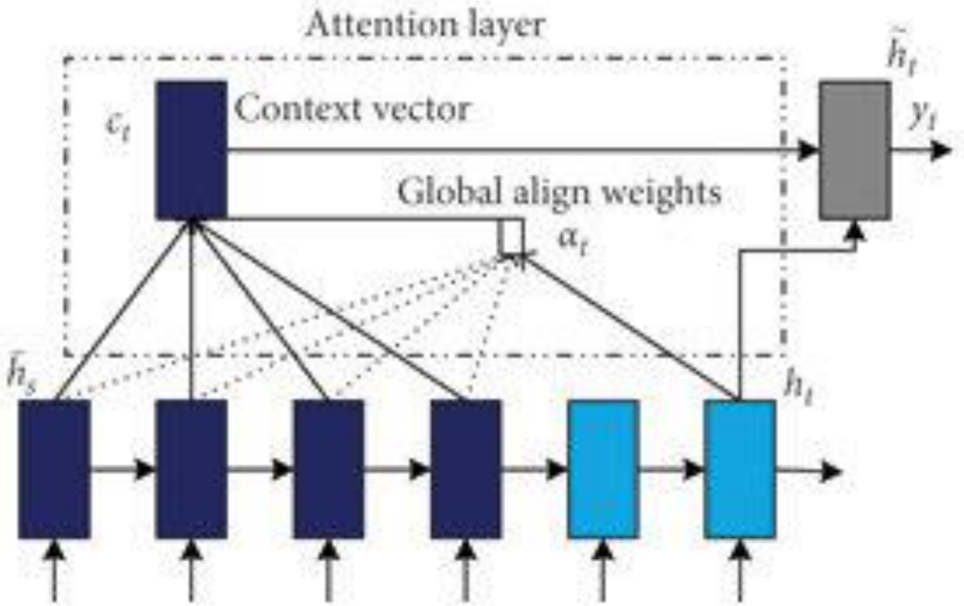
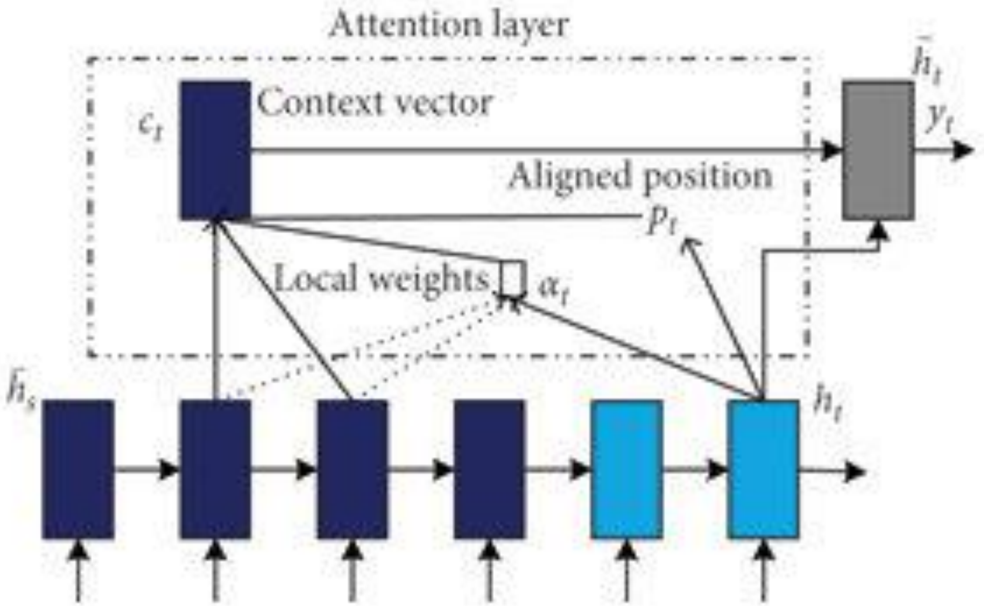


Fig. 11.5.1 Multi-head attention, where multiple heads are concatenated then linearly transformed.

Global Attention in More Details

- **Global Attention**: When attention is placed on all source states. In global attention, we require as many weights as the source sentence length.
- **Local Attention**: When attention is placed on a few source states.
- **Hard Attention**: When attention is placed on only one source state.


Attention layer
$\tilde{h}_t$
c_t
Context vector
y_t
Global align weights
α_t
$\bar{h}_s$
h_t


Attention layer
$\tilde{h}_t$
c_t
Context vector
y_t
Aligned position
p_t
Local weights
α_t
$\bar{h}_s$
h_t

The Problem with transformer!

- The model appeared to be limited on very long sequences. The reason for this was believed to be the fixed-length encoding of the source sequence.
- A potential issue with this encoder-decoder approach is that a neural network needs to be able to compress all the necessary information of a source sentence into a fixed-length vector.

The Longformer

- Transformer-based models are unable to process long sequences due to their self-attention operation, which scales quadratically with the sequence length.
- For example Longformer's attention mechanism is a drop-in replacement for the standard self-attention and combines a local windowed attention with a task motivated global attention.

- In multi-headed attention, **each attention head computes a different attention score**.
- settings with different **dilation configurations** per head improves performance by allowing some heads without dilation to focus on local context, while others with dilation focus on longer context.

The Transformer model computes attention scores as follows:

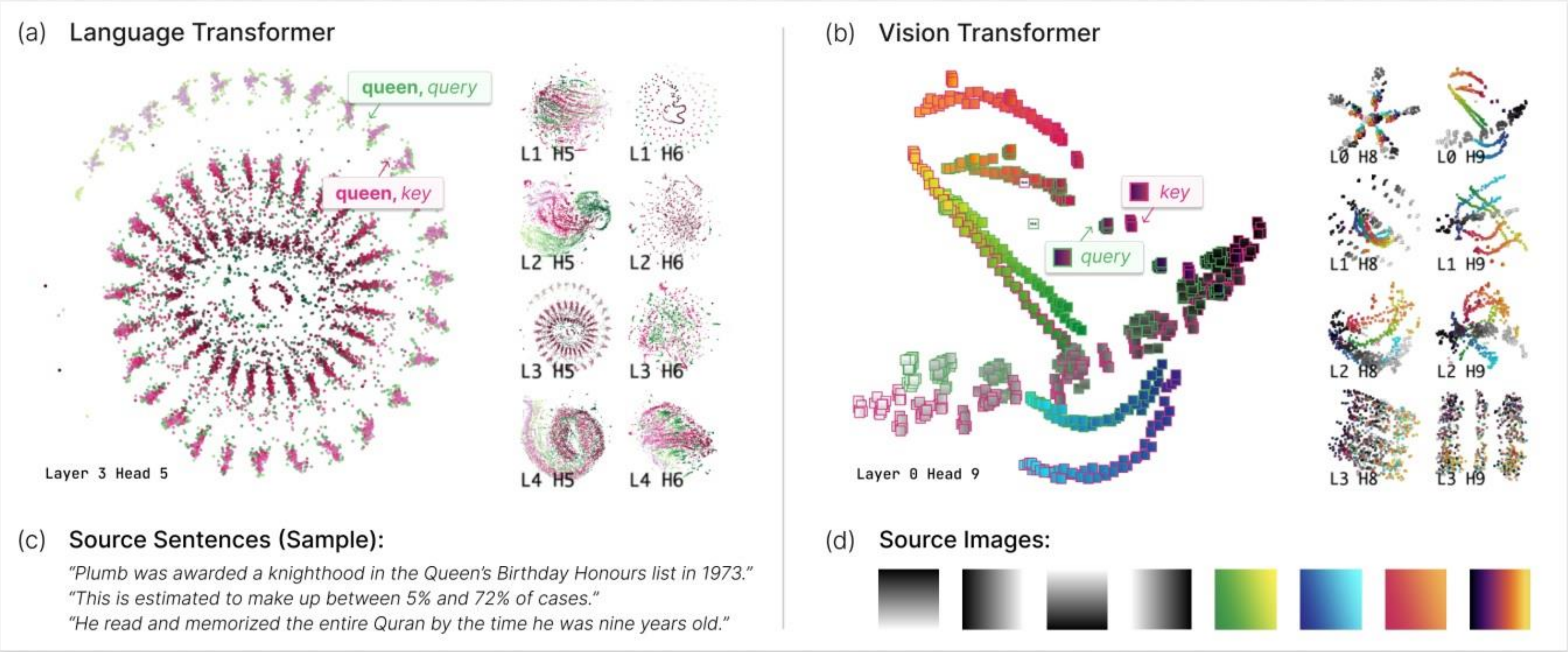
- Take the query vector for a word and calculate it's dot product with the transpose of the key vector of each word in the sequence including itself.
- The dot product between both vectors has zero mean and a variance of d.

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

- We use two sets of projections, Qs, Ks, Vs to compute attention scores of sliding window attention, **and Qg, Kg, Vg to compute attention scores for the global attention**

- In particular, we use small window sizes for the lower layers and increase window sizes as we move to higher layers.
- This allows the top layers to learn higher-level representation of the entire sequence while **having the lower layers capture local information**

- AttentionViz, allows users to explore transformer self-attention at scale **by creating a joint embedding space** for queries and keys.
- these visualizations reveal striking visual traces that can be linked to attention patterns.
- Each point in the scatterplot represents the query or key version of a word, as denoted by point color.


(a) Language Transformer
queen, query
queen, key
Layer 3 Head 5
L1 H5
L1 H6
L2 H5
L2 H6
L3 H5
L3 H6
L4 H5
L4 H6
(b) Vision Transformer
key
query
Layer 0 Head 9
L0 H8
L0 H9
L1 H8
L1 H9
L2 H8
L2 H9
L3 H8
L3 H9
(c) Source Sentences (Sample):
"Plumb was awarded a knighthood in the Queen's Birthday Honours list in 1973."
"This is estimated to make up between 5% and 72% of cases."
"He read and memorized the entire Quran by the time he was nine years old."
(d) Source Images:

- such as attention heads that group image patches by hue and brightness.
- Border colour denotes query embeddings of a patch (green) or key embeddings (pink).
- (c) Sample input sentences and (d) images (synthetic dataset) are provided for reference

- **Attention layers determine which pairs should interact, and what information should flow** between them.
- The **self-attention mechanism allows transformers to learn and use a rich set of relationships between elements** of a sequence, yielding significant performance improvements across various NLP and computer vision tasks.

- For instance, in our example sentence, “brown” and “capyba” are linked by an adjective-noun relation, while “capyba” and “is” form a subject-verb relation.
- To allow for several relation types, transformer attention layers consist of multiple attention heads, each of which can represent a different pattern of attention and information flow

- **Each attention head computes its own attention pattern using a bilinear form** computed from a query weight matrix WQ and key weight matrix WK.
- Concretely, for two embedding vectors x and y, attention f(x, y) is determined by the inner product of a query vector, WQx, and a key vector, WKy. Letting d be the dimension of WKy, we have:

$$f(x,y) = \frac{1}{\sqrt{d}} \langle W_Q x, W_K y \rangle$$

- Given embedding vectors $\{x_1, x_2, \ldots, x_n\}$ **<u>we compute the attention between xi and the other vectors using</u>** the SoftMax function:

$$attn(x_i, x_j) = \text{softmax}_j(f(x_i, x), \ldots, f(x_i, x_n)) = e^{f(x_i, x_j)} / \sum_k e^{f(x_i, x_k)}$$

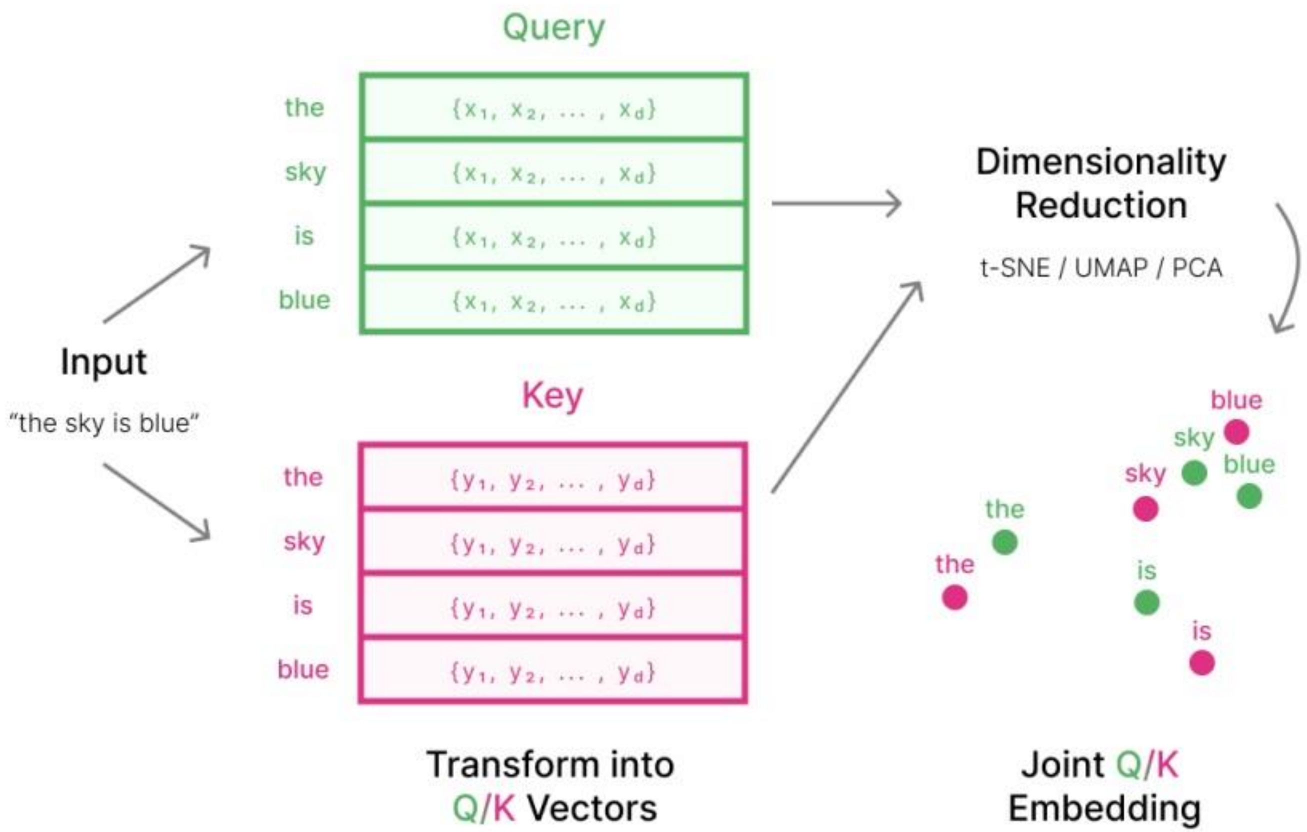
Beyond Single Inputs: Visualizing Embeddings and Activation Maximization

- This visual representation is for three transformer models: BERT (language), GPT-2 (language), and ViT (vision).
- BERT, or Bidirectional Encoder Representations from Transformers is a multi-layer transformer encoder. As a bidirectional model, BERT can attend to tokens (i.e., input elements) in either direction.

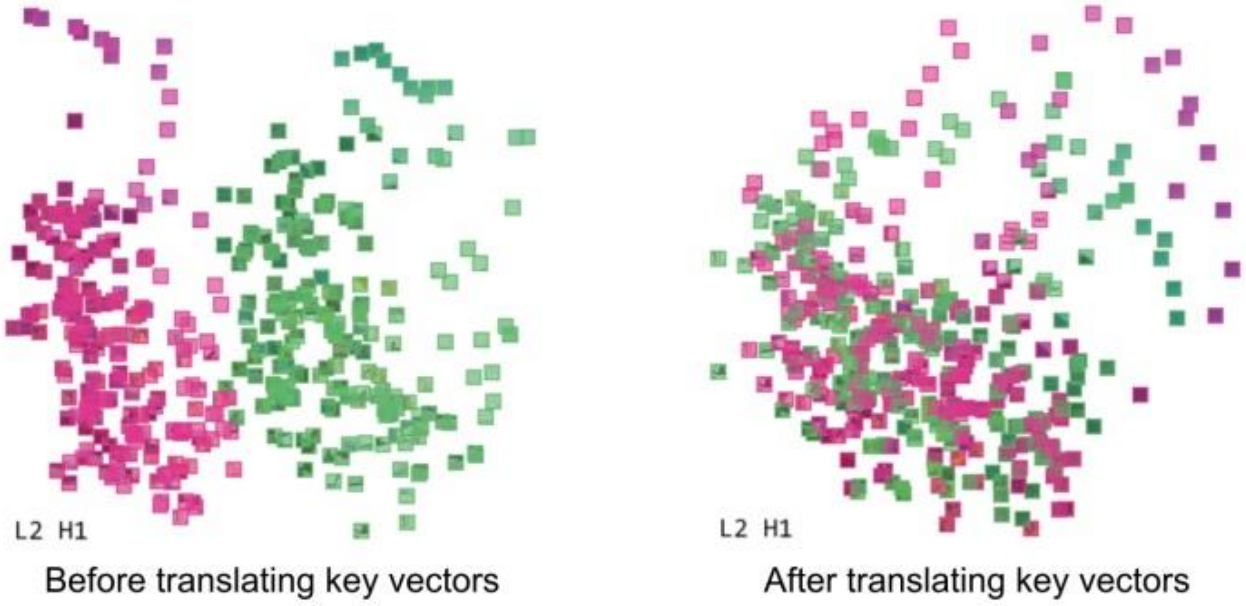
Goals

- we design a global matrix view to visualize query-key embeddings.
- **Helps in model interpretability.**
- to better understand the behavior of different attention heads and **what transformer models are learning through their characteristic self-attention mechanism**. Thus, they expressed the desire to be able to quickly and easily explore attention patterns.
- **could provide insights into “why large language models fail at reasoning tasks and math,”** for example.

- In the NLP case, given an input sentence, we first transform each token into its corresponding query and key vector.
- Then, we use tSNE/UMAP/PCA **to project these 1×d vectors into 2D/3D scatterplot coordinates**.
- G3 - Identify attention anomalies. Four researchers (E2-5) **wanted to identify irregularities and potential behavioural issues with transformers through attention pattern exploration.**



Input
"the sky is blue"
Query
the
sky
is
blue
{x1, x2, ... , xd}
Key
{y1, y2, ... , yd}
Transform into
Q/K Vectors
Dimensionality
Reduction
t-SNE / UMAP / PCA
Joint Q/K
Embedding



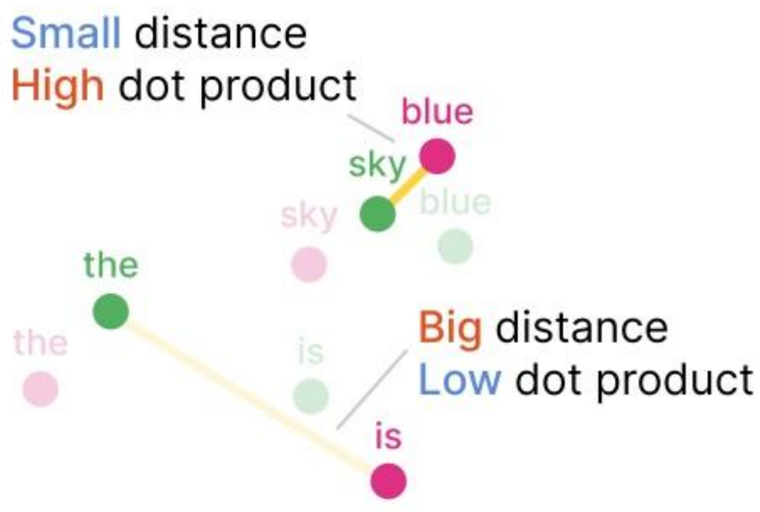
Left: original queries and keys in joint embedding space.
Right: Increased overlap after translating keys to align query and key centroids.

visually, it means queries might be a tiny cluster, surrounded by a loose cloud of keys.

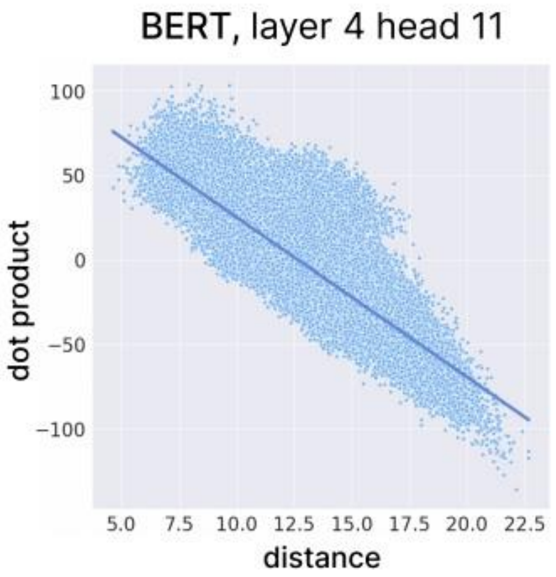
- A central observation is that the relative positions of query and key vectors can **offer clues about how attention will be distributed, since attention coefficients depend on the dot product between queries and keys.**
- To see why, consider a hypothetical situation where query and key vectors always have the same norm.

Distance as a Proxy for Attention

- **expecting distance to be inversely correlated with attention** in our joint query-key embeddings.
- Across multiple datasets and models, the relationship between distance and attention holds fairly well.
- For example, with Wiki-Auto data, the mean correlation between query-key distances and dot products is -0.938 for BERT and -0.792 for GPT



(a)



(b)

Fig. 4:

(a) Ideal distance-attention relationship, where query-key pairs with higher dot products are closer in the joint embedding space.

(b) Example attention head with a strong, negative correlation (-0.983) between query-key distance and dot product in BERT.

Color Encodings

- For language transformers, we support two positional colour schemes: normalized and discrete.
- To compute normalized position, it divide each token’s position in a sentence by the sentence length to produce a continuous colour scale.
- **Lighter hues** denote tokens closer to the beginning of the sentence
- We use the the same five colours to encode queries and keys at different positions, **using darker hues for the former**.

Matrix View

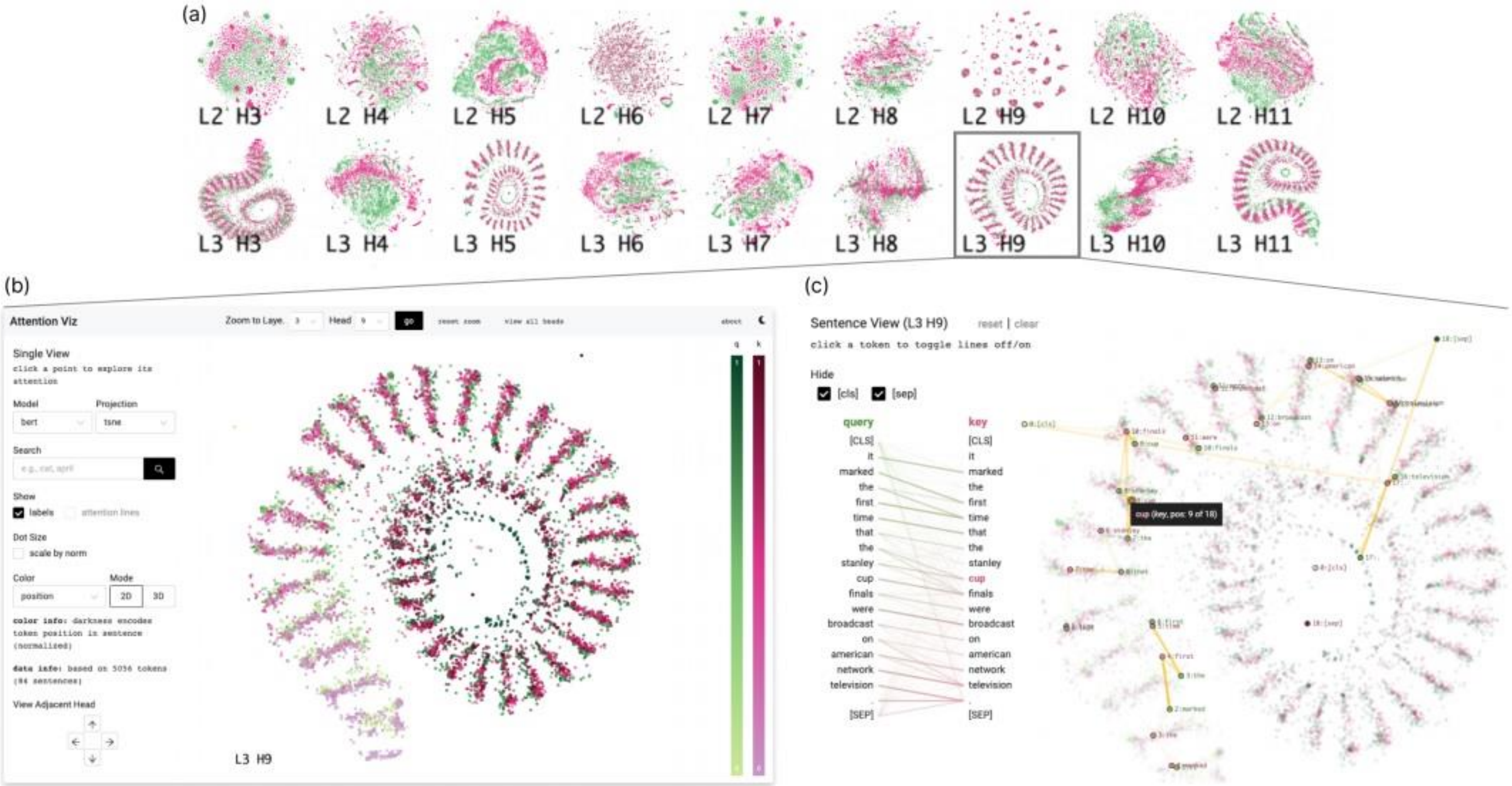


Fig. 5: Connecting form to function in BERT. **(a)** In Matrix View, there are several spiral-shaped plots in layer 3. **(b)** By zooming into one such head (*L3 H9*) using Single View, we can see positional attention patterns by using a light-to-dark color scheme that encodes position in the input sequence. **(c)** These patterns can be confirmed by exploring sentence-level visualizations.

- Sentence View.
- The opacity of the lines connecting query tokens in the left column and key tokens in the right column signifies their corresponding attention strength. Hovering on a token highlights token-specific attention lines.
- Users have the option of viewing the aggregate attention pattern for each attention head as well, to offer another layer of comparison

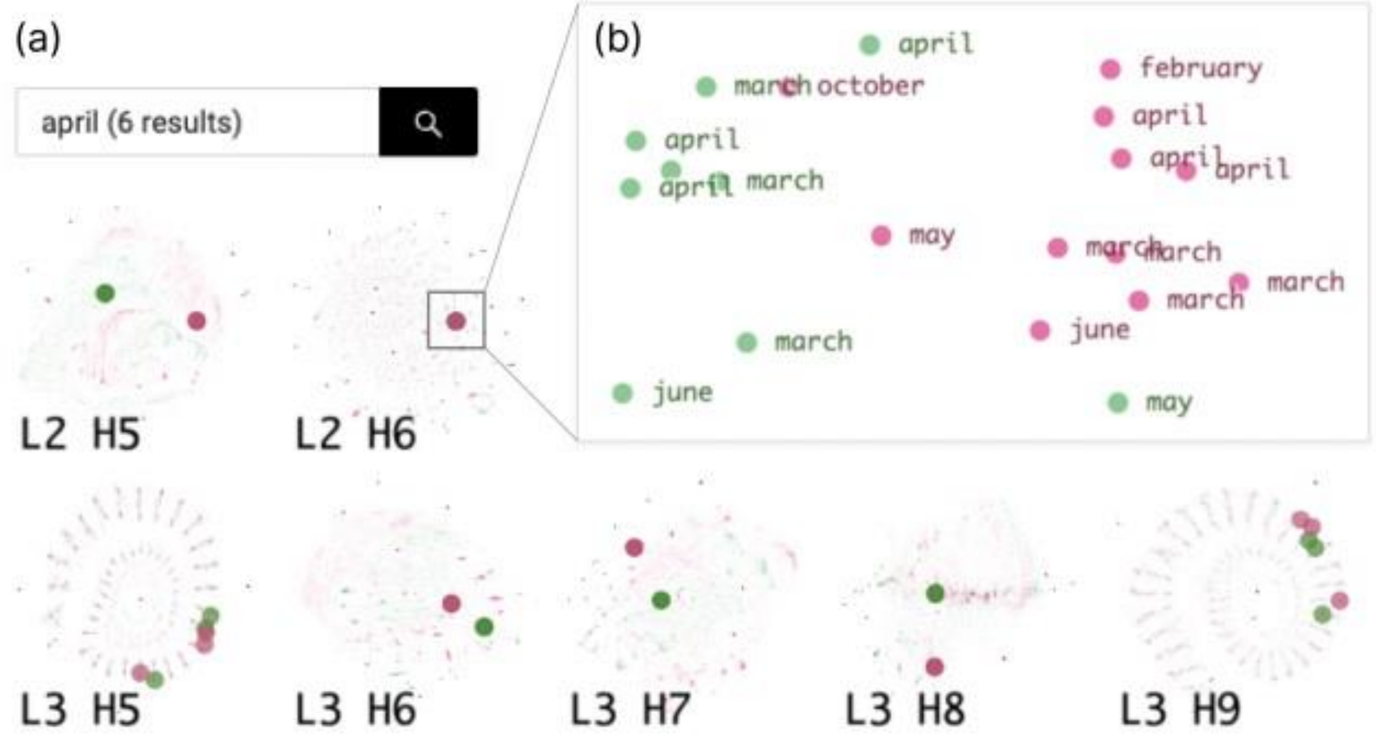


Fig. 6: Exploring attention patterns with global search. **(a)** Heads with fewer clusters of search results often demonstrate more semantic behavior, while heads with dispersed results focus more on token position. **(b)** Zooming into L2 H6, a head with one main result cluster, we indeed see a large group of semantically related query and key tokens.

Goal: Identifying Unexpected Behaviour

Global search patterns. The aggregate search feature in Matrix View can also be used to quickly scan for and compare attention trends across heads .

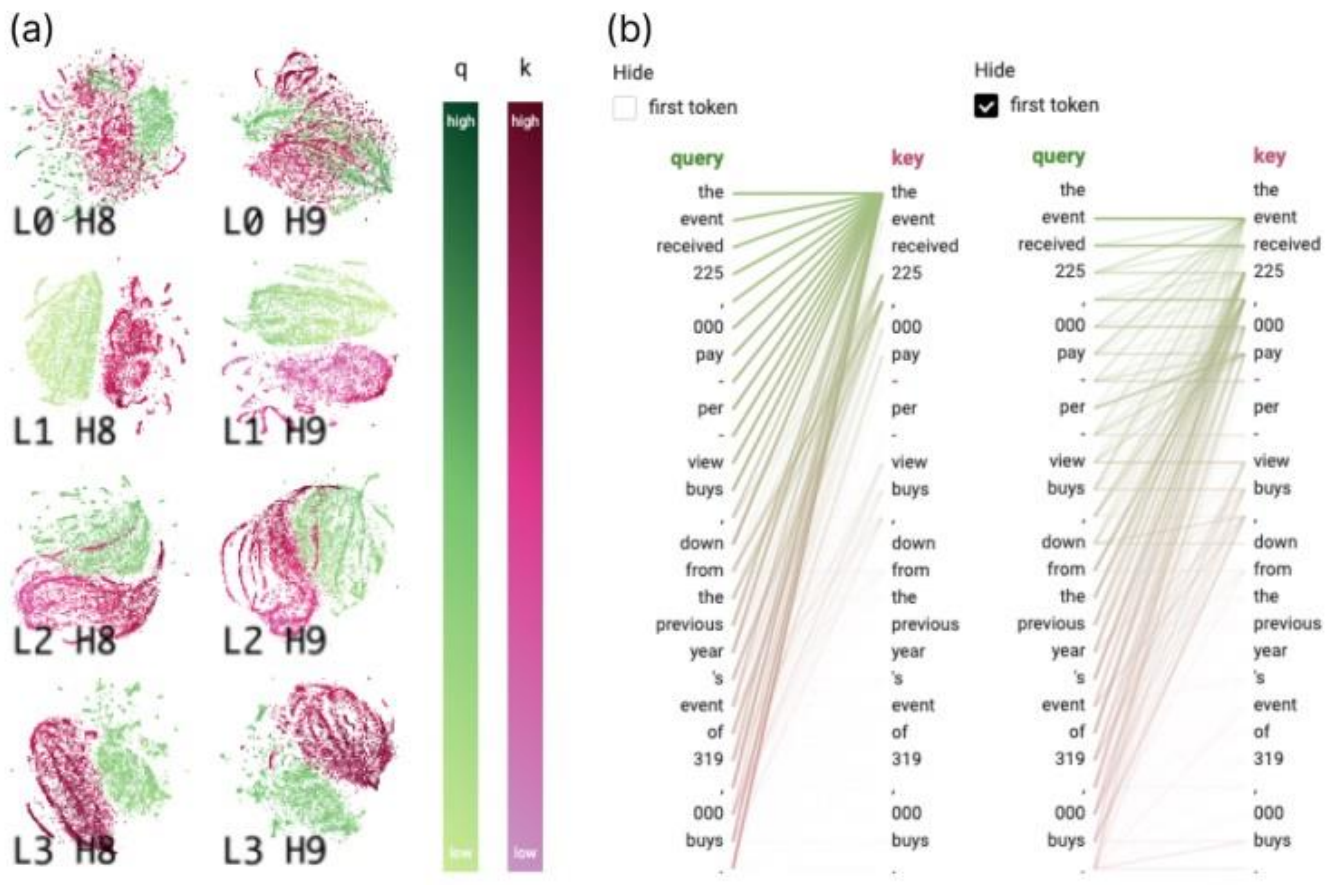


Fig. 12: Anomalies in GPT-2. **(a)** In early model layers, we witness a significant disparity between query-key norms for many attention heads (e.g., *L1 H8* prior to norm scaling). **(b)** Example of the prevalent "attend to first" pattern in later layers. Sentence View reveals latent attention behavior after hiding the first token.

- Norm disparities and null attention. While exploring GPT-2 in Matrix View, we observed that in early model layers, some query and key clusters were well-separated, even after key translation
- We also noticed that in many GPT-2 heads, most attention is directed to the first token (Fig. 12b), especially in later layers.
- briefly mentions that the first token is treated as a null position for attention receiving in GPT-2 "when the linguistic property captured by the attention head doesn't appear in the input text."

The Demo

https://attentionviz.com/

References

1. AttentionViz: A Global View of Transformer Attention (Catherine Yeh, Yida Chen, Aoyu Wu, Cynthia Chen, Fernanda Viégas, and Martin Wattenberg)
2. Longformer: The Long-Document Transformer (Iz Beltagy, Matthew E. Peters, Arman Cohan)
3. https://towardsdatascience.com/matters-of-attention-what-is-attention-and-how-to-compute-attention-in-a-transformer-model-4cbbd3250307
4. https://www.kdnuggets.com/2021/01/attention-mechanism-deep-learning-explained.html
5. https://attentionviz.com/
6. https://d2l.ai/chapter_attention-mechanisms-and-transformers/multihead-attention.html

THANK YOU