ECE598LV Implementation: Hacking How Transformers Understand A Sentence Using BertViz Tool

Zong Fan
Dept. Bioengineering
zongfan2@illinois.edu

May 5, 2022

1 Introduction

Since the Bidirectional Encoder Representations from Transformers (BERT) [1] achieved the state-of-art performance across all kinds of natural language processing (NLP) tasks, Transformer [3]-based methods seem to change the ways of dealing with NLP problems entirely. Nowadays, Transformer-based methods have dominated the top rankings in the General Language Understanding Evaluation (GLUE) benchmark (see the leaderboard). Unlike traditional recurrent networks, the Transformer model is designed based solely on attention mechanisms without any use of recurrence and convolution [3]. It makes the Transformer model easily to be parallelized, which significantly improves the computation efficiency thus enabling model pre-training on extremely large-scale corpus datasets. As shown in Fig. 1, the Transformer uses self-attention, which consists of 2 sub-modules, the scaled dot-product attention and the multi-head attention. They are working together to determine which words of a sentence the transformer should focus on.

Particular, BERT[1] and Generative Pre-trained Transformer (GPT) [2] models are the two kinds of most popular Transformer-based model. One big distinction between them also lies in the attention module. The BERT employs the self-attention module that processes bidirectional representation, allowing reading text from left-to-right and right-to-left, while the GPT employs the masked self-attention module for unidirectional processing, as shown in Fig. 2. Another difference is that the BERT uses the Transformer's encoder segment, while the GPT uses the decoder segment.

Using attention gives us a way to see how the model attends to different parts of the input sentence, largely improving the model interpretability for decision making. Many tools have been developed to visualize the attention of the Transformer-based model. In this implementation, an open-source tool called BertViz [4] was employed for model visualization from three perspectives: attention-head view, model view, and neuron view. A BERT and A GPT pre-trained model were investigated which were tested on the same sentence for comparison.

2 Method

2.1 Model

A BERT and A GPT model released in the Hugging Face model zoo were used: **BERT**: "bert-base-uncased"; **GPT-2**: "gpt2"

2.2 Testing data

A sentence: *"this bird cannot fly in the sky, because it is too heavy"*. In this sentence, "it" indicates the bird.

2.3 Attention head view

The attention-head view function provided by BertViz is to visualize the attention patterns produced by one or more attention heads in a given transformer layer. In this view, self-attention is represented as lines

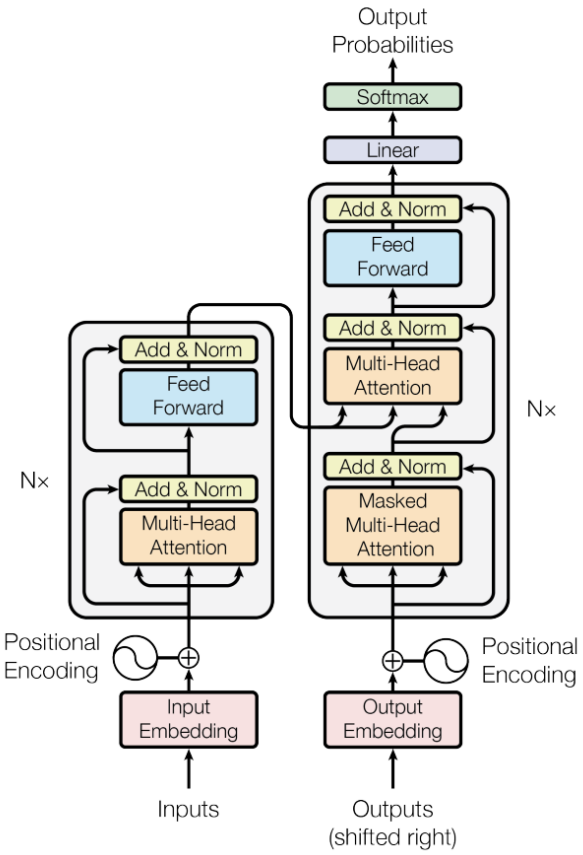


Figure 1: Transformer architecture. Image credit: Vaswani *et al.*, 2017 [3]

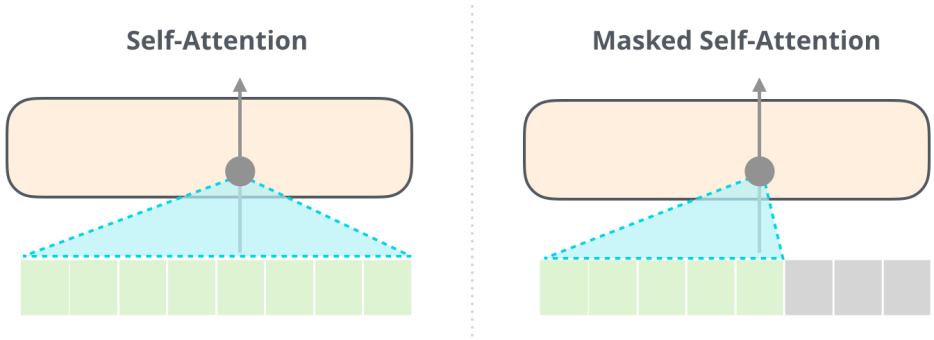


Figure 2: Self-attention in BERT vs masked self-attention in GPT. Image credit: Jay Alammar https://jalammar.github.io/illustrated-gpt2/

connecting the tokens that are attending (left) with the tokens being attended to (right). Colors represent the corresponding attention heads, while line weight reflects the attention score.

As shown in Fig 3, each head in the GPT-2 can produce a distinct attention pattern, since the attention heads do not share parameters. The top left figure shows all the tokens attend the first word "this" in the attention head of layer 7, while the head in the top right figure seems to generate attention dispersed evenly across previous words. The former pattern seems to be a null pattern that is produced when the linguistic property captured by the attention head doesn't appear in the input text. It would make the model more interpretable by disentangling the null attention from attention related to the first token. The bottom two

figures show the lexical pair patterns captured by the GPT-2. For example, $fly \leftrightarrow bird$ and $heavy \leftrightarrow bird$ show high attention, which correctly describes the relationship between behavior or characteristic and the entity. Understanding this relationship might help the model understand the reasoning process that why the bird cannot fly.

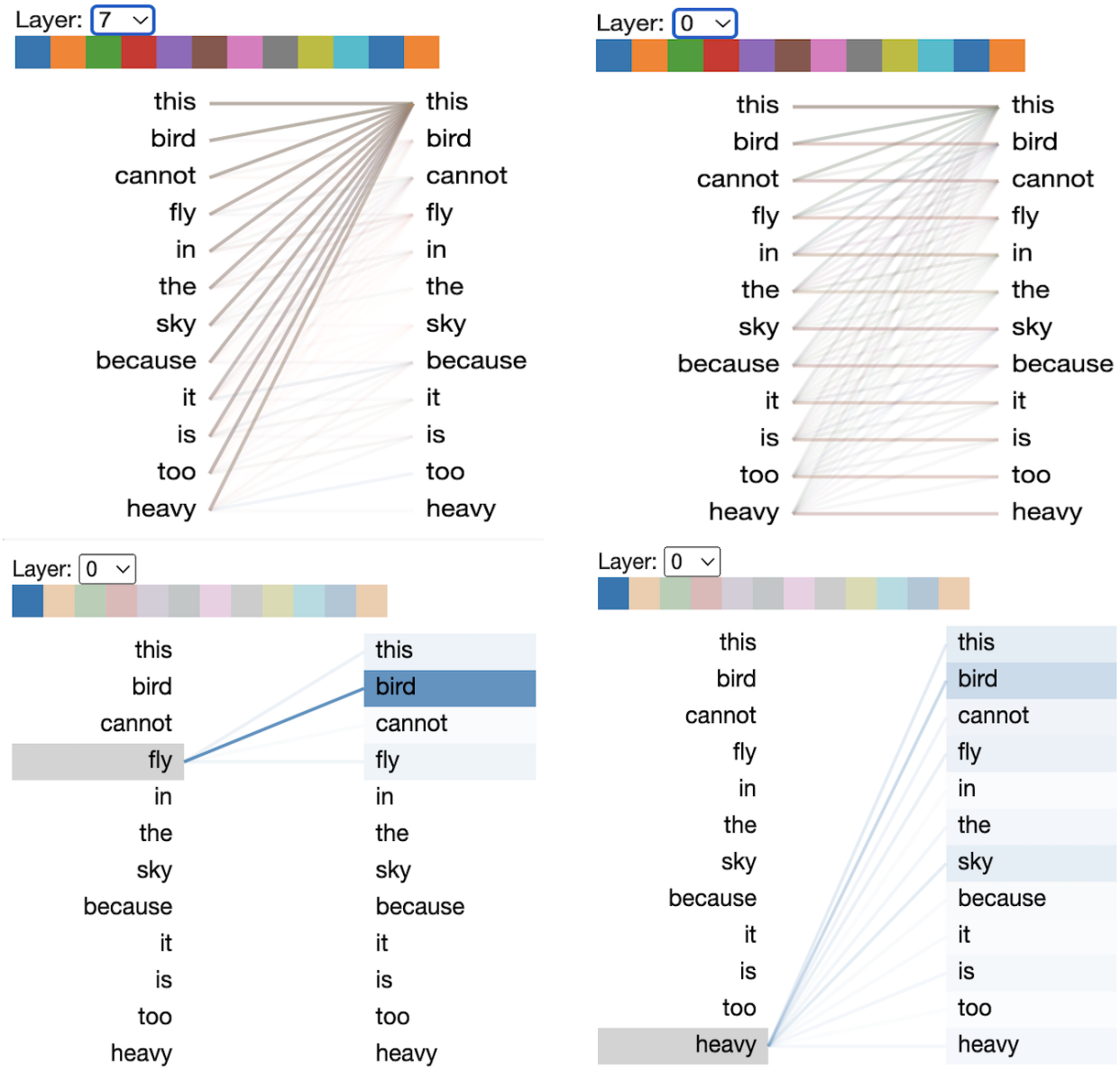


Figure 3: Attention head view of GPT-2 on the sample text. Top left and right images represent the attention head of layer 7 and layer 0, respectively. center figures represent different layers. The bottom left and right images represent the attentions of tokens "fly" and "heavy" on the layer 0 ahd head 0.

For comparison, the head attentions in the BERT model were investigated, as shown in Fig 4. We can clearly see the difference between the unidirectional representation in GPT-2 and the bidirectional representation in BERT. In addition, a delimiter-focused attention[4] pattern is observed in the top left figure, where most attention of tokens attend the sentence separator. Such a pattern means that an attention head can't find anything else in the input sentence to focus on, showing that BERT has designated a small set of neurons focusing on [SEP]. The bottom two figures show the bag-of-words attention pattern [4], where the attention is divided evenly across all words in the same sentence separated by the [SEP] token. It

implies that the BERT might capture the sentence-specific meaning by focusing computing a bag-of-words embedding in the same sentence. All these results showed the distinct attention patterns acquired by the GPT-2 and BERT.

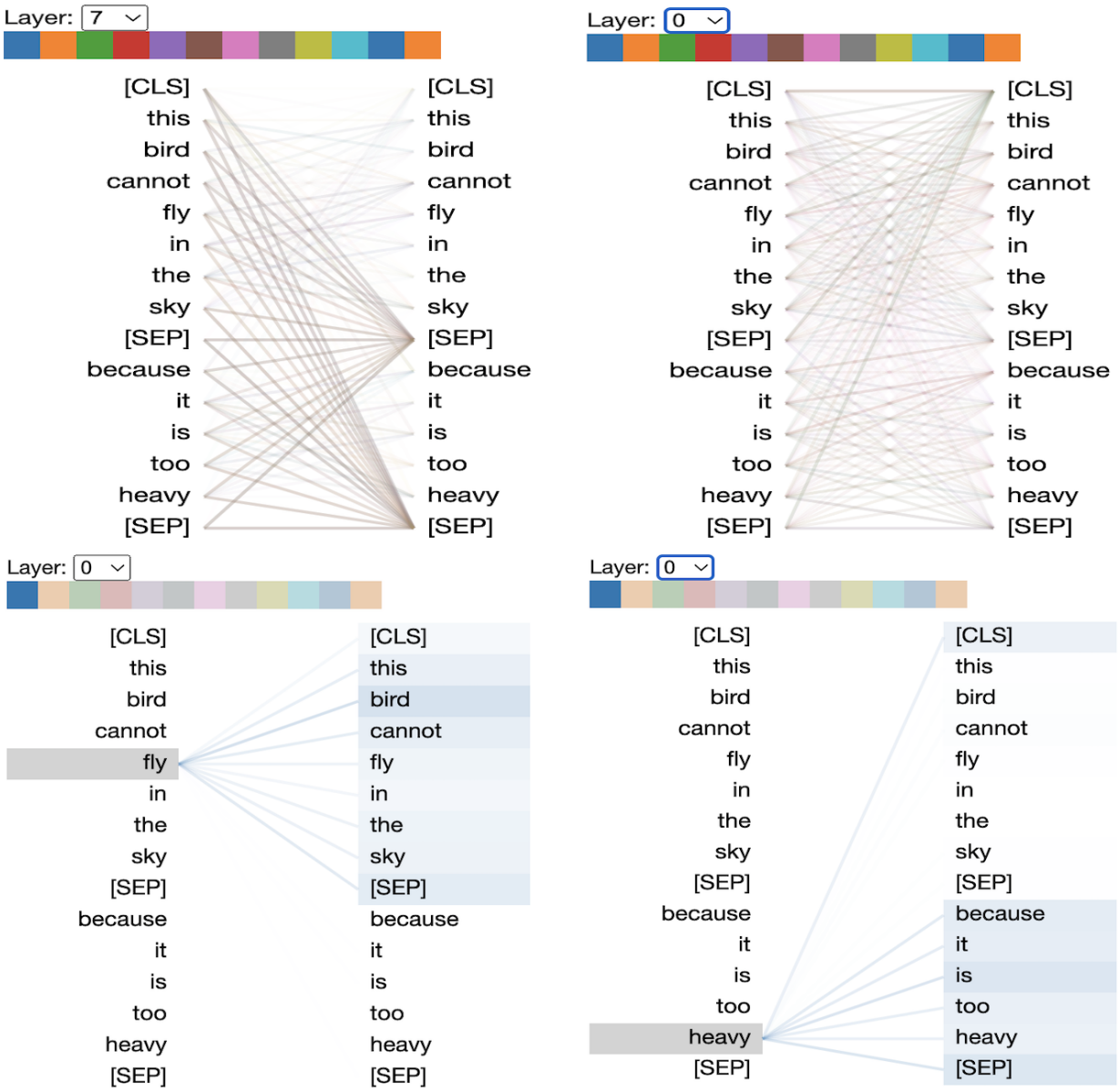
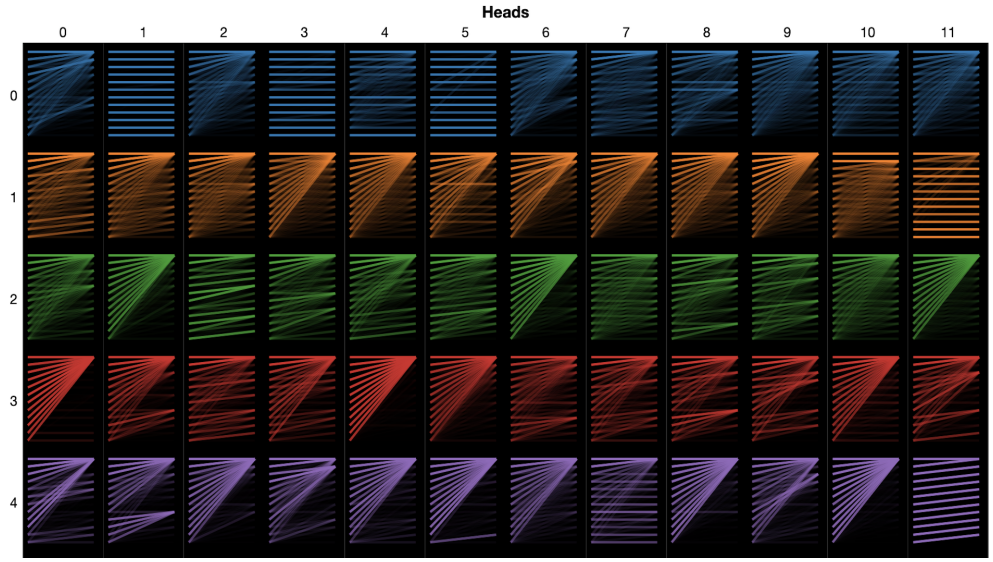


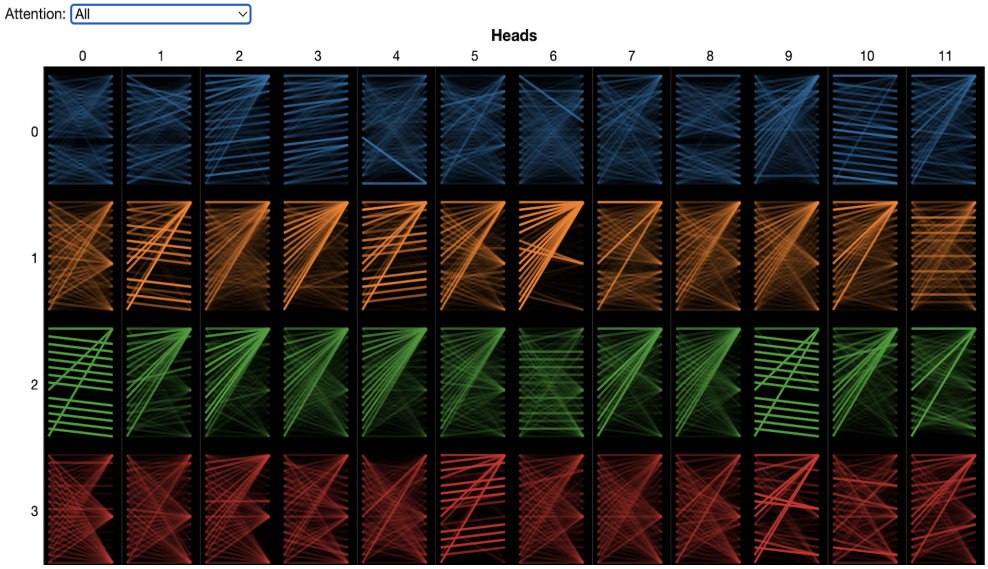
Figure 4: Attention head view of BERT on the sample text. Each figure corresponds to those of GPT-2's. [SEP] means the sentence separator; [CLS] means the beginning of the sentence.

2.4 Model view

The model view provides a bird's-eye view of attention across all of the layers and heads of the model for a particular input, as shown in Fig. 5. It enables us to check the attention heads across all layers and see how attention patterns evolve throughout the model. For example, we can observe that each token seems to focus on the same token (layer 0, head 1) or on the previous token (layer 1, head 0). Similar patterns described in the section 2.3 are easily identified, like delimiter-focused attention (layer 3, head 0) and bag-of-words attention (layer0, head 0).



(a) GPT-2



(b) BERT

Figure 5: Model view samples of GPT-2 and BERT, showing first 4 layers and all 12 heads. Attention heads are presented in a Tabular form, where rows represents the layers and columns represent the heads.

Of course, the attention head of a specific layer and head could be enlarged for checking the detailed tokens, as shown in Fig. 6. The similar patterns discussed above can be observed.

2.5 Neuron view

The neuron view visualizes the individual neurons in the query and key vectors and shows how they interact to produce attention scores, providing the internal explanation of a particular attention head and showing how BERT forms these patterns. Given a selected token, it traces the computation of attention from that token to the other tokens in the sequence. The element-wise product of the vectors shows how individual neurons contribute to the attention The token "heavy" and "fly" discussed above are also visualized in Fig. 7. For example, the bottom left figure shows a bag-of-words attention pattern, where the element-wise products are positive for tokens in the same sentence and negative for tokens in different sentences. The reason is

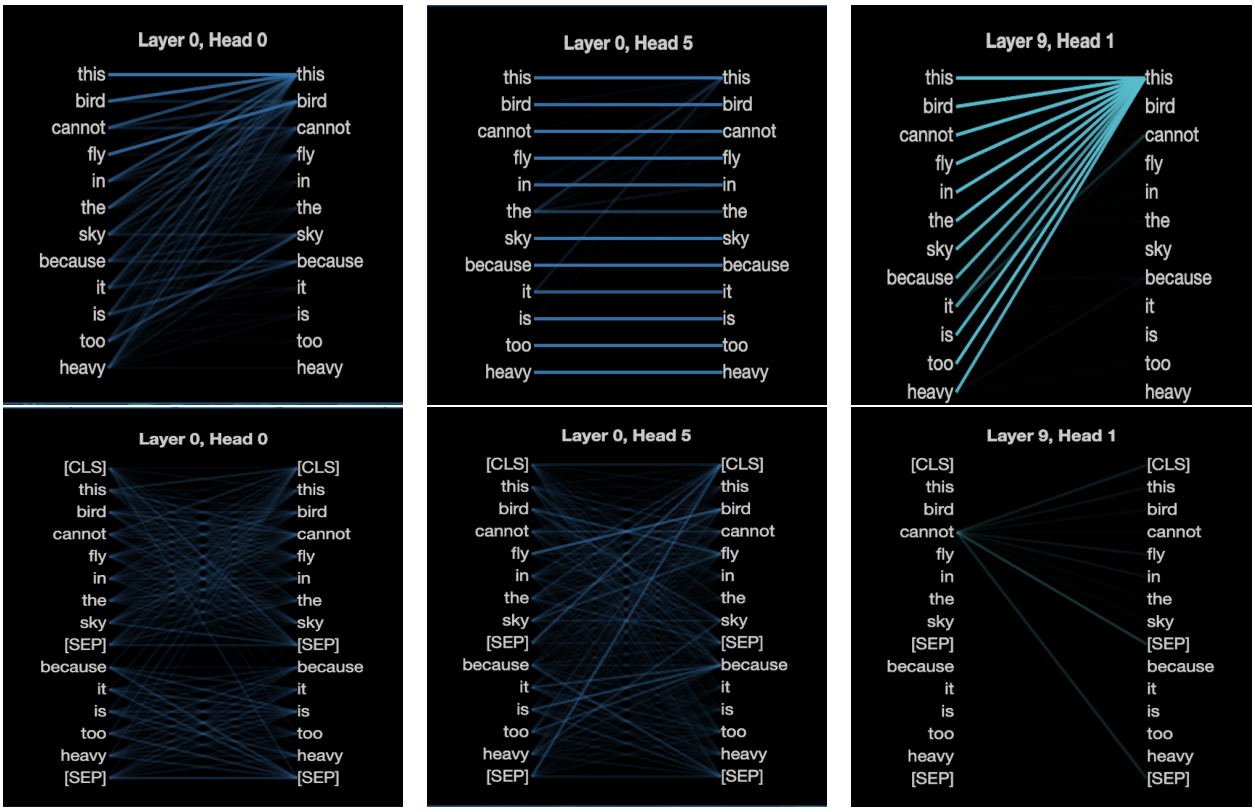


Figure 6: Attention head view particular layer and head of GPT-2 (top row) and BERT (bottom row) in the model view.

that the corresponding query and key neurons have high-magnitude values of the same sign for tokens in the same sentence, but of opposite sign for tokens in the opposite sentence. When specific neurons are linked to a tangible outcome, it implies that human intervention might be introduced into the model to alter the values of the relevant neurons, showing the benefit of model interpretability.

3 Conclusion

In this implementation, we employed the BertViz tool for visualizing attention in Transformer-based language models. By analyzing a sample text sentence on both BERT and GPT-2, we found attention visualization could provide insights into how the language models try to understand the sentence. In this way, hacking into the language model could give us better model interpretability and potential controllability.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[2] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[4] J. Vig. Bertviz: A tool for visualizing multihead self-attention in the bert model. In *ICLR Workshop: Debugging Machine Learning Models*, 2019.

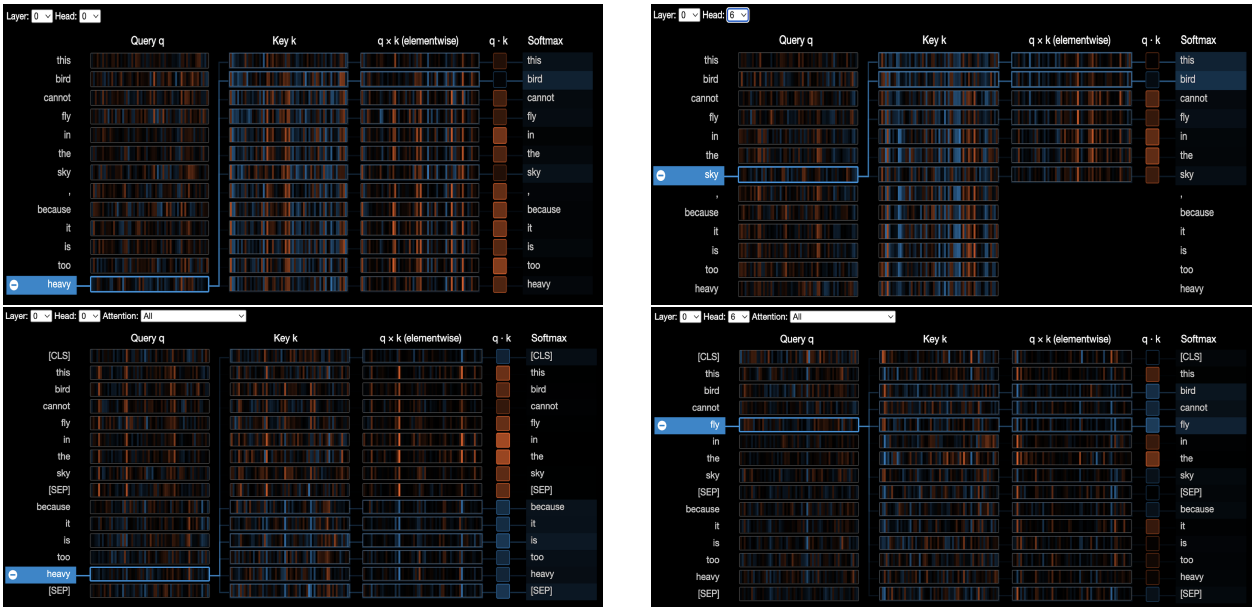


Figure 7: Neuron view examples of GPT-2 (top row) and BERT (bottom row). Query q: The 64-element query vector of the selected token paying attention; Key k: The 64-element key vector of each token receiving attention; $q \times k$ (element-wise): The element-wise product of the selected token's query vector and each key vector; $q\dot{k}$: The dot product of the selected token's query vector and each key vector; Softmax: The softmax of the scaled dot-product from previous column. This equals the attention received by the corresponding token. Positive and negative values are colored blue and orange, respectively, with color saturation showing the value magnitude.