A Multiscale Visualization of Attention in the Transformer Model

Jesse Vig
Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304
jesse.vig@parc.com

Abstract

The Transformer is a sequence model that forgoes traditional recurrent architectures in favor of a fully attention-based approach. Besides improving performance, an advantage of using attention is that it can also help to interpret a model by showing how the model assigns weight to different input elements. However, the multi-layer, multi-head attention mechanism in the Transformer model can be difficult to decipher. To make the model more accessible, we introduce an open-source tool that visualizes attention at multiple scales, each of which provides a unique perspective on the attention mechanism. We demonstrate the tool on BERT and OpenAI GPT-2 and present three example use cases: detecting model bias, locating relevant attention heads, and linking neurons to model behavior.

1 Introduction

In 2018, the BERT (Bidirectional Encoder Representations from Transformers) language representation model achieved state-of-the-art performance across NLP tasks ranging from sentiment analysis to question answering (Devlin et al., 2018). Recently, the OpenAI GPT-2 (Generative Pretrained Transformer-2) model outperformed other models on several language modeling benchmarks in a zero-shot setting (Radford et al., 2019).

Underlying BERT and GPT-2 is the Transformer model, which uses a fully attention-based approach in contrast to traditional sequence models based on recurrent architectures (Vaswani et al., 2017). An advantage of using attention is that it can help interpret a model by showing how the model assigns weight to different input elements (Bahdanau et al., 2015; Belinkov and Glass, 2019), although its value in explaining individual predictions may be limited (Jain and Wallace, 2019). Various tools have been developed to visualize attention in NLP models, ranging from attention-matrix heatmaps (Bahdanau et al., 2015; Rush et al., 2015; Rocktäschel et al., 2016) to bipartite graph representations (Liu et al., 2018; Lee et al., 2017; Strobelt et al., 2018).

One challenge for visualizing attention in the Transformer is that it uses a multi-layer, multi-head attention mechanism, which produces different attention patterns for each layer and head. BERT-Large, for example, which has 24 layers and 16 heads, generates $24 \times 16 = 384$ unique attention structures for each input. Jones (2017) designed a visualization tool specifically for multi-head attention, which visualizes attention over multiple heads in a layer by superimposing their attention patterns (Vaswani et al., 2017, 2018).

In this paper, we extend the work of Jones (2017) by visualizing attention in the Transformer at multiple scales. We introduce a high-level *model view*, which visualizes all of the layers and attention heads in a single interface, and a low-level *neuron view*, which shows how individual neurons interact to produce attention. We also adapt the tool from the original encoder-decoder implementation to the decoder-only GPT-2 model and the encoder-only BERT model.

2 Visualization Tool

We now present a multiscale visualization tool for the Transformer model, available at https://github.com/jessevig/bertviz. The tool comprises three views: an attention-head view, a model view, and a neuron view. Below, we describe these views and demonstrate them on the GPT-2 and BERT models. We also present three use cases: detecting model bias, locating relevant attention heads, and linking neurons to model behavior. A video demonstration of the tool can be found at https://vimeo.com/340841955.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 37–42
Florence, Italy, July 28 - August 2, 2019.

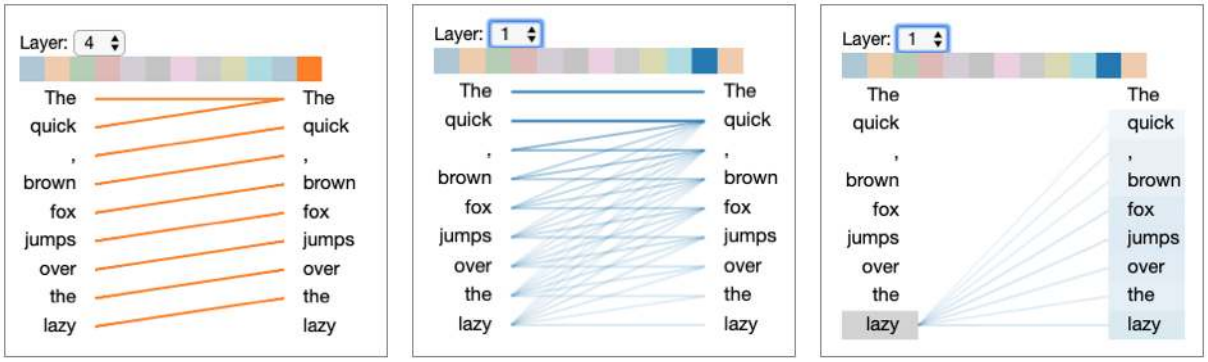

Figure 1: Attention-head view for GPT-2, for the input text *The quick, brown fox jumps over the lazy*. The left and center figures represent different layers / attention heads. The right figure depicts the same layer/head as the center figure, but with the token *lazy* selected.

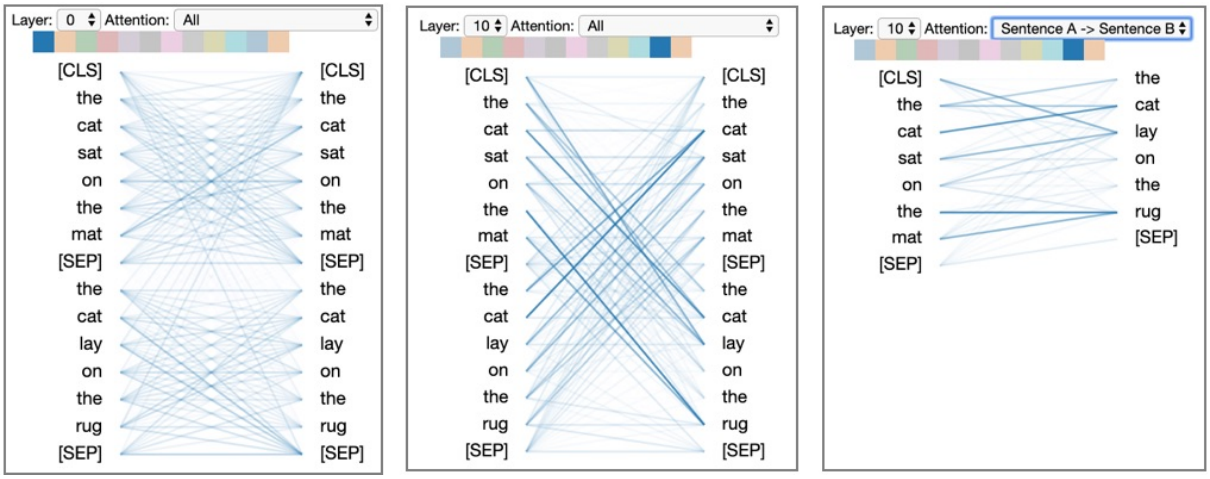

Figure 2: Attention-head view for BERT, for inputs *the cat sat on the mat* (Sentence A) and *the cat lay on the rug* (Sentence B). The left and center figures represent different layers / attention heads. The right figure depicts the same layer/head as the center figure, but with *Sentence A* → *Sentence B* filter selected.

2.1 Attention-head view

The *attention-head view* visualizes the attention patterns produced by one or more attention heads in a given layer, as shown in Figure 1 (GPT-2[1]) and Figure 2 (BERT[2]). This view closely follows the original implementation of Jones (2017), but has been adapted from the original encoder-decoder implementation to the encoder-only BERT and decoder-only GPT-2 models.

In this view, self-attention is represented as lines connecting the tokens that are attending (left) with the tokens being attended to (right). Colors identify the corresponding attention head(s), while line weight reflects the attention score. At the top of the screen, the user can select the layer and one or more attention heads (represented by the colored squares). Users may also filter attention by token, as shown in Figure 1 (right); in this case the target tokens are also highlighted and shaded based on attention weight. For BERT, which uses an explicit sentence-pair model, users may specify a sentence-level attention filter; for example, in Figure 2 (right), the user has selected the *Sentence A* → *Sentence B* filter, which only shows attention from tokens in Sentence *A* to tokens in Sentence *B*.

Since the attention heads do not share parameters, each head learns a unique attention mechanism. In the head shown in Figure 1 (left), for example, each word attends to the previous word in the sentence. The head in Figure 1 (center), in contrast, generates attention that is dispersed roughly evenly across previous words in the sentence (excluding the first word). Figure 2 shows attention heads for BERT that capture sentence-pair patterns, including a within-sentence pattern (left) and a between-sentence pattern (center).

[1] *GPT-2 small* pretrained model.
[2] *BERT-base, uncased* pretrained model.

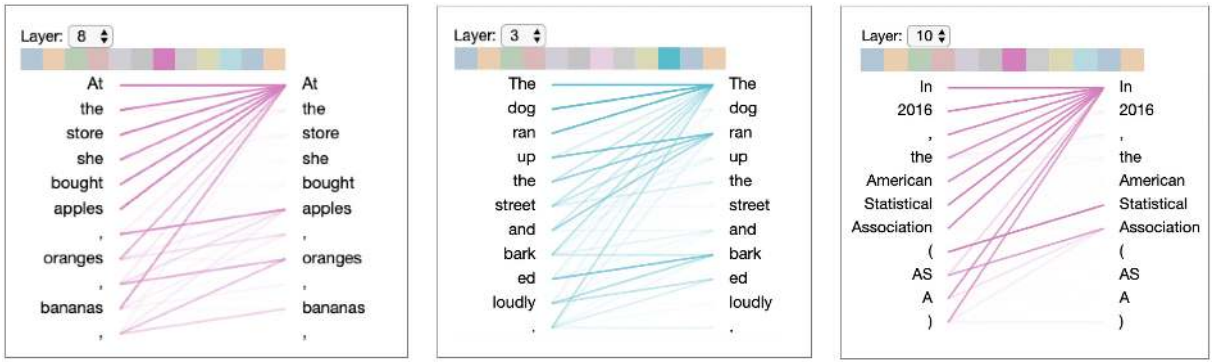

Figure 3: Examples of attention heads in GPT-2 that capture specific lexical patterns: list items (left); verbs (center); and acronyms (right). Similar patterns were observed in these attention heads for other inputs. Attention directed toward first token is likely null attention (Vig and Belinkov, 2019).

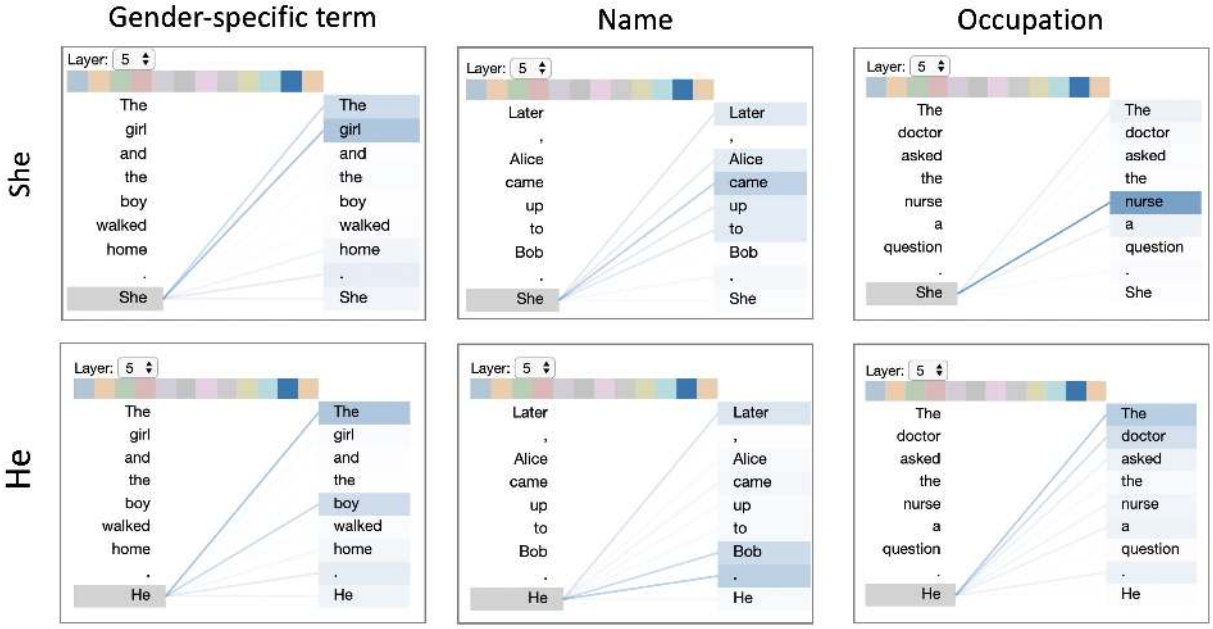

Figure 4: Attention pattern in GPT-2 related to coreference resolution suggests the model may encode gender bias.

Besides these coarse positional patterns, attention heads also capture specific lexical patterns, such as those as shown in Figure 3. Other attention heads detected named entities (people, places, companies), paired punctuation (quotes, brackets, parentheses), subject-verb pairs, and other syntactic and semantic relations. Recent work shows that attention in the Transformer correlates with syntactic constructs such as dependency relations and part-of-speech tags (Raganato and Tiedemann, 2018; Voita et al., 2019; Vig and Belinkov, 2019).

Use Case: Detecting Model Bias

One use case for the attention-head view is detecting bias in the model, which we illustrate for the case of conditional language generation using GPT-2. Consider the following continuations generated[3] from two input prompts that are identical except for the gender of the pronouns (generated text underlined):

- *The doctor asked the nurse a question. She* ***said, "I'm not sure what you're talking about."***
- *The doctor asked the nurse a question. He* ***asked her if she ever had a heart attack.***

In the first example, the model generates a continuation that implies *She* refers to *nurse*. In the second example, the model generates text that implies *He* refers to *doctor*. This suggests that the model's coreference mechanism may encode gender bias (Zhao et al., 2018; Lu et al., 2018). Figure 4 shows an attention head that appears to

[3]Using GPT-2 small model with greedy decoding.

perform coreference resolution based on the perceived gender of certain words. The two examples from above are shown in Figure 4 (right), which reveals that *She* strongly attends to *nurse*, while *He* attends more to *doctor*. By identifying a source of potential model bias, the tool could inform efforts to detect and control for this bias.

2.2 Model View

The *model view* (Figure 5) provides a birds-eye view of attention across all of the model's layers and heads for a particular input. Attention heads are presented in tabular form, with rows representing layers and columns representing heads. Each layer/head is visualized in a thumbnail form that conveys the coarse shape of the attention pattern, following the *small multiples* design pattern (Tufte, 1990). Users may also click on any head to enlarge it and see the tokens.

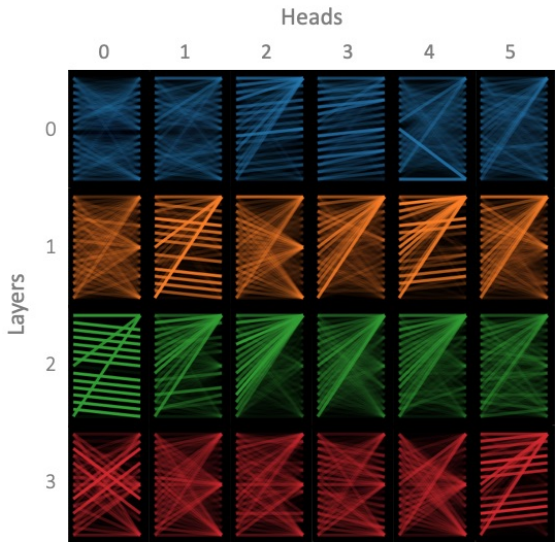

Figure 5: Model view of BERT, for same inputs as in Figure 2. Excludes layers 4–11 and heads 6–11.

The model view enables users to quickly browse the attention heads across all layers and to see how attention patterns evolve throughout the model.

Use Case: Locating Relevant Attention Heads

As discussed earlier, attention heads in BERT exhibit a broad range of behaviors, and some may be more relevant for model interpretation than others depending on the task. Consider the case of paraphrase detection, which seeks to determine if two input texts have the same meaning. For this task, it may be useful to know which words the model finds similar (or different) between the two sentences. Attention heads that draw connections *between* input sentences would thus be highly relevant. The model view (Figure 5) makes it easy to find these inter-sentence patterns, which are recognizable by their cross-hatch shape (e.g., layer 3, head 0). These heads can be further explored by clicking on them or accessing the attention-head view, e.g., Figure 2 (center). This use case is described in greater detail in Vig (2019).

2.3 Neuron View

The *neuron view* (Figure 6) visualizes the individual neurons in the query and key vectors and shows how they interact to produce attention. Given a token selected by the user (left), this view traces the computation of attention from that token to the other tokens in the sequence (right).

Note that the Transformer uses scaled dot-product attention, where the attention distribution at position i in a sequence x is defined as follows:

$$\alpha_i = \text{softmax}\Big(\frac{q_i \cdot k_1}{\sqrt{d}}, \frac{q_i \cdot k_2}{\sqrt{d}}, ..., \frac{q_i \cdot k_N}{\sqrt{d}}\Big) \quad (1)$$

where q_i is the query vector at position i, k_j is the key vector at position j, and d is the dimension of k and q. N=i for GPT-2 and N=len(x) for BERT.[4] All values are specific to a particular layer / head.

The columns in the visualization are defined as follows:

- **Query q**: The query vector of the selected token that is paying attention.
- **Key k**: The key vector of each token receiving attention.
- **q×k (element-wise)**: The element-wise product of the query vector and each key vector. This shows how individual neurons contribute to the dot product (sum of element-wise product) and hence attention.
- **q · 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 is the attention score.

Whereas the attention-head view and the model view show *what* attention patterns the model learns, the neuron view shows *how* the model forms these patterns. For example, it can help identify neurons responsible for specific attention patterns, as discussed in the following use case.

[4]GPT-2 only considers the context up to position i, while BERT considers the entire sequence.

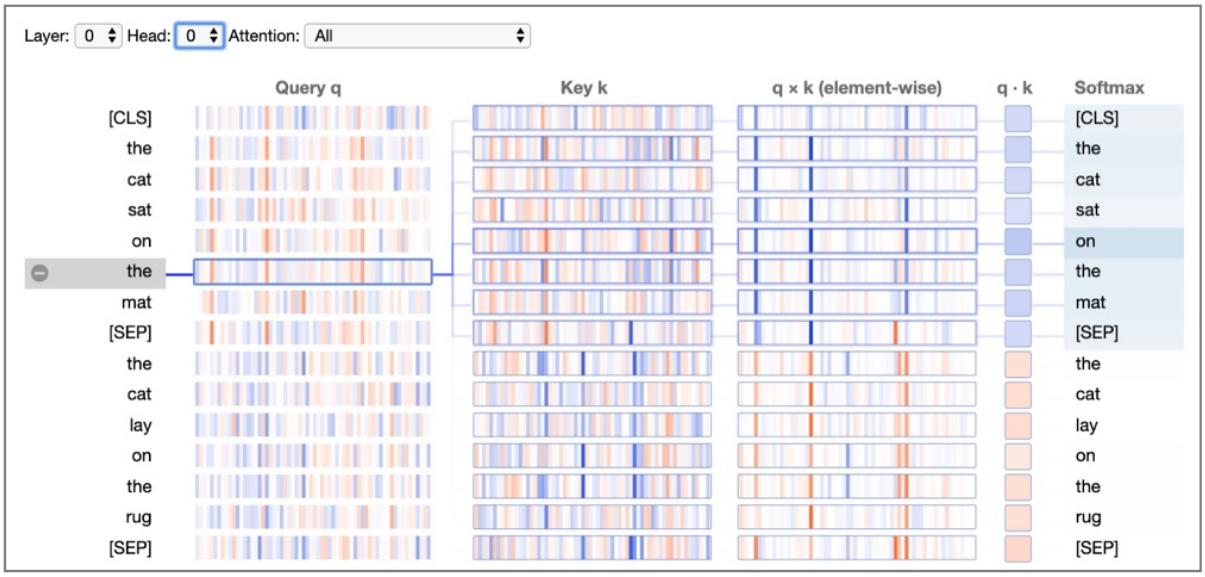

Figure 6: Neuron view of BERT for layer 0, head 0 (same one depicted in Figure 2, left). Positive and negative values are colored blue and orange, respectively, with color saturation based on magnitude of the value. As with the attention-head view, connecting lines are weighted based on attention between the words.

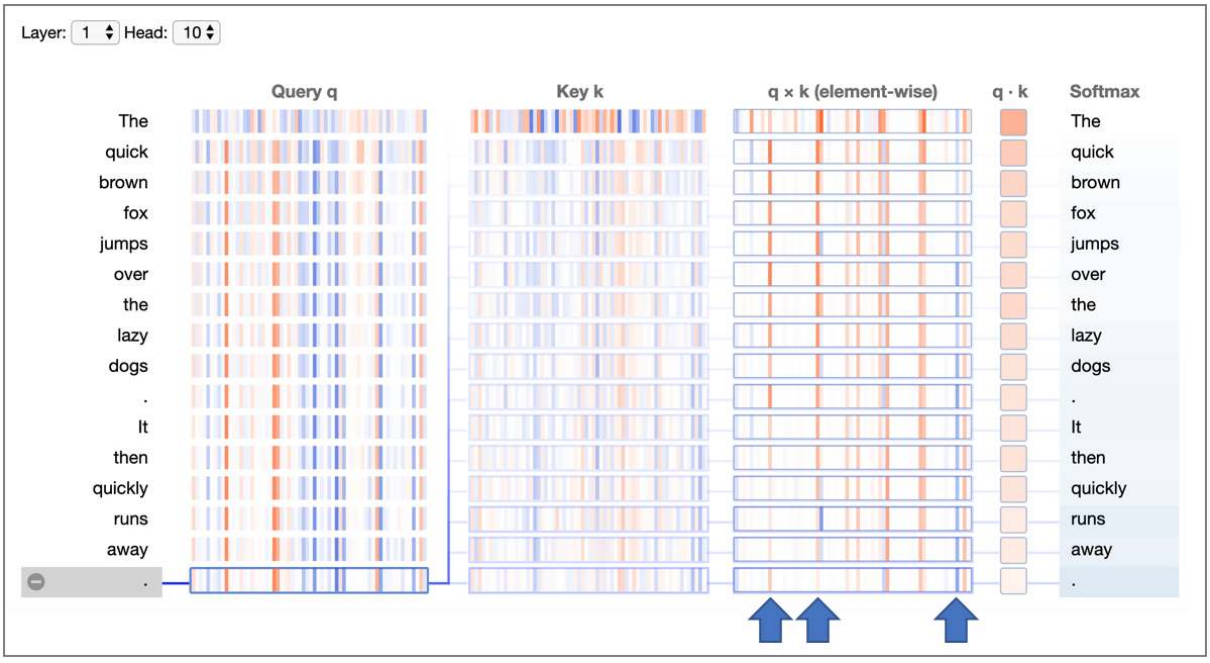

Figure 7: Neuron view of GPT-2 for layer 1, head 10 (same one depicted in Figure 1, center) with last token selected. Blue arrows mark positions in the element-wise products where values decrease with increasing distance from the source token (becoming darker orange or lighter blue).

Use Case: Linking Neurons to Model Behavior

To see how the neuron view might provide actionable insights, consider the attention head in Figure 7. For this head, the attention (rightmost column) decays with increasing distance from the source token. This pattern resembles a context window, but instead of having a fixed cutoff, the attention decays continuously with distance.

The neuron view provides two key insights about this attention head. First, the attention weights appear to be largely independent of the content of the input text, based on the fact that all the query vectors have very similar values (except for the first token). Second, a small number of neuron positions (highlighted with blue arrows) appear to be mostly responsible for this distance-decaying attention pattern. At these neuron positions, the element-wise product $q \times k$ decreases as the distance from the source token increases (either becoming darker orange or lighter blue).

When specific neurons are linked to a tangible outcome, it presents an opportunity to intervene in the model (Bau et al., 2019). By altering the relevant neurons—or by modifying the model weights that determine these neuron values—one could control the attention decay rate, which might be useful when generating texts of varying complexity. For example, one might prefer a slower decay rate (longer context window) for a scientific text compared to a children's story. Other heads may afford different types of interventions.

3 Conclusion

In this paper, we introduced a tool for visualizing attention in the Transformer at multiple scales. We demonstrated the tool on GPT-2 and BERT, and we presented three use cases. For future work, we would like to develop a unified interface to navigate all three views within the tool. We would also like to expose other components of the model, such as the value vectors and state activations. Finally, we would like to enable users to manipulate the model, either by modifying attention (Lee et al., 2017; Liu et al., 2018; Strobelt et al., 2018) or editing individual neurons (Bau et al., 2019).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. ICLR*.

Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2019. Identifying and controlling important neurons in neural machine translation. In *Proc. ICLR*.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *TACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv Computation and Language*.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. *CoRR*, abs/1902.10186.

Llion Jones. 2017. Tensor2tensor transformer visualization. `https://github.com/tensorflow/tensor2tensor/tree/master/tensor2tensor/visualization`.

Jaesong Lee, Joong-Hwi Shin, and Jun-Seok Kim. 2017. Interactive visualization and manipulation of attention-based neural machine translation. In *EMNLP: System Demonstrations*.

Shusen Liu, Tao Li, Zhimin Li, Vivek Srikumar, Valerio Pascucci, and Peer-Timo Bremer. 2018. Visual interrogation of attention-based models for natural language inference and machine comprehension. In *EMNLP: System Demonstrations*.

Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. 2018. Gender bias in neural natural language processing. *CoRR*, abs/1807.11714.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Technical report.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *EMNLP Workshop: BlackboxNLP*.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomas Kocisky, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *Proc. ICLR*.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proc. EMNLP*.

H. Strobelt, S. Gehrmann, M. Behrisch, A. Perer, H. Pfister, and A. M. Rush. 2018. Seq2Seq-Vis: A Visual Debugging Tool for Sequence-to-Sequence Models. *ArXiv e-prints*.

Edward Tufte. 1990. *Envisioning Information*. Graphics Press, Cheshire, CT, USA.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2tensor for neural machine translation. *CoRR*, abs/1803.07416.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Jesse Vig. 2019. BertViz: A tool for visualizing multihead self-attention in the BERT model. In *ICLR Workshop: Debugging Machine Learning Models*.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. In *ACL Workshop: BlackboxNLP*.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *NAACL-HLT*.