BERTVIZ: A TOOL FOR VISUALIZING MULTI-HEAD SELF-ATTENTION IN THE BERT MODEL

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

ABSTRACT

We introduce BertViz, an open-source tool for visualizing self-attention in the BERT language representation model. BertViz extends earlier work by visualizing attention at three levels of granularity: the attention-head level, the model level, and the neuron level. We describe how each of these views can help to interpret the BERT model, and we demonstrate a debugging use case.

1 INTRODUCTION

In 2018, the BERT language representation model achieved state-of-the-art performance across NLP tasks ranging from sentiment analysis to question answering (Devlin et al., 2018). Key to BERT's success was its underlying Transformer model (Vaswani et al., 2017a), which uses a bidirectional, multi-head self-attention architecture. An advantage of using attention is that it can help to interpret a model's decisions by showing how the model attends to different parts of the input (Bahdanau et al., 2015; Belinkov & Glass, 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). A visualization tool designed specifically for the multi-head self-attention in the Transformer (Jones, 2017) was introduced in Vaswani et al. (2017b) and released in the Tensor2Tensor repository (Vaswani et al., 2018).

In this paper we introduce BertViz, a tool for visualizing attention in the BERT model that builds on the work of Jones (2017). We extend the existing tool in two ways: (1) we adapt it to the BERT model, and (2) we add two visualizations: the *model view* and the *neuron view*. We also demonstrate how BertViz may be used to analyze and debug BERT.

2 THE BERTVIZ TOOL

BertViz is an open-source tool for visualizing multi-head self-attention in the BERT model, available at https://github.com/jessevig/bertviz. BertViz comprises three views: an attention-head view, a model view, and a neuron view, described below. A video demonstration of the tool can be found at https://youtu.be/187JyiA4pyk.

2.1 ATTENTION-HEAD VIEW

The *attention-head view* visualizes the attention patterns produced by one or more attention heads in a given transformer layer. Figures 1–3 show examples of this view for the following input sentences: (a) *the cat sat on the mat* and (b) *the cat lay on the rug*.[1] 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

[1] All examples in this paper use the $BERT_{BASE}$ pre-trained model.

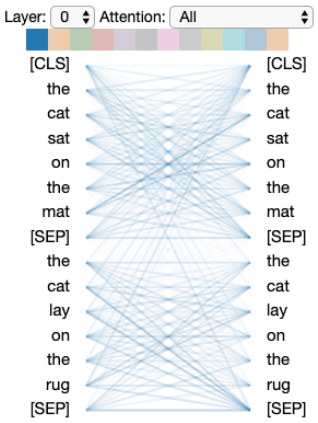

Figure 1: Attention-head view (layer 0, head 0)

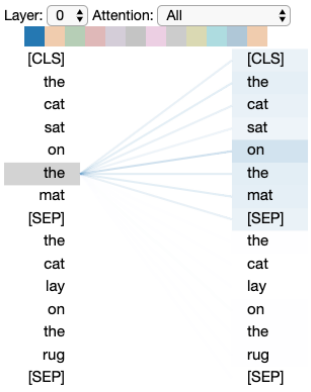

Figure 2: Attention-head view (layer 0, head 0), with token *the* selected.

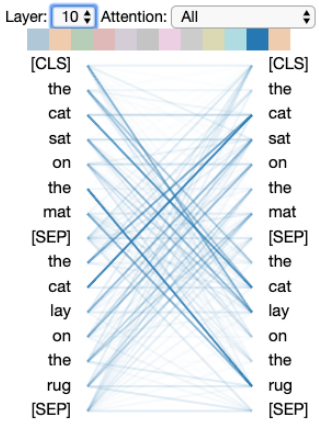

Figure 3: Attention-head view (layer 10, head 10)

top of the screen, the user can select the layer and one or more attention heads (represented by the colored patches), as well as a sentence-level *attention filter*, e.g., a filter that only shows attention from sentence *A* to sentence *B*. Users may also filter attention by token (Figure 2), in which case the target tokens are also highlighted and shaded based on attention strength.

The purpose of the attention-head view is to show how attention flows between tokens for a particular layer/head. In the attention head depicted in Figure 1, for example, one can see that attention is distributed fairly evenly across words in the same sentence. In the attention head in Figure 3, in contrast, attention is focused primarily on related words within the opposite sentence.

This view represents the basic visualization paradigm for BertViz and closely follows the original Tensor2Tensor implementation. The key difference is that the original tool was developed for encoder-decoder models, while BertViz is designed for the encoder-only BERT model. BertViz is also tailored to specific features of BERT, such as explicit sentence-pair (sentence *A* / *B*) modeling.

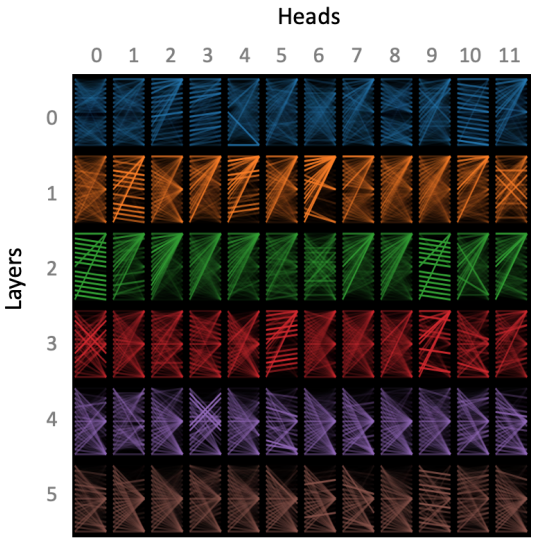

Figure 4: Model view, layers 0 - 5

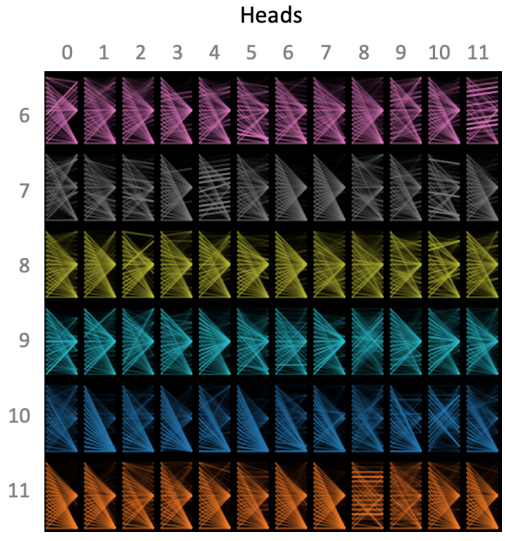

Figure 5: Model view, layers 6–11

2.2 Model View

The *model view* provides a birds-eye view of attention across all of the model's layers and heads, as shown in Figures 4 and 5 (for the same input as in Figures 1–3). Attention heads are presented in tabular form, with rows representing layers and columns representing attention 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.

The model view enables users to quickly browse all attention heads and see how attention patterns evolve through the layers of the model. For example, one can see in Figure 4 that layers 0–2 capture many low-level patterns, e.g., attention to the next word in the sentence (layer 2, head 0). In contrast, layers 9–11 in Figure 5 appear to focus attention on sentence separators, which may encode higher-level sentence representations.

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 scores. 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). The computation is visualized from left to right with the following columns:

- **Query q**: The 64-element query vector of the token paying attention. Only the query vector of the selected token is used in the computations.
- **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 $\cdot$ k**: The dot product of the selected token's query vector and each key vector. This equals the sum of the element-wise product from the previous column.
- **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 reflecting the magnitude of the value. As with the attention-head view, the connecting lines are weighted based on attention between the words. The element-wise product of the vectors is included to show how individual neurons contribute to the dot product and hence attention.

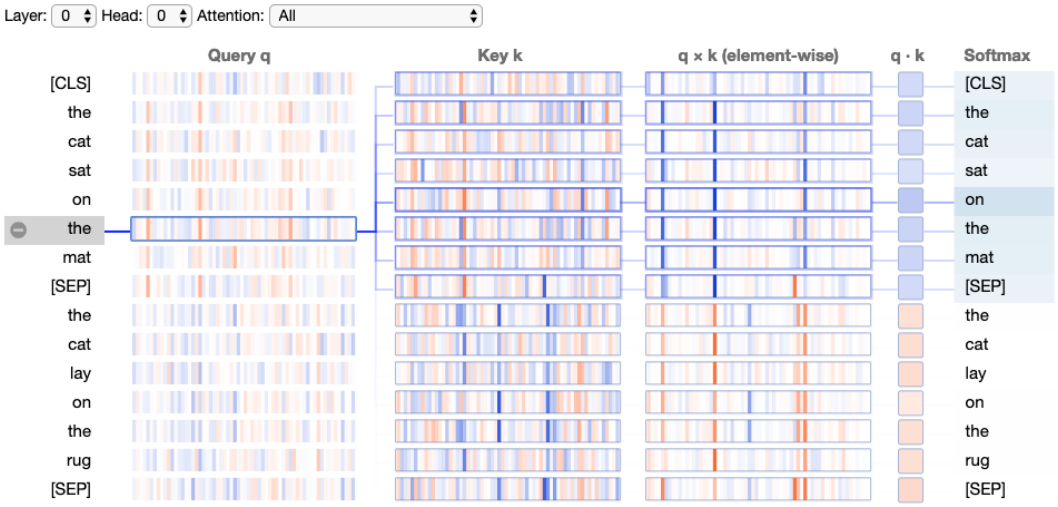

Figure 6: Neuron view for layer 0, head 0, with token *the* selected. This is the same head/token as in Figure 2. Positive and negative values are colored blue and orange, respectively.

Whereas the attention-head and model views show *what* attention patterns BERT learns, the neuron view shows *how* BERT forms these patterns. Consider the neuron view in Figure 6, which shows the attention head from Figure 1 that exhibited a within-sentence attention pattern. From the neuron view, we can see that 3 to 4 neurons (shown in dark blue / orange in $q \times k$ column) mostly determine the dot product and hence attention. The element-wise products are positive for tokens in the same sentence and negative for tokens in different sentences. The reason is 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. The result is the within-sentence attention pattern.

3 BertViz as Debugging Tool: A Case Study

In addition to helping interpret BERT, BertViz may also serve as a debugging tool. We illustrate this capability with an example from the task of paraphrase identification, recognizing that further study is needed to see how the tool generalizes to a broader range of debugging scenarios.

In our scenario, an analyst is running a BERT-based paraphrase detection model[2] on chat data. She wants to know why BERT did not classify these sentences as paraphrases: (a) *the eminem tune was totally awesome.* and (b) *dude, that song from eminem was sick!* (*sick* is a synonym of *awesome* here). She opens the *model view* of BertViz and sees an attention head with a crosshatch shape similar to that of Figure 3, suggesting that the head captures relationships *between* sentences. The analyst brings up the *attention-head* view (Figure 7). She sets the *Sentence A* $\rightarrow$ *Sentence B* (top) and *Sentence B* $\rightarrow$ *Sentence A* (bottom) attention filters to highlight the between-sentence attention. She sees that no attention connects *awesome* and *sick*, suggesting the model missed this relationship.

To test this hypothesis, the analyst replaces *sick* with *terrific* (Figure 8), and sees that *terrific* does attend to *awesome* (see bottom of figure); the model also correctly identifies the two sentences as paraphrases. She also tries an unrelated word: *available* (Figure 9). The attention pattern is almost identical to the one for *sick* (Figure 7), reinforcing her belief that the model sees *sick* as unrelated to *awesome* just as *available* is unrelated to *awesome*. The model also identifies these sentences as not being paraphrases.

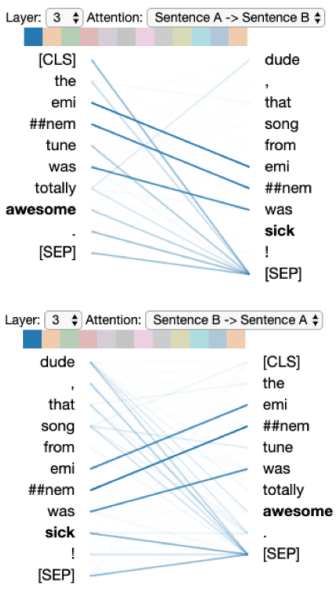

Figure 7: Model incorrectly predicts *paraphrase* = *False*. Note the lack of attention between *awesome* and *sick*.

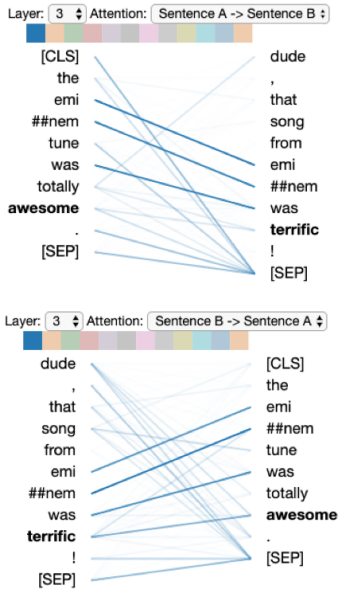

Figure 8: Model correctly predicts *paraphrase* = *True*. Note how *terrific* attends to *awesome* (bottom of figure).

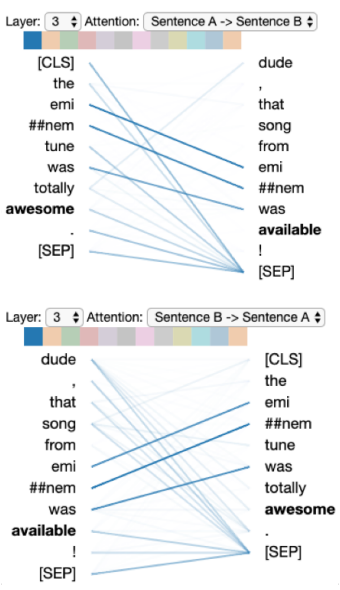

Figure 9: Model correctly predicts *paraphrase* = *False*. Note the lack of attention between *available* and *awesome*.

[2] $BERT_{BASE}$ fine-tuned on the MRPC paraphrase corpus (Dolan & Brockett, 2005)

4 CONCLUSION

In this paper, we introduced BertViz, a tool for visualizing attention in the BERT model. We showed how BertViz may serve as an analysis tool and provided a use case of how it might be used for debugging. For future work, we would like to evaluate empirically how attention impacts model predictions across a range of tasks. We would also like to extend the tool to other models such as the OpenAI GPT (Radford et al., 2018) and the Transformer-XL (Dai et al., 2019). Further, we would like to integrate the three views into a single unified interface, and expose the value vectors in addition to the queries and keys. 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. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

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

Yonatan Belinkov and James Glass. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics (TACL) (to appear)*, 2019.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. 2019. URL http://arxiv.org/abs/1901.02860.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv Computation and Language*, 2018. URL https://arxiv.org/pdf/1810.04805.pdf.

Bill Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*. Asia Federation of Natural Language Processing, January 2005.

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

Jaesong Lee, Joong-Hwi Shin, and Jun-Seok Kim. Interactive visualization and manipulation of attention-based neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 121–126, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-2021. URL https://www.aclweb.org/anthology/D17-2021.

Shusen Liu, Tao Li, Zhimin Li, Vivek Srikumar, Valerio Pascucci, and Peer-Timo Bremer. Visual interrogation of attention-based models for natural language inference and machine comprehension. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 36–41, Brussels, Belgium, November 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D18-2007.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. Technical report, 2018.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomas Kocisky, and Phil Blunsom. Reasoning about entailment with neural attention. In *International Conference on Learning Representations (ICLR)*, 2016.

Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural*

Language Processing, pp. 379–389, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1044. URL https://www.aclweb.org/anthology/D15-1044.

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

Edward Tufte. *Envisioning Information*. Graphics Press, Cheshire, CT, USA, 1990. ISBN 0-9613921-1-8.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. 2017a.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. Technical report, 2017b. URL https://arxiv.org/pdf/1706.03762.pdf.

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. Tensor2tensor for neural machine translation. *CoRR*, abs/1803.07416, 2018. URL http://arxiv.org/abs/1803.07416.