EXBERT: A Visual Analysis Tool to Explore Learned Representations in Transformer Models

Ben Hoover
IBM Research
MIT-IBM Watson AI Lab

Hendrik Strobelt
IBM Research
MIT-IBM Watson AI Lab

Sebastian Gehrmann
Harvard SEAS

{benjamin.hoover,hendrik.strobelt}@ibm.com
gehrmann@seas.harvard.edu

Abstract

Large Transformer-based language models can route and reshape complex information via their multi-headed attention mechanism. Although the attention never receives explicit supervision, it can exhibit recognizable patterns following linguistic or positional information. Analyzing the learned representations and attentions is paramount to furthering our understanding of the inner workings of these models. However, analyses have to catch up with the rapid release of new models and the growing diversity of investigation techniques. To support analysis for a wide variety of models, we introduce EXBERT, a tool to help humans conduct flexible, interactive investigations and formulate hypotheses for the model-internal reasoning process. EXBERT provides insights into the meaning of the contextual representations and attention by matching a human-specified input to similar contexts in large annotated datasets. By aggregating the annotations of the matched contexts, EXBERT can quickly replicate findings from literature and extend them to previously not analyzed models.

1 Introduction

Learned contextualized representations of a neural network can contain meaningful information. Uncovering this information plays a vital role in understanding and interpreting the learned structure of neural networks (Belinkov and Glass, 2019). One way to identify information is to probe the representations by using them as features in classifiers for linguistic tasks, or by identifying contexts that lead to similar patterns (Tenney et al., 2019b; Conneau et al., 2018; Strobelt et al., 2017).

With Transformers (Vaswani et al., 2017) overtaking recurrent models as the primary architectures for many NLP tasks, analyzing attention has become another common strategy for interpretability (Raganato and Tiedemann, 2018a; Clark et al., 2019). These efforts focus on selecting a model, such as BERT (Devlin et al., 2019), and exploring the Transformer's contextual embeddings and attentions across layers to determine whether and where it learns to represent linguistic features. Previous studies have uncovered specific attention heads that learn particular dependencies (Vig and Belinkov, 2019; Clark et al., 2019).

However, once the standard linguistic probing tasks are exhausted, it is challenging to develop new hypotheses to test. Toward that end, interactive visualizations provide a successful strategy to develop new insights and strategies. Visualization tools can offer concise summaries of useful information and allow interaction with large models. Attention visualizations have thus taken significant steps toward these goals of making explorations fast and interactive for the user (Vig, 2019). However, interpreting attention patterns without understanding the attended-to embeddings, or relying on attention alone as a faithful explanation, can lead to faulty interpretations (Brunner et al., 2019; Jain and Wallace, 2019; Wiegreffe and Pinter, 2019; Li et al., 2019).

To address this challenge, we developed **EXBERT**, a tool that combines the advantages of static analyses with a dynamic and intuitive view into both the attentions and internal representations of the underlying model. EXBERT provides these insights for any user-specified model and corpus by probing whether the representations capture meaningful information. We demonstrate that EXBERT can replicate insights from the analysis by Clark et al. (2019) and easily extend it to other models. It is open-source, extensible, and compatible with many current Transformer architectures, both autoregressive and masked language models. **EXBERT** is available at `exbert.net`.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 187–196
July 5 - July 10, 2020.

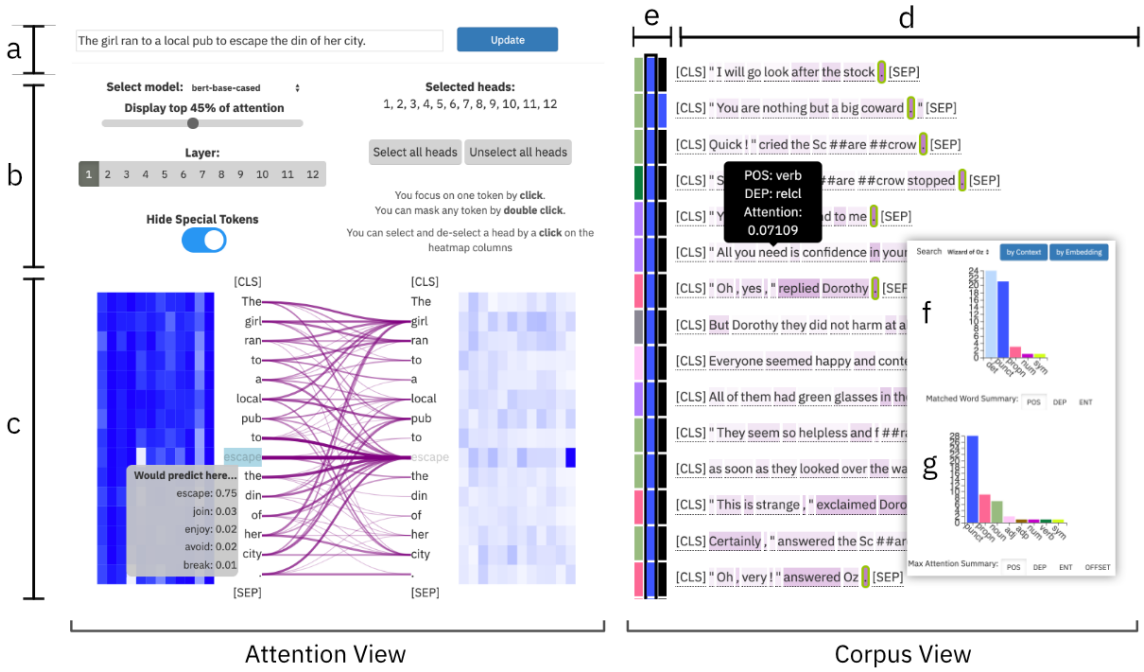

Figure 1: An overview of the different components of the tool. Users can enter a sentence in (a) and modify the attention view through selections in (b). Self attention is displayed in (c), with attentions directed as coming *from* the left column and pointing *to* the right. The blue matrix on the left shows a head's attention (column) out of a token (row), whereas the right-hand matrix shows attention into each token by each head. The top-k predictions for each token are shown on hover in the gray box. The most similar tokens to the MASKed "escape" token in (c) are shown and summarized in (d-g), taken from an annotated corpus (shown: Wizard of Oz). Every token in (d) displays its linguistic metadata on hover. The metadata of the results in (d) are summarized in the histograms (f) and (g) for the matched token (green highlight) and the token of max attention. The colored bars on the histogram correspond to colors in the columns of (e), where the center column summarizes the metadata of the matched token, and the adjacent columns represent the metadata of the words to the left and right of the matched token.

2 Background

2.1 Transformer Models

The Transformer architecture, as defined by Vaswani et al. (2017), relies on multiple sequential applications of *self attention* layers. Self-attention is the process by which each token within an input sequence Y of length N computes attention weights over all tokens in the input. As part of this process, the inputs are projected into a key, query, and value representation with W_k, W_q, and W_v. The Transformer applies I of these *attention heads* in parallel, using separate weights. We denote each head with the superscript $^{(i)}$.

$$\mathbf{A}^{(i)} = \text{softmax}\Big((YW_q^{(i)})(YW_k^{(i)})^\top\Big).$$

This computation yields a matrix in $\mathbb{R}^{N\times N}$ where the entry $\mathbf{A}_{ij}$ represents the attention out of token y_i into token y_j.[1] The representation for each attention head $h^{(i)}$ is then multiplied by the value,

$$h^{(i)} = \mathbf{A}^{(i)}(YW_v^{(i)}).$$

The representations $h^{(1)}, \ldots, h^{(I)}$ are concatenated and followed by a linear projection layer. The output of this projection we call the *token embedding* $E^{(l)}$, which is used as input to layer $l+1$.

2.2 Transformer Analysis

The analysis of learned contextual representation in neural networks has been a widely investigated topic in NLP (Belinkov and Glass, 2019). Before the advent of large pretrained models, analyses focused on models trained for specific tasks like machine translation. Some showed that Transformer models, similar to recurrent models, can

[1] For autoregressive models like GPT-2 (Radford et al., 2019), this matrix is triangular since attention cannot point toward unseen tokens.

effectively encode syntactic properties in their representations (Raganato and Tiedemann, 2018b; Mareček and Rosa, 2018). Researchers have developed suites of probing techniques, agnostic to the underlying model, that can capture these properties across many different linguistic tasks (Tenney et al., 2019b; Conneau et al., 2018). Over the past year, similar tests have primarily been applied to BERT (Devlin et al., 2019) and its derivatives (e.g., Sanh et al., 2019; Liu et al., 2019). Similar to task-specific models, Goldberg (2019) found that BERT clearly encodes syntax within some of its attentions. Moreover, Tenney et al. (2019a) demonstrated that linguistic information is very localized within the representations in different layers.

In parallel, individual attention heads of Transformer models have also received much focus. Clark et al. (2019) showed that individual heads recognize standard Part of Speech (POS) and Dependency (DEP) relationships (e.g., Objects of the Preposition (POBJ) and Determinants (DET)) with high fidelity. Vig and Belinkov (2019) also explored the dependency relations across heads and discovered that initial layers typically encode positional relations, middle layers capture the most dependency relations, and later layers look for unique patterns and structures. These insights are exposed interactively through EXBERT.

3 Overview

EXBERT focuses on displaying a succinct view of both the attention and the internal representations of each token. Figure 1 shows an overview of the tool's two main components. The **Attention View** provides an interactive view of the self-attention of the model, where users can change layers, select heads, and view the aggregated attention. The **Corpus View** presents a user with aggregate statistics that aim to describe and summarize the hidden representations of a currently selected token or set of attention heads. For simplicity, the tool defaults to focus on single-sentence examples.

3.1 Attention View

The attention $\mathbf{A}$ can be understood as an adjacency matrix, which is conducive to a representation of curves pointing from each token to every other token. However, since $\mathbf{A}$ is not symmetric, a visualization has to separate the *outgoing* and *incoming* attention of a token. We achieve this by duplicating the tokens of input Y and presenting it in two vertical sections, connected through the attention.

Hovering over a token will reduce the displayed attention graph to the incoming/outgoing attention of that token. We display the top predictions of the model at that position. Clicking on a token freezes the filtered attention view.

Many models introduce special tokens (e.g., "[CLS]", "<|endoftext|>") for downstream classification or generation tasks. These tokens often receive very high attention and act as a null operation (Clark et al., 2019). We provide a switch to hide the special tokens of the model and renormalize based on the other attentions to provide easier visualization of subtle attention patterns.

3.2 Corpus View

Representations, on the other hand, cannot be easily visualized footnoteSee Strobelt et al. (2017) for a discussion why heat-maps are not an appropriate visualization of hidden states. but they can be understood by searching for similar representations in an annotated corpus. The results of this search are presented in the **Corpus View** with the highest-similarity matches shown first. The histograms display the accumulated features of the matched representations and the token that receives the most attention.

Searching Inspired by Strobelt et al. (2017, 2018), EXBERT performs a nearest neighbor search of embeddings on a reference corpus as follows. A corpus is first split by sentence and its tokens labeled for desired metadata (e.g., POS, DEP, NER). The model then processes this corpus, and its embeddings $E^{(l)}$ are stored at every layer l and indexed for a Cosine Similarity (CS) search using faiss (Johnson et al., 2019). The top 50 most similar tokens matching a query embedding are displayed and summarized for the user in the context of their use in the annotated corpus.

To supplement the layer embeddings $E^{(l)}$ and enable exploration of the attention heads, we derive a *Context Embedding* $C^{(l)}$, which we define as the concatenation of heads before the linear projection at the layer's output. Formally, this is defined as:

$$C^{(l)} = \text{Concat}(\widetilde{\text{h}}^{(l,1)}, \ldots, \widetilde{\text{h}}^{(l,n)}),$$

where $\widetilde{\text{h}}^{(l,i)}$ is defined as the $L2$ normalized representation of head i at layer l to enable CS searching by head. To search the corpus for any subset of

heads $H_s \subseteq \{1, \ldots, n\}$, we set all values of $\widetilde{\mathrm{h}}^{(l,i)}$ to 0 in C^l, where $i \notin H_s$.

Bidirectional vs. Autoregressive Behavior EXBERT is flexible to accommodate both bidirectional and autoregressive Transformer architectures, but the tool behaves slightly differently for each. Bidirectional models have histogram summaries for the nearest neighbor matches across the corpus and allow interactive MASKing of tokens. When hovering over any token, the interface will show what the language model would predict at that token.

Autoregressive models will also search for the nearest neighbors to a selected token's embedding, but the interface will instead summarize the metadata of the following token (indicated in red font). Hovering over any token in the **Attention View** will display what the model would predict *next*.

3.3 Extending EXBERT

EXBERT runs Huggingface's unified API for Transformer models (Wolf et al., 2019) which allows any Transformer model from that API to take full advantage of the **Attention View**.

Similarity searching requires the user to first annotate a corpus with the desired model. Scripts to aid annotation of a corpus from a custom model is provided in the code repository.[2]

To display metadata from a corpus in a custom domain, users will need to align the transformer model's tokenization scheme to extracted metadata (e.g., DNA Sequences and their properties). EXBERT accomplishes this by first tokenizing, normalizing, and labeling the sentence with spaCy (Honnibal and Montani, 2017). If these tokens are split further by the Transformer's tokenization scheme, each word-piece receives the metadata of its parent token. Note that special tokens like "[CLS]" and "<|endoftext|>" have no linguistic features assigned to them.

4 Case Study: BERT

Clark et al. (2019) performed an extensive analysis to determine which heads in a base sized BERT Transformer model learned which dependencies. We show here how some of their insights are easily accessible through the EXBERT interface (Devlin et al., 2019) for the case-sensitive BERT-base model, which has 12 layers and 12 heads per layer.

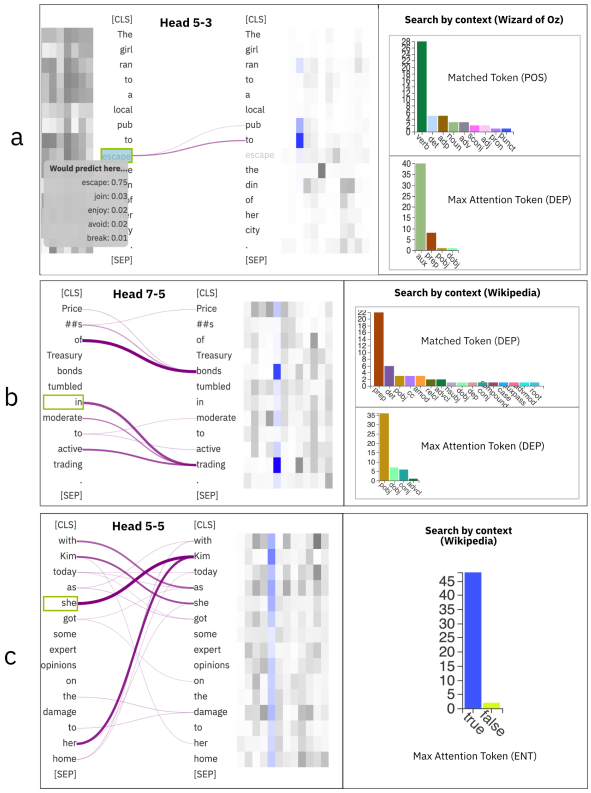

Figure 2: Exploration of different attention heads for pretrained model $\text{BERT}_{\text{base}}$ and different corpora. (a) shows head 5-3 expecting looks at the presents of an auxiliary verb (AUX) to predict that the MASK should be a verb. Head 7-5 in (b) shows a head that has learned to attend to Objects of the Preposition (POBJ). Finally, (c) shows Head 5-5 learning correct co-reference.

We use the notation <layer>-<head> to refer to a single head at a single layer, and <layer>-[<heads>] to describe the cumulative attention of heads at a layer (e.g., 4-[1,3,9] to describe the aggregated attention of heads 1, 3, and 9 at layer 4).

4.1 Behind the Heads

Figure 2 shows examples where distinct heads learn evident linguistic features. Figure 2a shows that the MASKed verb "escape" points to the auxiliary verb (AUX) "to". If we search over the annotated Wizard of Oz[3], we see that the tokens matching the MASK's most similar contexts at Head 5-3 are verbs and that the attention out of these matched words goes primarily to an AUX dependency.

Figure 2b shows that Head 7-5 finds relationships between prepositions (PREP) and their objects (POBJ) in the input sentence. By searching for the token "in" across a subset of the "Wikipedia" corpus (Merity et al., 2016), we confirm that many

[2] https://github.com/bhoov/exbert.

[3] http://www.gutenberg.org/ebooks/55

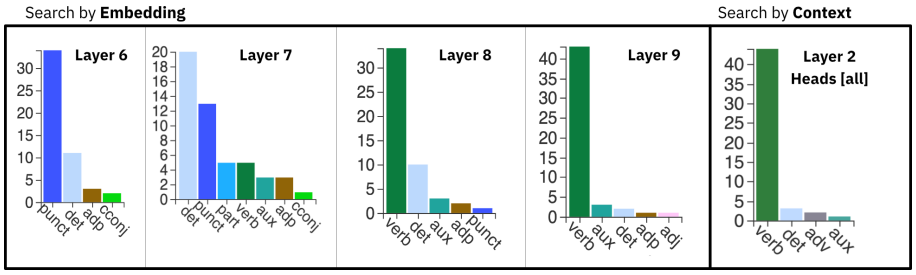

Figure 3: A progression of the information encoded by a nearest neighbor embedding (left) and context (right) searches for the MASKed token "escape" in Figure 2a and the sentence, "The girl ran to a local pub to escape the din of her city." Note that heads encode verb information (dark green) significantly earlier than the embeddings.

other annotated sentences exhibit this pattern.

Figure 2c seemingly finds a head that determines co-reference to entity relationships, as both "she" and "her" are pointing strongly at "Kim" and little to everything else. Because the parse tree is absent in the annotated corpus, we are unable to search for co-reference patterns. However, the corpus search does reveal that this head learns to match pronouns to Entities rather than common gendered words such as "woman" or "mother".

4.2 Behind the Mask

Earlier layers of a BERT model can capture particular linguistic information (Clark et al., 2019; Vig and Belinkov, 2019). We now explore this behavior for a MASKed token across layers. We look at the following sentence, also shown in Figure 2a:

The girl ran to a local pub to escape the din of her city.

We begin by masking the "escape" token in the example sentence at layer 1 and search what information is behind the "[MASK]" token's embedding (Figure 1). Note that at this early layer, there is no meaningful linguistic information encoded in a MASK token's embedding, and the matching embeddings are most similar to punctuation (PUNCT) and determinants (DET), which are the most common tokens in English (Figure 1f). Additionally, the maximum attention out of the MASKed token points to itself (Figure 1c).

As layers progress, more VERB information is encoded in the token's embedding, as shown in Figure 3. At layer 6, the model does not relate the MASKed word to verbs, but by layer 9 it is convinced that the MASK should be a verb. Note that accumulated head information confidently captured a "verb" pattern in a significantly earlier layer.

5 Case Study: GPT-2

5.1 Gender Bias

We now use EXBERT to explore the problem of gender bias and co-reference in autoregressive Transformers (Zhao et al., 2018), a problem inherent in the training data that infects the model's understanding of language (Font and Costa-jussà, 2019). Take the following sentence:

The man visited the nurse and told him to attend to his patients.

We aim to detect whether the model thinks "nurse" is male or female before it sees the masculine pronoun "him" referring to "nurse". Because GPT-2 is trained to predict the next word, we can do this by selecting the token "told" and hovering over it to see the prediction of that pronoun. These results are shown in Figure 4a, and from the probabilities, we can see that GPT-2 predicts "her" with 90% probability. The next closest token "him" is only 6%. Figure 4b shows that replacing "nurse" with "doctor" alters the prediction to be strongly in favor of predicting "him" at 68% probability, while "her" falls to 18%. The attention patterns in the final three layers remain ostensibly the same for both sentences.

5.2 Heads up

In contrast to BERT, GPT-2 is an autoregressive language model. This makes it more difficult to detect some dependencies by looking at attention patterns (e.g., PREP looking for its POBJ in the future). However, EXBERT can offer similar insights as above using slightly altered methods. The following experiments use the smaller configuration of GPT-2 with 12 layers and 12 heads (Radford et al., 2019).

Exploring the heads in GPT-2 reveals that GPT-

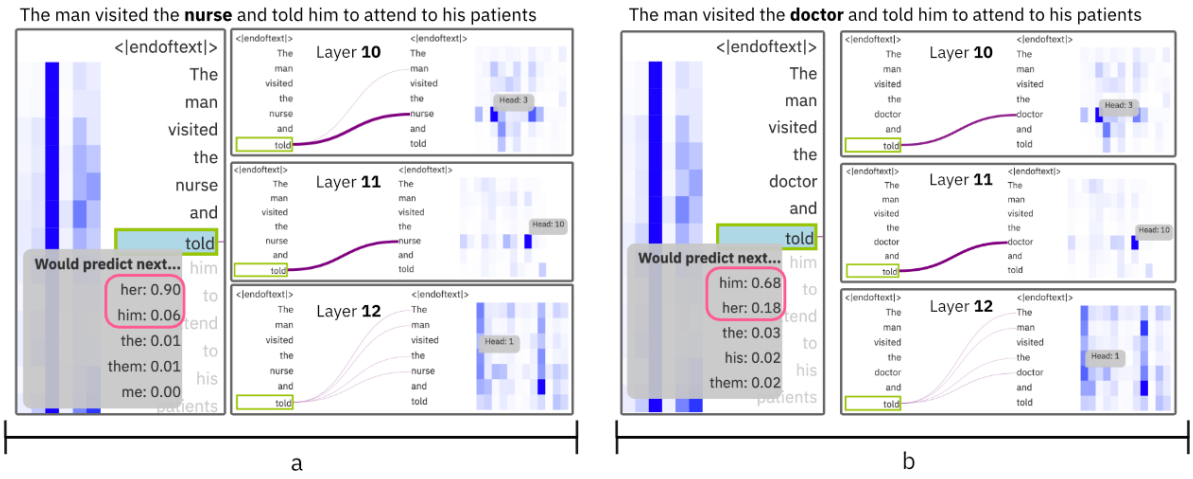

Figure 4: Highlights the bias of the GPT-2 model for generation. (a) "nurse" prompts the model to predict "her". (b) shows "doctor" causing the model to predict "him"

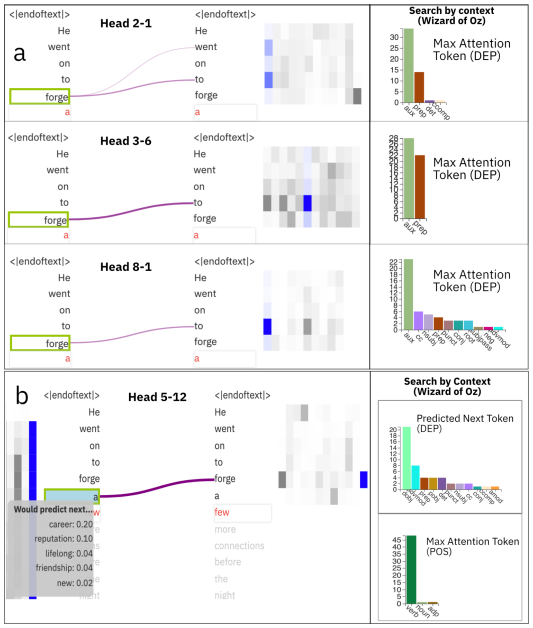

Figure 5: Discovering simple head patterns in GPT-2 using the sentence. (a) shows strong detection of the AUX dependency. (b) shows a head detecting the DOBJ dependency

2's heads also learn distinct syntactic structure. Figure 5a shows a few heads at different layers that seemingly learn the AUX dependency. Heads at earlier layers show an affinity for the AUX pattern, but also confuse "to" with a preposition even though a verb directly follows. This behavior hints that these heads look primarily to match the word "to" rather than its contextual meaning.

Similarly, Figure 5b shows a head that attends predominantly to a preceding verb and matches contexts in which the following word is a DOBJ. Interestingly, the more complex DOBJ dependency is picked up by a head as early as layer 5-12, whereas a simpler dependency like the AUX pattern is only clearly detected later in Layer 8.

6 Discussion

In this paper, we introduced an interactive visualization tool, EXBERT, that can reveal an intelligible structure in the learned representations of Transformer models. We demonstrated, through an attention visualization and nearest neighbor searching techniques, that EXBERT can replicate research that explores what attentions and representations learn and detect biases in text inputs.

We acknowledge that EXBERT is limited compared to more global analyses since it only presents statistics across a small number of neighbors for a single token at a time. These neighbors do not necessarily reveal a head's or an embedding's global behavior. However, EXBERT can effectively narrow the scope and refine hypotheses through quick testing iterations. These hypotheses about the model behavior can, in a later step, be evaluated by robust statistical tests on a global level.

To assist researchers with their model investigations, we host a demo of the tool with multiple models at `exbert.net`.

References

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Gino Brunner, Yang Liu, DamiÃąn Pascual, Oliver Richter, and Roger Wattenhofer. 2019. On the validity of self-attention as explanation in transformer models.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of bert's attention. *CoRR*, abs/1906.04341.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Joel Escudé Font and Marta R. Costa-jussà. 2019. Equalizing gender biases in neural machine translation with word embeddings techniques. *CoRR*, abs/1901.03116.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.

Xintong Li, Guanlin Li, Lemao Liu, Max Meng, and Shuming Shi. 2019. On the word alignment from neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1293–1303, Florence, Italy. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

David Mareček and Rudolf Rosa. 2018. Extracting syntactic trees from transformer encoder self-attentions. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 347–349.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *CoRR*, abs/1609.07843.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

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

Alessandro Raganato and Jörg Tiedemann. 2018b. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Hendrik Strobelt, Sebastian Gehrmann, Michael Behrisch, Adam Perer, Hanspeter Pfister, and Alexander M. Rush. 2018. Seq2seq-vis: A visual debugging tool for sequence-to-sequence models. *CoRR*, abs/1804.09299.

Hendrik Strobelt, Sebastian Gehrmann, Hanspeter Pfister, and Alexander M Rush. 2017. LSTMVis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. *IEEE transactions on visualization and computer graphics*, 24(1):667–676.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. *CoRR*, abs/1906.05714.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. *CoRR*, abs/1906.04284.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, RÃl'mi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Transformers: State-of-the-art natural language processing.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.

A Recreating the experiments

We allow direct linking to an experimental setup in the interface. A list of the links to reproduce our results is given below (all links in the supplementary material are correct at the time of publishing, but may be changed in the distant future):

- Overview (Figure 1): https://bit.ly/2OfD6Vt
- Behind the Heads (Figure 2)
  - (a): https://bit.ly/2GJUihs
  - (b): https://bit.ly/38Ycss8
  - (c): https://bit.ly/2S8qGzO
- Behind the Mask (Figure 3): https://bit.ly/2RJ952n
- GPT-2 Bias (Figure 4): https://bit.ly/36ELwMo
- Heads Up (Figure 5):
  - (a): https://bit.ly/2vAcgRe
  - (b): https://bit.ly/2S9qHDs

B Additional Material

In addition to the content presented in the main paper, we have recorded a short **video demo** showing how to use the tool to probe for particular patterns at https://youtu.be/e31oyfo_thY.

A **Lite version** of the tool, without the corpus searching, demoing many common Transformer models is hosted by Huggingface at huggingface.co/exbert.

C Additional figures

Figure 6: The most similar embeddings, in context, to the MASKed token “escape” in the sentence: “The girl ran to a local pub to escape the din of her city” at the output of layer 12 of $\text{BERT}_{\text{base}}$ (shown in Figure 2a). Corpus results are annotated excerpts from the Wizard of Oz. Notice how at the output layer all attentions are primarily to the word itself or the final punctuation mark of the sentence, indicating that the most important information is likely already encoded in the selected token’s embedding.