Visualizing Transformers for NLP: A Brief Survey

Adrian M.P. Braşoveanu
MODUL Technology GmbH
Am Kahlenberg 1, 1190, Vienna, Austria
and
Transilvania University, Braşov, Romania
Email: adrian.brasoveanu@modul.ac.at

Răzvan Andonie
Central Washington University
Ellensburg, WA, USA
and
Transilvania University, Braşov, Romania
Email: razvan.andonie@cwu.edu

***Abstract*—The introduction of Transformer neural networks has changed the landscape of Natural Language Processing during the last three years. While models inspired by it have managed to lead the boards for a variety of tasks, some of the mechanisms through which these performances were achieved are not necessarily well-understood. Our survey is focused mostly on explaining Transformer architectures through visualizations. Since visualization enables some degree of explainability, we have examined the various Transformer facets that can be explored through visual analytics. The field is still at a nascent stage and is expected to witness dynamic growth in the near future, since the results are already interesting and promising. Currently, some of the visualizations are relatively close to their original models, whereas others are model-agnostic. The visualizations designed to explore the Transformer architectures enable some additional features, like exploration of all neuronal cells or attention maps, therefore providing an advantage for this particular task. We conclude by proposing a set of requirements for future Transformer visualization frameworks.**

***Index Terms*—Natural Language Processing, Transformers, BERT, Attention Maps, Visualization of Neural Networks**

I. Introduction

To correctly interpret a text, it is important to understand both its content (e.g., meaning of words, sentences or phrases) and context (e.g., where, when and why was this text produced?). This indicates that the application of Deep Learning (DL) models on texts should cover the morphological, syntactic, semantic and pragmatic layers. Crafting networks that operate on so many different levels is a challenging task due to the sparseness of the training data.

The first implementation of a Transformer network [1] proved that it was possible to design such networks that achieve good results for Natural Language Processing (NLP) tasks with a set of multiple sequential attention layers. The Transformer model itself is simple and consists from an encoder and a decoder. The encoder typically contains a set of multi-head attention layers, add and normalize and feed-forward layers. The multi-head attention mechanism provides the model with the ability to orchestrate information from different representation subspaces (e.g., multiple weight matrices) at various positions (e.g., different words in a sentence) [1]. Its outputs are fed either in other encoders or into decoders, depending on the architecture. There is no fixed number of encoders and decoders which can be included in this architecture, but they will typically be paired (e.g., 10 encoders and 10 decoders). In newer architectures, encoders and decoders can also be used for different tasks (e.g., encoder for Question Answering, and decoder for Text Comprehension) [2]. While the model was initially developed for machine translation tasks, it has been tested on multiple domains and was demonstrated to work well.

In the last three years, hundreds of papers and language models inspired by Transformers were published, the best-known being BERT [3], RoBERTa [4], AlBERT [5], XLNet [6], DistilBERT [7], and Reformer [8]. Some of the most popular Transformer models are included in the Transformers library, maintained by HuggingFace [9].

Many of these models are complex and include significant architectural improvements compared to the early Transformer and BERT models. Explaining their information processing flow and results is therefore difficult, and a convenient and very actual approach is visualization. Our survey is focused on visualization techniques used to explain Transformer architectures, with an emphasis on the most recent architectures. We investigate two large tool classes: (i) model-agnostic tools that can be used to explain BERT predictions; and (ii) custom visualizations that are focused only on explaining the inner workings of Transformers. To the best of our knowledge, this is the first survey dedicated to this topic.

The rest of the paper is organized as follows: Section II presents the motivation and methodology of this survey; Section III showcases the two classes of tools; whereas Section IV discusses the various findings. The paper concludes with some thoughts on the future of Transformers visualizations.

II. Background and Methodology

While NLP technology has significantly improved since the introduction of Transformers [1], it is still difficult to: (i) capture specialized domain knowledge (e.g., medicine, business, sports, etc) without external Knowledge Graphs [10]; (ii) remove or flag bias or propaganda [11]; and (iii) perform fast retraining [12]. The need to quickly update NLP models in case of unforeseen events suggests that developers will be well-served by explainable AI and visualization libraries, especially since debugging Transformers is a complex task. Visualizations are particularly important, as they help us debug the various problems that such models exhibit and which can only be discovered through large-scale analyses.

978-1-7281-9134-8/20/$31.00 ©2020 IEEE
DOI 10.1109/IV51561.2020.00051

Traditional visualization libraries are based on the classic grammar of graphics philosophy [13] which is focused on the idea that visualizations are compositional by design. They provide various *visualization primitives* like circles or squares and *a set of operations* that can be applied on top of these primitives in order to create more complex shapes or animations. Unfortunately such traditional visualization libraries like D3.js [14], Vega [15] or Tableau[1], do not offer specific functions for visualizing feature spaces, neural network layers or support for iterative design space exploration [16] when designing AI models. What this means is that for AI tasks, a lot of the functionality will have to be developed from scratch.

When visualizing more complex models like those built with Transformers, we typically need to understand all the facets of the problem, from the data and training procedure, to the input, network layers, weights or various outputs of the neural network. This can sometimes be accomplished by using model-agnostic tools specifically built for benchmarking or hyperparameter scoring, such as *Weights and Biases*. We include such tools in our survey only if examples of how to use them for visualizing Transformers already exist, either in scientific papers or other types of media posts (Medium posts, GitHub, etc.).

The second big class of visualizations discussed in this article is, naturally, the class of visualizations specifically built around Transformers, either for the purpose of explaining it (like ExBERT [17]), or for explaining certain model specific attributes (like embeddings or attention maps [18]).

We selected the libraries and visualizations presented here by reviewing the standard Computer Science (CS) libraries (e.g., IEEE, ACM, Elsevier, Springer, Wiley), but also online media posts (YouTube, Medium, GitHub and arXiv). In this extremely dynamic research field, some articles might be published on arXiv even up to a year before they are accepted for publication in a traditional conference or journal, time in which they might already garner hundreds of citations. The original BERT article [3] and also one of the first articles that used visualization to explain it [19] [2] were cited over a hundred times before being published in conference proceedings.

Due to space limitations, we resume ourselves to discussing only the most interesting visualizations, especially in the model-agnostic visualization section, as otherwise this article could easily triple in size.

III. Visualizing Transformers

Using visualization in order to explain the AI processes is an expanding research field. The main idea behind AI user interfaces should be to augment and expand user capabilities rather than replace intelligence [20]. While not necessarily needed in order to understand the next section, several recent surveys about visualizations and DL can help provide additional context to the interested readers. We particularly recommend the following: the introduction on how Convolution Neural Networks "see" the world from [21], the discussion on visual interpretability from [22], and the discussion on the importance of visualizing features from [23].

Some of the papers that guided the selection of the works discussed here were those created by Hendrik Strobelt, especially Lstmvis [24] and Seq2Seq-Vis [25] as they establish the basic principles of using visualization for debugging visual networks. He also co-authored papers that explain how to creatively use visualizations for the discovery of neural architectures [26] or to create model-agnostic visualizations for temporal debugging of classifier confusion [27].

An early survey about the role of visualization in Computer Vision [28] and another one on the interpretation of black box DNNs for Computer Vision [29] can help us form an understanding of the entire DL visualization domain.

A. Visualizing Transformers with Model-Agnostic Tools

1) Explainable AI Libraries: Explainable[3] AI (XAI) is the key to enterprise adoption of the current wave of AI technologies, from vision to NLP and symbolic computation. A survey focused on the five Ws (Why? What? When? Who? Where?) [30] presents various methods through which visualizations can be incorporated into the process of explaining the results of the AI models and defines the terminology of the field. Early XAI libraries have featured visualizations in order to understand the features incorporated into the ML models, whereas more recent libraries are focused on visualizing the key neural network layers like embeddings or attention maps [31].

Regardless of the DL domain (e.g., NLP, Speech Processing, Computer Vision, etc), in order to address calls for more transparency when designing large-scale neural networks, a first step is to explain what each model contains in terms of input, processing and output - the latter in terms of the contribution of each feature to the results. Traditionally, variable importance was used to describe this contribution [32]. Due to the fast development of many new neural models, there was a need for a more nuanced description. Shapley values represent an attempt to create such a set of more nuanced descriptors. The contributions of all features included in a particular model are taken together, and then a score signifying the importance of that feature within that set of features is generated. If some features are added or removed from this set, naturally the Shapley value for a particular feature will change accordingly.

Some of the early model-agnostic XAI libraries that were applied to NLP and Transformers visualizations include LIME [33] and SHapley Additive exPlanations (SHAP) [34]. The later was introduced in order to unify multiple explanation methods into a single model for interpreting predictions. Both SHAP and LIME can be used with classical ML libraries like scikit-learn and XGBoost, as well as with modern DL libraries like PyTorch or Keras / TensorFlow. SHAP provides visualizations for summary and dependency plots. Unfortunately, both

[1] www.tableau.com

[2] Article [19] has garnered 149 citations at the moment of the submission, before being published in a conference or journal.

[3] *Explainable* refers to the possibility to explain from a technical point of view the prediction of an algorithm.

have been proven to be easily fooled by adversarial attack strategies [35].

The visualizations created with LIME and SHAP are typically restrained to classic charts (e.g., line, bars or word clouds. The summary plots or interaction charts [34] from SHAP are relatively easy to understand, whereas the more complex force plot charts like feature impact [36] are not necessarily easy to use as they require a certain learning time. While the feature impact chart simply plots the expected feature impact with red (features with positive contribution to the prediction result) or blue (features with a likely negative contribution to the result) colors and should in theory be an easy to understand chart, there are no direct (e.g., in chart via a legend) explanations on how to interpret the start or end values, or what do the indicators placed on top of various components mean in some cases. The interpretation of such force plot charts is generally missing and people need to read additional documentation in order to understand the results. This is far from ideal, as, in our opinion, visualizations need to be self-explanatory.

Another XAI alternative to SHAP and LIME, ELI5 [37], is currently routinely used for explaining BERT predictions, and seems to be somewhat secure, at least at the current time.

Perhaps the most interesting explainable AI library is AllenNLP's Interpret [38]. Initially designed to be model-agnostic, it is increasingly used to explain various attacks on Transformer models. One of its main visualizations is a saliency map that showcases gradient's loss. The library works for a variety of tasks, from reading comprehension to text classification, Named Entity Recognition (NER) and coreference resoution (e.g., finding all expressions that refer to a single entity). The initial paper also demonstrates a word-level Hot Flip attack in which words are replaced in a sentiment model causing a shift from positive to negative in the final prediction results. Many of the BERT visualizations are model specific, therefore, in our view, using Interpret is perhaps best in situations in which we might want to test multiple models, as we will not be too focused on the internals of each model in such a scenario.

Many other explainabile AI libraries use Shapley values for computing feature importance. However, in many cases we were only barely able to discover mentions of their usage for NLP and even then this was mentioned more like a possibility than a reality (e.g., DeepExplain [4]). In such situations we have not included them in this survey.

2) Hyperparameter Optimization and Benchmarking: When testing new models, benchmarking and fine-tuning are the two operations were we might spend the most time, as even if the scores are good, we might want to try different hyperparameter settings (e.g., learning rate, number of epochs, batch size, etc) [39], [40]. A *hyperparameter sweep* (or trial) is a central notion in both hyperparameter optimization and benchmarking and involves running one or multiple models with different values for their hyperparameters. Since quite often the main goal behind running hyperparameter optimization or benchmarking is improving existing models, we have treated these two types of libraries as a single class.

TensorBoard[5] is typically deployed with Google TensorFlow distributions. It is a specialized dashboard that includes most of the visualizations needed for ML experiments, from tracking, computing and visualizing metrics, to model profiling and embeddings. Since it was the first mover in the space, it is used in many projects with all the major libraries like TensorFlow, PyTorch or FastAI.

Neptune[6] is an open-source ML benchmarking suite. It has been used for a variety of collaborative benchmarking tasks and supports notebook tracking. This eases the development of ML models for programmers, therefore it is widely used in the industry. *Sacred*[7] and *Comet.ml*[8] are Neptune alternatives that provide basic charting capabilities and dedicated dashboards.

Weights and Biases[9] provides perhaps the largest sets of visualization and customization capabilities. It comes packed with advanced visualizations that include parallel coordinates [41], perhaps the best method to navigate hyperparameter sweeps. It is the easiest and the most agile solution to integrate with production code or Jupyter notebooks out of all the ones mentioned here.

Besides *Weights and Biases*, another solution that is currently popular for fine-tuning and benchmarking Transformer models is *Ray* [42], a distributed benchmarking framework. It also contains its own fine-tuning engine called *Tune* [43].

B. Specialized Transformer Visualizations

We have examined around 50 papers describing Transformer visualizations. We have only selected papers that presented Transformer visualizations related to NLP. We have also eliminated all the papers that have used only classic bar or line charts. In general, we preferred to focus on the works that tried to visualize as many different aspects of Transformer models as possible.

As expected, a large number of visualizations are solely dedicated to Transformer models. This is mainly due to the fact that visualizations often come as an afterthought when developing new models. They are not necessarily always the focus of new research, but rather they help support the new research. For example, a large set of static and dynamic visualizations was simply developed in order to teach and explain Transformers and BERT. No other types of neural networks have led to such an increased demand for custom visualizations since the days of the Kohonen's Self-Organizing Maps [44] or Manbelbrot's fractals [45]. We have decided to also include such supporting visualizations in our survey, as they help explain Transformer networks. We have labelled most of these as tutorials.

[4]https://github.com/marcoancona/DeepExplain

[5]https://www.tensorflow.org/tensorboard
[6]https://neptune.ai/
[7]https://github.com/IDSIA/sacred
[8]https://www.comet.ml/site/
[9]https://www.wandb.com/

1) Transformer Tutorials: Jay Alammar has produced two good BERT introductory tutorials: *A Visual Guide to Using BERT for the First Time*[10] and *The Illustrated BERT, ELMo, and co.*[11]. The first one uses emoticons (emotion icons) and simple illustrations to explain the basics behind BERT, whereas the second tutorial is focused on transfer learning.

Jesse Vig's series on deconstructing BERT is similar in nature, but provides more dynamic illustrations. Some of these are extracted from Vig's visualization papers which will be discussed in the next paragraph. Following the same tradition, two other good tutorials deserve to be mentioned here: Peter Bloem's *Transformers from Scratch*[12] and Samira Abnar's *From Attention in Transformers to Dynamic Routing in Capsule Networks*[13]. Illustrated tutorials are available for many of the models.

The last set of tutorials that we think deserve a special mention are the ones focused on annotated models, from Harvard NLP's *The Annotated Transformers*[14] or Ama Arora's *The Annotated GPT-2*[15] to most of the tutorials and online demos (e.g., Write with Transformers demos) included in the Transformers[16] library [9].

2) Transformer Visualizations: The recent success of Transformers helped power many NLP tasks to the top of the leaderboards. BERT visualizations have focused on explaining these great results through visualizations, therefore highlighting: (i) the role of embeddings and relational dependencies within the Transformer learning processes [62]; (ii) the role of attention during pre-training or training (e.g., [63] or [18]) or (iii) the importance of various linguistic phenomena encoded in its language model like direct objects, noun modifiers, possesive pronouns or coreferents [19].

Current XAI methods for Transformer models have further developed and supported the idea that understanding the linguistic information which is encoded in the resulting models is key towards understanding the good performances in NLP tasks. For example, by using structural probing [64], structured perceptron parsers [65]) or visualization (e.g., as demonstrated through BERT embeddings and attention layers visualizations like those from [19] and [18]), one should be able to understand what kind of linguistic information is encoded into a Transformer model, but also what has changed since previous runs. Probing tasks [66] are simple classification problems focused on linguistic features designed to help explore embeddings and language models.

We have discovered two large classes of Transformer visualizations:

- *Focused* - visualizations centered on a single subject like attention. The papers themselves might present multiple visualizations, but these visualizations are not single tools.
- *Holistic* - visualizations or systems which seek to explain the entire Transformer model or lifecycle.

The most important papers dedicated to focused visualizations are summarized in Table I. The main characteristic that connects these papers is their dedication to a single topic, regardless of the number of visualizations included in them. We have analyzed the following characteristics:

- *Topic* - the main topic of the paper (e.g., attention, representation, information probing);
- *Visualization Subject* - since visualizations included in these papers were focused on a large set of subjects from Transformer components (e.g., attention heads), to correlation between tasks (e.g., via Pearson correlation charts) or performance (e.g., accuracy or other metrics represented via line charts), we have decided to extract all these in a separate column in order to understand what kind of charts we might be interested in creating when exploring a certain topic.
- *Chart Type* - includes the various types of visual metaphors used for rendering the chosen subjects.

We can clearly distinguish several large topics in this group of focused papers: the relation between attention and model outputs (e.g., especially in [46], [47], [49], [51]), the analysis of captured linguistic information via probing (e.g., in [52], [54]), the interpretation of information interaction (e.g., in [48], [52]), and multilingualism (e.g., in [54], [56]). Papers that work on similar topics also tend to use the same kind of visual metaphors. This happens sometimes due to replication of a previous study (e.g., [47] replicates the experiments from [46] in order to prove that attention weights do not explain everything), whereas in other cases this happens due to the fact that there is no need for more complicated visual metaphors (e.g., line charts are used in more than half of the papers in order to represent performance). Besides the widespread use of the matrix charts that represent attention maps, one chart type that deserves to be highlighted in this category is the attention graph [48] which tracks the information flow between the input tokens for a given prediction.

Some of the most interesting tools or papers included in the category of *holistic visualizations* are compared in Table II. These visualization systems typically integrate most of the components of a Transformer and provide detailed summaries for them. We have examined two large classes of attributes:

- *Components* represents the various components of the neural networks: from corpus, to embeddings, positional heads, attention maps or outputs.
- *Summary* includes the various types of views that offer us information about the state of a neuron or a layer, as well as overviews, statistics or details about the various errors encountered. Statistics might include different types of information: from correlations between *layers* or *neurons* to *statistical analyses of the results*. The *errors* column represents any error analysis method through which we

[10]http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/
[11]http://jalammar.github.io/illustrated-bert/
[12]http://www.peterbloem.nl/blog/transformers
[13]https://samiraabnar.github.io/articles/2019-03/capsule
[14]http://nlp.seas.harvard.edu/2018/04/03/attention.html
[15]https://amaarora.github.io/2020/02/18/annotatedGPT2.html
[16]https://github.com/huggingface/transformers

TABLE I
ARTICLES FOCUSED ON EXPLAINING TRANSFORMER TOPICS THROUGH VISUALIZATIONS.

Article	Topic	Visualization Subject	Chart Type
Jain [46]	relation between attention and outputs	feature importance correlation permutation adversarial attention charts	Kendall τ statistics / histograms / line charts permutation scatterplots adversarial histograms and scatterplots
Wiegreffe [47]	explaining attention	permutation performance	permutation scatterplots multiple line charts
Hao [48]	information interactions interpretation	attention scores information flow between tokens evaluation accuracy correlation of attention heads	simple attention map attribution graphs line charts Pearson correlation coefficient chart
Abnar [49]	attention flow	raw attention graph raw attention map	attention graph attention map
Vig [50]	causal mediation analysis	indirect effects effects comparison attention heads	averaged attention heatmaps / line charts line chart attention heads
Voita [51]	analysis of the multi-head self-attention	layers visualizations attention maps for the rare words head dependency scores and distribution charts performance active heads	importance charts / matrix charts attention maps bar charts line charts line charts
Voita [52]	evolution of representations in Transformers	token changes and influences distances between tasks or layers token occurences	line charts line charts t-SNE clustering
Voita [53]	information theoretic probing of classifiers	code components learning curves and performance	bar charts line charts
Tenney [54]	analysis of captured linguistic information	summary statistics layer-wise metrics probing of predictions across layers	bar chart bar+distribution chart multiple bar charts
Dufter [55]	multilinguality	embeddings positional embeddings performance	Principal Components Analysis cosine similarity matrices line charts
Egger [56]	probing low-resource languages	stability of training size probing tasks and downstream tasks	bar chart Pearson correlation charts
Song [57]	role of BERT intermediate layers	clustering on intermediate layers	Principal Component Analysis

TABLE II
COMPARISON OF HOLISTIC TRANSFORMER VISUALIZATIONS.

Article	Components					Summary				
	corpus	embeddings	positional heads	attention map	outputs	errors	neuron	layers	overview	statistics
BertViz [58]		✓	✓	✓			✓	✓	✓	✓
Clark [19]		✓	✓	✓		✓	✓	✓	✓	✓
VisBERT [59]		✓	✓	✓			✓	✓	✓	✓
ExBERT [17]	✓	✓	✓	✓	✓			✓	✓	✓
AttViz [60]	✓	✓	✓	✓	✓			✓	✓	✓
Kobayashi [61]		✓	✓	✓				✓	✓	✓

can highlight where a particular error comes from (e.g., corpus, training procedure, layer, etc). While it can be argued that neuron or layer views should be included in the components section, the way these views are currently implemented suggests they are rather summaries, as neurons or layers can have different states.

We have eventually decided against including chart types in Table II, as each visualization suite or paper included some novel visualization types besides attention maps (matrix charts), parallel coordinates or line and bar charts.

In our view, none of the examined visualization systems has yet managed to examine all the facets of the Transformers. This is perhaps due to the fact that this area is relatively new and there is no consensus on what needs to be visualized. While it is quite obvious that individual neurons or attention maps (regardless of if they are averaged or not) are useful, and it is best to visualize them, the same can not be said about the training corpora today, as only a small number of systems considered this aspect (e.g., [60] and [17]). This is not really ideal, as lots of errors might simply come from a bad corpora, but researchers might simply not be aware of them [67]. Errors themselves are only seriously discussed in a single publication [19]. ExBERT [17] and AttViz [60] deserve a special mention here, as they combine different views on the corpus, embeddings and attention maps in order to provide a holistic image of a Transformer model.

A study that looks at the similarity and stability of neural representations in language models and brains [68] shows that combining predictive modelling with Representation Similarity Analysis (RSA) techniques can yield promising results. This article deserves a special mention as it can be included in both focused and holistic visualizations. Their visualizations are rather basic in terms of design, but they contain lots of insights, as for example one of the tables they produced showcases the RSA results for various layers of multiple models like BERT, Elmo and others. These kind of analyses are rather new and we hope they will become more common in the next years, as they might help us clarify which language models are more similar to the human brains.

BERT Lang Street[17] [69] showcases a simple dashboard for tracking the progress of multilingual BERT models for various languages, tasks and datasets. However, since the results are generally presented as tables, it is not included in any of the two Transformer visualization categories we have identified.

Our analysis suggests that in some areas (e.g., corpus visualizations, inputs/outputs, error analysis) there is definitely a lot of room for improvements. Future frameworks should definitely consider these areas and possibly add new ones, as this field is rather young.

IV. Discussion

There are several available options for understanding the inner workings, as well as the results produced by Transformer networks. Each of them have their own advantages or shortcomings, briefly discusses in the following.

Model-agnostic tools like the XAI libraries or the hyperparameter optimization and benchmarking tools can be used with a variety of networks. Due to this, model-agnosticism the visualization skills learned while debugging a certain network (e.g., a Convolutional Neural Network) will be easily transferred to debugging and optimizing other networks (e.g., Recurrent Neural Networks). By building a transferable set of skills, users might be more reluctant to try model-specific approaches, like those from the second category discussed in this paper. Some of these model-agnostic tools might be more susceptible to various adversarial attacks (e.g., as already mentioned LIME and SHAP are easily fooled by simple adversarial strategies [35]), whereas some other tools might not provide us with sufficiently advanced visualizations to match our needs (e.g., some of the dashboards included in the hyperparameter optimization and benchmarking subsection provide only basic charts). If the users are already comfortable with some of these options, then they might well be their Swiss-Army knife for any scenario, whereas if they will need specific visualization scenarios (e.g., visualize a specific attention map), it is possible that they will eventually use the Transformer focused visualizations.

We have started our exploration of Transformer visualizations with a short set of tutorials. While they are definitely great at explaining the inner working of these models, such tutorials should only be seen as starting points in our debugging, optimizing and visualization journey. Some of the most useful tools discovered during this exploration include: visualization of attention maps (e.g., [19]) or embeddings [18], parallel coordinates plots [19], and the inclusion of corpus views from ExBERT [17].

Current generation of pre-trained language models based on Transformers [9] was shown to be relatively good at picking up syntactic cues like noun modifiers, possessive pronouns, prepositions or co-referents [19] and semantic cues like entities and relations [70], but has not performed well at capturing different perspectives [11], global context [71] or relation extraction [72]. This may be due to the fact that biases can also be already included in the embeddings and later propagated to the downstream tasks [73].

The two large classes of Transformer visualizations we examined (focused on explaining Transformer topics or holistic) are proof that the field is extremely dynamic. While many of the articles focused on explaining Transformer topics like attention or information probing tend to use classic statistical chart types (e.g., bar charts, line charts, PCA, or Pearson correlation charts), we do not consider this a bad thing as we are still in the exploration phase of this technology. Some of these articles also showcase new charts like attention graphs or attention maps.

The second class of visualizations includes tools like BertViz [18], AttViz [60], VisBERT [59] or ExBERT [17], that aim to visualize the entire lifecycle of a neural network from corpora and inputs to the model outputs mainly through following the information flow through the various components. They also offer detailed statistics for neurons or network layers. Since most of the models included in this category are rather new, it is expected that this class will expand in the next years.

One important thing to note about visualization methods is that they can easily be imported into other domains. The averaged attention heatmaps used by Vig in his causal mediation analysis for NLP [50], for example, were later reused for protein analysis in biology [74]. Similarly, attention maps [18] developed for BERT models are now used in a wide variety of disciplines, from vision and speech to biology or genetics.

The end goal of future visualization frameworks should be to visualize the entire lifecycle of the Transformer models, from inputs and data sources (e.g., training corpora), to embeddings or attention maps, and finally outputs. In the end, errors observed when creating such models can come from a variety of sources: from the text corpora, from some random network layer or even from some external Knowledge Graph that might feed some data into the model. Tracking such errors would be extremely expensive without visualizations.

V. Conclusion

While the current wave of visualizations aspire to be model-agnostic, we think the directions opened by the various Transformer / BERT visualizations are worthy of expanding upon. In fact, since this is an ubiquitous architecture today that has also

[17]Available at: https://bertlang.unibocconi.it/

branched from NLP into areas like semantic video processing, natural language understanding (e.g., speech, translation) and generation (e.g., text generation, music generation), the next generation XAI libraries will probably be built upon it. Going beyond current visualizations that are model-agnostic, future frameworks will have to provide visualization components that focus on the important Transformer components like corpora, embeddings, attention heads or additional neural network layers that might be problem-specific. By focusing on the common components from larger architectures, it should be also able to reduce the reliability of current visualizations on the underlying models. Other important features that should be included in future frameworks are the ability to summarize the model's state (e.g., through averaged attention heatmaps or similar visualization mechanisms) at various levels (e.g., neurons, layers, inputs and outputs), as well as the possibility to compare multiple settings for one or multiple models.

One interesting direction is the automated development of model specific visualizations, as more complex neural networks might also include a lot of specific components that can not always be included into more general model agnostic frameworks.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 5998–6008. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need

[2] M. Raghu and E. Schmidt, "A survey of deep learning for scientific discovery," *CoRR*, vol. abs/2003.11755, 2020. [Online]. Available: https://arxiv.org/abs/2003.11755

[3] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 4171–4186. [Online]. Available: https://doi.org/10.18653/v1/n19-1423

[4] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: http://arxiv.org/abs/1907.11692

[5] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=H1eA7AEtvS

[6] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 5754–5764. [Online]. Available: http://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding

[7] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter," *CoRR*, vol. abs/1910.01108, 2019. [Online]. Available: http://arxiv.org/abs/1910.01108

[8] N. Kitaev, L. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=rkgNKkHtvB

[9] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, "Huggingface's transformers: State-of-the-art natural language processing," *CoRR*, vol. abs/1910.03771, 2019. [Online]. Available: http://arxiv.org/abs/1910.03771

[10] W. Liu, P. Zhou, Z. Zhao, Z. Wang, Q. Ju, H. Deng, and P. Wang, "K-BERT: enabling language representation with knowledge graph," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 2020, pp. 2901–2908. [Online]. Available: https://aaai.org/ojs/index.php/AAAI/article/view/5681

[11] S. Chen, D. Khashabi, W. Yin, C. Callison-Burch, and D. Roth, "Seeing things from a different angle: Discovering diverse perspectives about claims," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 542–557. [Online]. Available: https://doi.org/10.18653/v1/n19-1053

[12] L. Hou, L. Shang, X. Jiang, and Q. Liu, "Dynabert: Dynamic BERT with adaptive width and depth," *CoRR*, vol. abs/2004.04037, 2020. [Online]. Available: https://arxiv.org/abs/2004.04037

[13] L. Wilkinson, *The Grammar of Graphics, Second Edition*, ser. Statistics and computing. Springer, 2005.

[14] M. Bostock, V. Ogievetsky, and J. Heer, "D^3 data-driven documents," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2301–2309, 2011. [Online]. Available: https://doi.org/10.1109/TVCG.2011.185

[15] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-lite: A grammar of interactive graphics," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 1, pp. 341–350, 2017. [Online]. Available: https://doi.org/10.1109/TVCG.2016.2599030

[16] D. Park, S. Kim, J. Lee, J. Choo, N. Diakopoulos, and N. Elmqvist, "Conceptvector: Text visual analytics via interactive lexicon building using word embedding," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 361–370, 2018. [Online]. Available: https://doi.org/10.1109/TVCG.2017.2744478

[17] B. Hoover, H. Strobelt, and S. Gehrmann, "exbert: A visual analysis tool to explore learned representations in transformers models," *CoRR*, vol. abs/1910.05276, 2019. [Online]. Available: http://arxiv.org/abs/1910.05276

[18] J. Vig, "A multiscale visualization of attention in the transformer model," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations*, M. R. Costa-jussà and E. Alfonseca, Eds. Association for Computational Linguistics, 2019, pp. 37–42. [Online]. Available: https://doi.org/10.18653/v1/p19-3007

[19] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What does BERT look at? an analysis of bert's attention," *CoRR*, vol. abs/1906.04341, 2019. [Online]. Available: http://arxiv.org/abs/1906.04341

[20] J. Heer, "Agency plus automation: Designing artificial intelligence into interactive systems," *Proc. Natl. Acad. Sci. USA*, vol. 116, no. 6, pp. 1844–1850, 2019. [Online]. Available: https://doi.org/10.1073/pnas.1807184115

[21] Z. Qin, F. Yu, C. Liu, and X. Chen, "How convolutional neural networks see the world - A survey of convolutional neural network visualization methods," *Math. Found. Comput.*, vol. 1, no. 2, pp. 149–180, 2018. [Online]. Available: https://doi.org/10.3934/mfc.2018008

[22] Q. Zhang and S. Zhu, "Visual interpretability for deep learning: a survey," *Frontiers Inf. Technol. Electron. Eng.*, vol. 19, no. 1, pp. 27–39, 2018. [Online]. Available: https://doi.org/10.1631/FITEE.1700808

[23] A. Nguyen, J. Yosinski, and J. Clune, "Understanding neural networks via feature visualization: A survey," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, ser. Lecture Notes in Computer Science, W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen,

and K. Müller, Eds. Springer, 2019, vol. 11700, pp. 55–76. [Online]. Available: https://doi.org/10.1007/978-3-030-28954-6_4

[24] H. Strobelt, S. Gehrmann, H. Pfister, and A. M. Rush, "Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 667–676, 2018. [Online]. Available: https://doi.org/10.1109/TVCG.2017.2744158

[25] 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," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 1, pp. 353–363, 2019. [Online]. Available: https://doi.org/10.1109/TVCG.2018.2865044

[26] D. Cashman, A. Perer, R. Chang, and H. Strobelt, "Ablate, variate, and contemplate: Visual analytics for discovering neural architectures," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 1, pp. 863–873, 2020. [Online]. Available: https://doi.org/10.1109/TVCG.2019.2934261

[27] A. P. Hinterreiter, P. Ruch, H. Stitz, M. Ennemoser, J. Bernard, H. Strobelt, and M. Streit, "Confusionflow: A model-agnostic visualization for temporal analysis of classifier confusion," *CoRR*, vol. abs/1910.00969, 2019. [Online]. Available: http://arxiv.org/abs/1910.00969

[28] C. Seifert, A. Aamir, A. Balagopalan, D. Jain, A. Sharma, S. Grottel, and S. Gumhold, "Visualizations of deep neural networks in computer vision: A survey," in *Transparent Data Mining for Big and Small Data*. Springer, 2017, pp. 123–144.

[29] V. Buhrmester, D. Münch, and M. Arens, "Analysis of explainers of black box deep neural networks for computer vision: A survey," *CoRR*, vol. abs/1911.12116, 2019. [Online]. Available: http://arxiv.org/abs/1911.12116

[30] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau, "Visual analytics in deep learning: An interrogative survey for the next frontiers," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 8, pp. 2674–2693, 2019. [Online]. Available: https://doi.org/10.1109/TVCG.2018.2843369

[31] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K. Müller, Eds., *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, ser. Lecture Notes in Computer Science. Springer, 2019, vol. 11700. [Online]. Available: https://doi.org/10.1007/978-3-030-28954-6

[32] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003. [Online]. Available: http://jmlr.org/papers/v3/guyon03a.html

[33] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should I trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, B. Krishnapuram, M. Shah, A. J. Smola, C. C. Aggarwal, D. Shen, and R. Rastogi, Eds. ACM, 2016, pp. 1135–1144. [Online]. Available: https://doi.org/10.1145/2939672.2939778

[34] S. M. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions

[35] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju, "How can we fool LIME and shap? adversarial attacks on post hoc explanation methods," *CoRR*, vol. abs/1911.02508, 2019. [Online]. Available: http://arxiv.org/abs/1911.02508

[36] S. M. Lundberg, B. Nair, M. S. Vavilala, M. Horibe, M. J. Eisses, T. Adams, D. E. Liston, D. K.-W. Low, S.-F. Newman, J. Kim *et al.*, "Explainable machine-learning predictions for the prevention of hypoxaemia during surgery," *Nature biomedical engineering*, vol. 2, no. 10, pp. 749–760, 2018.

[37] A. Fan, Y. Jernite, E. Perez, D. Grangier, J. Weston, and M. Auli, "ELI5: long form question answering," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, A. Korhonen, D. R. Traum, and L. Màrquez, Eds. Association for Computational Linguistics, 2019, pp. 3558–3567. [Online]. Available: https://doi.org/10.18653/v1/p19-1346

[38] E. Wallace, J. Tuyls, J. Wang, S. Subramanian, M. Gardner, and S. Singh, "Allennlp interpret: A framework for explaining predictions of NLP models," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019 - System Demonstrations*, S. Padó and R. Huang, Eds. Association for Computational Linguistics, 2019, pp. 7–12. [Online]. Available: https://doi.org/10.18653/v1/D19-3002

[39] A. Florea and R. Andonie, "Weighted random search for hyperparameter optimization," *Int. J. Comput. Commun. Control*, vol. 14, no. 2, pp. 154–169, 2019. [Online]. Available: https://doi.org/10.15837/ijccc.2019.2.3514

[40] R. Andonie, "Hyperparameter optimization in learning systems," *J. Membr. Comput.*, vol. 1, no. 4, pp. 279–291, 2019. [Online]. Available: https://doi.org/10.1007/s41965-019-00023-0

[41] J. Heinrich and D. Weiskopf, "Parallel coordinates for multidimensional data visualization: Basic concepts," *Comput. Sci. Eng.*, vol. 17, no. 3, pp. 70–76, 2015. [Online]. Available: https://doi.org/10.1109/MCSE.2015.55

[42] P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, M. Elibol, Z. Yang, W. Paul, M. I. Jordan, and I. Stoica, "Ray: A distributed framework for emerging AI applications," in *13th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2018, Carlsbad, CA, USA, October 8-10, 2018*, A. C. Arpaci-Dusseau and G. Voelker, Eds. USENIX Association, 2018, pp. 561–577. [Online]. Available: https://www.usenix.org/conference/osdi18/presentation/nishihara

[43] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, and I. Stoica, "Tune: A research platform for distributed model selection and training," *CoRR*, vol. abs/1807.05118, 2018. [Online]. Available: http://arxiv.org/abs/1807.05118

[44] T. Kohonen, *Self-Organizing Maps*, ser. Springer Series in Information Sciences. Springer, 1995, vol. 30. [Online]. Available: https://doi.org/10.1007/978-3-642-97610-0

[45] B. Mandelbrot, *Fractal Geometry of Nature*. W. H. Freeman, 1977.

[46] S. Jain and B. C. Wallace, "Attention is not explanation," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 3543–3556. [Online]. Available: https://doi.org/10.18653/v1/n19-1357

[47] S. Wiegreffe and Y. Pinter, "Attention is not not explanation," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Association for Computational Linguistics, 2019, pp. 11–20. [Online]. Available: https://doi.org/10.18653/v1/D19-1002

[48] Y. Hao, L. Dong, F. Wei, and K. Xu, "Self-attention attribution: Interpreting information interactions inside transformer," *CoRR*, vol. abs/2004.11207, 2020. [Online]. Available: https://arxiv.org/abs/2004.11207

[49] S. Abnar and W. H. Zuidema, "Quantifying attention flow in transformers," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds. Association for Computational Linguistics, 2020, pp. 4190–4197. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.385/

[50] J. Vig, S. Gehrmann, Y. Belinkov, S. Qian, D. Nevo, Y. Singer, and S. M. Shieber, "Causal mediation analysis for interpreting neural NLP: the case of gender bias," *CoRR*, vol. abs/2004.12265, 2020. [Online]. Available: https://arxiv.org/abs/2004.12265

[51] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov, "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, A. Korhonen, D. R. Traum, and L. Màrquez, Eds. Association for Computational Linguistics, 2019, pp. 5797–5808. [Online]. Available: https://doi.org/10.18653/v1/p19-1580

[52] E. Voita, R. Sennrich, and I. Titov, "The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7,*

2019, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Association for Computational Linguistics, 2019, pp. 4395–4405. [Online]. Available: https://doi.org/10.18653/v1/D19-1448

[53] E. Voita and I. Titov, "Information-theoretic probing with minimum description length," *CoRR*, vol. abs/2003.12298, 2020. [Online]. Available: https://arxiv.org/abs/2003.12298

[54] I. Tenney, D. Das, and E. Pavlick, "BERT rediscovers the classical NLP pipeline," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, A. Korhonen, D. R. Traum, and L. Màrquez, Eds. Association for Computational Linguistics, 2019, pp. 4593–4601. [Online]. Available: https://doi.org/10.18653/v1/p19-1452

[55] P. Dufter and H. Schütze, "Identifying necessary elements for bert's multilinguality," *CoRR*, vol. abs/2005.00396, 2020. [Online]. Available: https://arxiv.org/abs/2005.00396

[56] S. Eger, J. Daxenberger, and I. Gurevych, "How to probe sentence embeddings in low-resource languages: On structural design choices for probing task evaluation," *CoRR*, vol. abs/2006.09109, 2020. [Online]. Available: https://arxiv.org/abs/2006.09109

[57] Y. Song, J. Wang, Z. Liang, Z. Liu, and T. Jiang, "Utilizing BERT intermediate layers for aspect based sentiment analysis and natural language inference," *CoRR*, vol. abs/2002.04815, 2020. [Online]. Available: https://arxiv.org/abs/2002.04815

[58] J. Vig, "Visualizing attention in transformer-based language representation models," *CoRR*, vol. abs/1904.02679, 2019. [Online]. Available: http://arxiv.org/abs/1904.02679

[59] B. van Aken, B. Winter, A. Löser, and F. A. Gers, "Visbert: Hidden-state visualizations for transformers," in *Companion of The 2020 Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, A. E. F. Seghrouchni, G. Sukthankar, T. Liu, and M. van Steen, Eds. ACM / IW3C2, 2020, pp. 207–211. [Online]. Available: https://doi.org/10.1145/3366424.3383542

[60] B. Skrlj, N. Erzen, S. Sheehan, S. Luz, M. Robnik-Sikonja, and S. Pollak, "Attviz: Online exploration of self-attention for transparent neural language modeling," *CoRR*, vol. abs/2005.05716, 2020. [Online]. Available: https://arxiv.org/abs/2005.05716

[61] G. Kobayashi, T. Kuribayashi, S. Yokoi, and K. Inui, "Attention module is not only a weight: Analyzing transformers with vector norms," *CoRR*, vol. abs/2004.10102, 2020. [Online]. Available: https://arxiv.org/abs/2004.10102

[62] E. Reif, A. Yuan, M. Wattenberg, F. B. Viégas, A. Coenen, A. Pearce, and B. Kim, "Visualizing and measuring the geometry of BERT," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 8592–8600. [Online]. Available: http://papers.nips.cc/paper/9065-visualizing-and-measuring-the-geometry-of-bert

[63] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "VL-BERT: pre-training of generic visual-linguistic representations," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=SygXPaEYvH

[64] J. Hewitt and C. D. Manning, "A structural probe for finding syntax in word representations," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 4129–4138. [Online]. Available: https://doi.org/10.18653/v1/n19-1419

[65] R. H. Maudslay, J. Valvoda, T. Pimentel, A. Williams, and R. Cotterell, "A tale of a probe and a parser," *CoRR*, vol. abs/2005.01641, 2020. [Online]. Available: https://arxiv.org/abs/2005.01641

[66] A. Conneau, G. Kruszewski, G. Lample, L. Barrault, and M. Baroni, "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, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, I. Gurevych and Y. Miyao, Eds. Association for Computational Linguistics, 2018, pp. 2126–2136. [Online]. Available: https://www.aclweb.org/anthology/P18-1198/

[67] A. Brasoveanu, G. Rizzo, P. Kuntschik, A. Weichselbraun, and L. J. B. Nixon, "Framing named entity linking error types," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*, N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, Eds. European Language Resources Association (ELRA), 2018. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2018/summaries/612.html

[68] N. van der Heijden, S. Abnar, and E. Shutova, "A comparison of architectures and pretraining methods for contextualized multilingual word embeddings," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 2020, pp. 9090–9097. [Online]. Available: https://aaai.org/ojs/index.php/AAAI/article/view/6443

[69] D. Nozza, F. Bianchi, and D. Hovy, "What the [mask]? making sense of language-specific BERT models," *CoRR*, vol. abs/2003.02912, 2020. [Online]. Available: https://arxiv.org/abs/2003.02912

[70] X. Han, T. Gao, Y. Yao, D. Ye, Z. Liu, and M. Sun, "Opennre: An open and extensible toolkit for neural relation extraction," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019 - System Demonstrations*, S. Padó and R. Huang, Eds. Association for Computational Linguistics, 2019, pp. 169–174. [Online]. Available: https://doi.org/10.18653/v1/D19-3029

[71] D. Wadden, U. Wennberg, Y. Luan, and H. Hajishirzi, "Entity, relation, and event extraction with contextualized span representations," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Association for Computational Linguistics, 2019, pp. 5783–5788. [Online]. Available: https://doi.org/10.18653/v1/D19-1585

[72] T. Gao, X. Han, H. Zhu, Z. Liu, P. Li, M. Sun, and J. Zhou, "Fewrel 2.0: Towards more challenging few-shot relation classification," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Association for Computational Linguistics, 2019, pp. 6249–6254. [Online]. Available: https://doi.org/10.18653/v1/D19-1649

[73] H. Gonen and Y. Goldberg, "Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them," in *Proceedings of the 2019 Workshop on Widening NLP@ACL 2019, Florence, Italy, July 28, 2019*, A. Axelrod, D. Yang, R. Cunha, S. Shaikh, and Z. Waseem, Eds. Association for Computational Linguistics, 2019, pp. 60–63. [Online]. Available: https://www.aclweb.org/anthology/W19-3621/

[74] J. Vig, A. Madani, L. R. Varshney, C. Xiong, R. Socher, and N. F. Rajani, "Bertology meets biology: Interpreting attention in protein language models," *CoRR*, vol. abs/2006.15222, 2020. [Online]. Available: https://arxiv.org/abs/2006.15222

[75] I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, 2017.

[76] J. Burstein, C. Doran, and T. Solorio, Eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019. [Online]. Available: https://www.aclweb.org/anthology/volumes/N19-1/

[77] H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, 2019. [Online]. Available: http://papers.nips.cc/book/advances-in-neural-information-processing-systems-32-2019

[78] *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: https://openreview.net/group?id=ICLR.cc/2020/Conference

[79] K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. Association for Computational Linguistics, 2019. [Online]. Available: https://aclweb.org/anthology/volumes/D19-1/

[80] A. Korhonen, D. R. Traum, and L. Màrquez, Eds., *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. Association for Computational Linguistics, 2019. [Online]. Available: https://www.aclweb.org/anthology/volumes/P19-1/

[81] *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 2020. [Online]. Available: https://www.aaai.org/Library/AAAI/aaai20contents.php

[82] S. Padó and R. Huang, Eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019 - System Demonstrations*. Association for Computational Linguistics, 2019. [Online]. Available: https://aclweb.org/anthology/volumes/D19-3/