Towards Explaining Shortcut Learning Through Attention Visualization and Adversarial Attacks

Pedro Gonçalo Correia(✉) and Henrique Lopes Cardoso

Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
pedrogoncalocorreia@hotmail.com, hlc@fe.up.pt

Abstract. Since its introduction, the attention-based Transformer architecture has become the *de facto* standard for building models with state-of-the-art performance on many Natural Language Processing tasks. However, it seems that the success of these models might have to do with their exploitation of dataset artifacts, rendering them unable to generalize to other data and vulnerable to adversarial attacks. On the other hand, the attention mechanism present in all models based on the Transformer, such as BERT-based ones, has been seen by many as a potential way to explain these deep learning models: by visualizing attention weights, it might be possible to gain insights on the reasons behind these opaque models' decisions. This paper introduces Attentive-BERT, an interactive attention weights visualization tool for diagnosing BERT-based models, focusing on explaining the occurrence of shortcut learning. The distinctive feature of this tool is enabling the visual comparison of attention weights before and after a change to the model's input, in order to visually analyse adversarial attacks. Some illustrations of this use case are explored in this paper.

Keywords: Attention visualization tool · BERT · Shortcut learning · Adversarial attacks

1 Introduction

Over the past few years, the field of Natural Language Processing (NLP) has seen remarkable progress, with language representation models exhibiting ever-increasing gains on existing benchmarks. In fact, since its introduction, the Transformer architecture [37] has served as a base for many models that have achieved state-of-the-art performance on many NLP tasks [1,7,35].

This apparent success raises the question: are these models making progress towards being able to understand language like a human does? Unfortunately, studies have shown that these models often exploit biases in the dataset, leveraging spurious features in order to achieve the reported performances [12,14,27].

This research is supported by Calouste Gulbenkian Foundation and by Fundação para a Ciência e a Tecnologia, through LIACC (UIDB/00027/2020).

L. Iliadis et al. (Eds.): EANN 2023, CCIS 1826, pp. 558–569, 2023.
https://doi.org/10.1007/978-3-031-34204-2_45

This phenomenon, known as shortcut learning, undermines the models' capability to generalize to data that does not present such spurious artifacts. This reveals not only that the model is unable to actually grasp the meaning of its inputs, but also that it is susceptible to unexpected failures, for example, in real world applications [12]. Thus, being able to diagnose a model to determine the occurrence of shortcut learning is desirable.

One way to assess whether the model is relying on undesired characteristics in the data is by testing it against adversarial attacks: small perturbations made to inputs intentionally designed to deceive a machine learning model into outputting an incorrect answer, without changing the semantics of the input [13]. If the model is changing its prediction based on differences that should be irrelevant to the meaning of the input and, by extension, the output result, then it is likely that the model is not learning to perform well in the task for the right reasons. In other words, successful adversarial attacks are a sign that shortcut learning may be occurring.

However, when an adversarial attack succeeds, the reason for the model's failure is not obvious. Deep learning algorithms are often labelled black box models [6,18,22] because their complexity and huge amount of parameters makes them extremely hard to interpret by humans. Despite that, many efforts have been made seeking to understand the language representation models' inner workings [8,15,19]. One topic of high interest for studying the explainability of these models is the visualization and interpretation of the attention mechanism [16,33,40], one of the core components of the Transformer and of the models that are based on it. As a high-level description, this mechanism consists of assigning non-negative weights to each input token, according to its relevance, and then doing the weighted sum of the tokens' representations based on those weights [10].

This paper presents AttentiveBERT[1], an interactive visualization tool of attention weights for diagnosing the Transformer-based model BERT [7] and its variants, with a focus on the analysis of adversarial attacks. This tool enables the visualization of attention weights for a single input, as well as comparing the attention weights of two inputs. It is possible to try to generate an adversarial attack for a given input or, for convenience, to include a pre-generated list of adversarial attacks to visualize. Additionally, this tool includes informative and didactic content to explain the main relevant concepts, such as the attention mechanism, shortcut learning and adversarial attacks.

The rest of the paper is structured as follows. In Sect. 2 we review relevant concepts for understanding the attention mechanism, shortcut learning, and related work concerning attention weights visualization. In Sect. 3 we describe in detail the AttentiveBERT tool, exposing several use cases. In Sect. 4 we show how to generate adversarial attacks in the tool and how differences can be observed. Section 5 concludes.

[1] Available on Github: https://github.com/Goncalerta/AttentiveBERT.

2 Background

2.1 Attention Mechanism

Just like humans don't tend to process the entirety of information received by their senses at once, the idea that machine learning models could focus their attention selectively on parts of their inputs first came from computer vision [10, 25]. Soon after that, this concept was introduced in the context of NLP for the task of machine translation [2], with the proposal of an encoder-decoder architecture that employed an *attention* mechanism in its decoder to decide which parts of the input state it should give more relevance to.

A prominent model largely based on attention mechanisms is the Transformer [37], an encoder-decoder architecture initially proposed for the task of machine translation that dispenses recurrence and convolutions (as used in recurrent and convolutional neural networks, the prior state-of-the-art architectures). Each encoder of the Transformer has six attention heads, which independently calculate the *self-attention* from the output of the previous encoder (or the word embeddings for the first encoder) so that the results are then combined. In short, in the *self-attention* mechanism, each input token becomes a weighted mean of all input tokens, where the weights given to each pair of tokens are known as *attention weights* and are calculated by the model.

Many recent models that have achieved state-of-the-art performance in NLP have been based on the Transformer architecture. One such example is the Bidirectional Encoder Representations from Transformers, best known as BERT [7]. As the name implies, this architecture only uses the stack of encoders from the Transformer architecture, discarding the decoders. BERT also adds special tokens to its input: $[CLS]$, whose corresponding output representation is to be fed into a classifier for classification tasks, $[SEP]$, to mark the end of a sentence, and $[MASK]$, to mark that a word has been masked for tasks such as masked language modeling.

2.2 Shortcut Learning

Given the impact shortcut learning may have on the usefulness of a model, it is important to study this phenomenon. Previous work has tried to explain and interpret the occurrence of shortcut learning in NLP.

Han *et al.* [15] develop a method to measure artifacts in the training data, such as lexical overlap, based on influence functions. This method is based on measuring how increasing the importance of a given training example would affect the model's prediction for the test input [18]. Intuitively, a positive influence score means that removing this training example would result in the model's confidence decreasing, while a negative influence means that it would result in the confidence increasing. If the model is exploiting a given data artifact then it is likely that it will be more influenced by training examples that also exhibit that artifact. Han *et al.* conclude that influence functions are consistent with

gradient-based saliency maps in the task of sentiment analysis, but not on natural language inference. Their analysis of influence functions on natural language inference also revealed that their model might have relied on spurious artifacts such as lexical overlap between the premise and the hypothesis.

Other analyses have shown that the performance of BERT-based models for the natural language inference task could be explained by spurious artifacts such as the presence of the words "not", "do", "is", and bigrams such as "will not" [8,14,27]. Other found artifacts in natural language inference have to do with the sentence length, with shorter sentences being associated with entailment and longer sentences with the neutral relation [14].

Branco *et al.* [5] performed experiments to study whether language representation models are learning generalizable features in commonsense tasks, or simply exploiting shortcuts. In one experiment, they remove relevant parts of the input and retrain the task, finding that in some tasks the models retained their performance, suggesting that they are solving the tasks using shortcuts.

2.3 Visualizing Attention Weights

Other than the performance gains on the models in which they are applied, attention mechanisms also have the advantage of being an instrument to try to interpret models that employ them. In fact, by visualizing attention weights, studies have tried to explain these models' decisions.

Kovaleva *et al.* [19] manually inspected self-attention heatmaps of BERT models, identifying five types of patterns: *Vertical*, in which attention is given mainly to special tokens $[CLS]$ and $[SEP]$; *Diagonal* in which attention is given mainly to the previous or next token in the input; *Vertical + Diagonal* which is a mix of the previous types; *Block*, in which the attention is mainly given to tokens in the same sentence; and *Heterogeneous* which is highly variable depending on the input, without a distinct structure. They noted that the first three types of patterns are less likely to capture interpretable linguistic features, because they only take into account adjacency of the tokens and special tokens.

Because of the potential that attention weights seem to offer to interpret language models, past work has already introduced interactive tools to visualize these weights. Lee *et al.* [21] introduced a tool to visualize and manipulate (manually and automatically) beam search trees and attention weights of neural machine translation models. Strobelt *et al.* [34] presented a tool to visually analyse each stage of the translation process of a trained sequence-to-sequence model, one of which was the attention stage. Vig [38] introduced an open-source visualization tool of Transformer-based models named BertViz. This tool provided three view modes: an attention head-view, to visualize a single attention head in a bipartite graph; a model view, which presents bipartite graphs of every head in every layer of the model in a tabular form; and a neuron view, which shows how the query and key learned parameters produce the attention weights. However, the tool only supports the visualization for a single input at a time, while in the context of adversarial attacks it is desirable to compare the difference in the attention patterns of two almost identical, yet different inputs.

3 AttentiveBERT

In order to study the attention patterns in adversarial attacks, we have developed a visualization tool named AttentiveBERT. This is an interactive tool designed to visualize the attention weights of BERT-based models, both for a single input and for comparing pairs of inputs. Although the tool is focused on the study of adversarial attacks, it can also be used for visualizing attention weights in other contexts.

3.1 Tasks

AttentiveBERT is designed to work with three different, but similar classification tasks. All three tasks deal with pairs of sentences as input, classifying their relation with one of three labels, depending on the relation between the sentences.

The first one is *Natural Language Inference* (NLI) [24], also known as *Recognizing Textual Entailment* (RTE). It consists of, given a premise and an hypothesis, determining whether the relation between them is of entailment, contradiction or neutral.

The second task is *Argumentative Relation Identification* (ARI) [28,31]. This is a subtask of *Argument Mining* that, given two Argumentative Discourse Units (ADU), aims at classifying whether the first (known as source ADU) supports, attacks or has no relation to the second (known as target ADU).

The third task is *Fact Verification* (FV), also known as *Claim Verification* (CV). This is a subtask of *Fact Extraction and VERification* (FEVER) [36]. It can be formulated as, given a claim and evidence, whether that evidence supports or refutes the claim, or rather if there is not enough information to determine the veracity of the claim [3,8].

3.2 Models and Configuration

The tool is designed to work with any BERT-based model available on HuggingFace Transformers [41]. For that, a JSON configuration file for each model must be included in the configuration folder before running the tool. This configuration includes the source for the model, the task, an array of pre-generated adversarial attacks, and optional metadata.

For illustration purposes, the tool includes by default the configuration for a case-sensitive DistilBERT [32] model that was fine-tuned to the NLI task with the Stanford Natural Language Inference (SNLI) dataset [4].[2] This is the model analysed in this paper.

3.3 Single Input Visualization

In AttentiveBERT, it is possible to select one of the available tasks and models and run the model for an inputted pair of sentences, observing the result.

[2] https://huggingface.co/boychaboy/SNLI_distilbert-base-cased.

The visualizer (see Fig. 1) is composed of three main components. The first is a *confidence bar* on the top, showing the model's predicted label and its confidence for each label, in percentage. The second component is an *attention weights heatmap* on the left, showing, for every pair of tokens in the input, how much weight is being attributed to that pair. The higher the intensity of blue, the more weight is being attributed, while the color white represents a weight of zero. The third component is an *attention head selector* on the right. This is a matrix of buttons to select the attention head to visualize. The x-axis exposes the heads and the y-axis the model's layers. Instead of visualizing a single head's attention weights, it is also possible to visualize average attention weights over multiple heads or layers. For that, the axis label itself may be selected. If the average attention weight over all heads and layers in the model is desired, the circular button in the bottom left of this component may be selected.

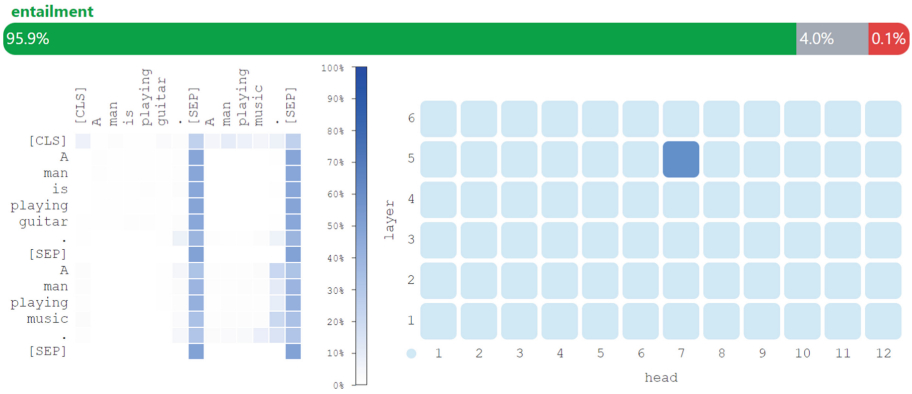

Fig. 1. Visualization of the attention weights of the 7th head on the 5th layer of a model for the premise "*A man is playing guitar.*" and hypothesis "*A man playing music.*".

3.4 Compare Inputs Visualization

In order to compare the attention weights before and after an adversarial attack, it is desirable to directly compare the attention weights of two inputs. For that, the tool offers the possibility of running the model with two different inputs, comparing the attention weights of both inputs in a single heatmap.

The resulting visualizer (see Fig. 2) is similar to the one described for single inputs. On top of the matrix of buttons, two new radio buttons are shown, to select the comparison mode. In "Compare heatmap" mode, each cell is divided in two, presenting the weight for the first input on its left side and the weight for the second input on its right side. In "Difference heatmap" mode, each cell presents the subtraction of the weights, with a green color for positive values and red for negative values, the intensity of the color denoting the absolute value.

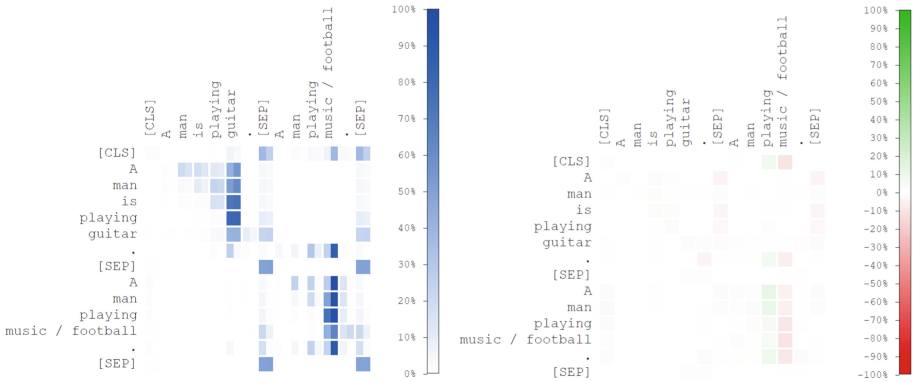

Fig. 2. Comparison of the attention weights for the premise "*A man is playing guitar.*" and hypotheses "*A man playing music.*" and "*A man playing football.*". On the left: "Compare heatmap" mode on the 4th head on the 4th layer. On the right: "Difference heatmap" mode for the average attention weights of the heads of the 6th layer.

The heatmap axes includes the tokens of both inputs, highlighting which tokens were changed, inserted or removed. In order to determine the differences between both inputs, an adaptation of the Wagner-Fischer dynamic programming algorithm for the edit distance of two strings [39] was used. The algorithm was adapted to look at tokens instead of individual characters and to disallow changes to the special tokens $[CLS]$ and $[SEP]$.

4 Generating Adversarial Attacks

AttentiveBERT can generate adversarial attacks for a given input using many techniques proposed in the literature, namely TextFooler [17], BERT-attack [23], BAE [11], the greedy algorithm proposed by Kuleshov *et al.* [20], PWWS [29], InputReduction [9], and Checklist [30]. The respective implementations offered by Textattack [26] were used. In the case of BERT-attack, the number of candidates was reduced to 8, given that the original value was unreasonably slow without significant gains. It is also possible to compose a list of pre-generated attacks, which AttentiveBERT can show in a tabular form.

By visually comparing the attention weights of the model given the input before and after the adversarial attacks, it is possible to notice interesting patterns.

For instance, in some cases where the model shifts prediction there is a shift in the average attention given to a token in favor of other tokens. This is the case with Fig. 3. On the left, a shift can be seen from the token "*summer*" to the token "*##arel*" and the prediction goes from contradiction (94%) to entailment (48%). Since the premise talks about a man in winter clothing, it is possible that having less attention in "*summer*" is responsible for the model changing the prediction. However, it is important to note that the the word

"*apparel*" might not have appeared often in the training set, as suggested by its fragmented tokenization. On the right, the model changes the prediction from neutral (86.4%) to entailment (75%). A decrease in attention to the changed token can be seen, which might explain the change in the prediction. In fact, the tokens "*selling*"/"*gathering*" are the only ones in the hypothesis that are not implied by the premise, so they are the relevant tokens to conclude that the correct label is neutral.

Fig. 3. On the left: Average attention weights difference over all heads for an example adversarial attack with premise "*A person wearing a straw hat, standing outside working a steel apparatus with a pile of coconuts on the ground.*" and hypotheses "*A person is selling coconuts.*" and "*A person is gathering coconuts.*". On the right: Comparison of the average attention weights difference over all heads for the pair of inputs with premise "*A person dressed in white and black winter clothing leaps a narrow, water-filled ditch from one frost-covered field to another, where a female dressed in black coat and pants awaits.*" and hypotheses "*The man is dressed in summer clothing.*" and "*The man is dressed in summer apparel.*", respectively. (some irrelevant tokens were omitted to reduce the size of the figure)

Another interesting finding is that, in this model, the 3rd head of the 2nd layer seemed to be specialized in, for each token, paying attention to tokens that appear later in the input and have a similar meaning. This may be relevant in explaining some adversarial attacks, as shown in Fig. 4, where the difference in predictions is, respectively: entailment (97.7%) to neutral (73.7%); entailment (97.2%) to contradiction (48.4%); neutral (97.3%) to contradiction (80.7%). In the first example, the words "*playground*" and "*play*" don't produce a strong connection as would be expected. The same happens in the second example, with the words "*river*" and "*wet*". In the third example, "*cup*" is much less associated with "*drinking*", "*alcoholic*" and "*beverage*" than "*beverage*" is. These weaker associations may be the cause of the model's wrong predictions.

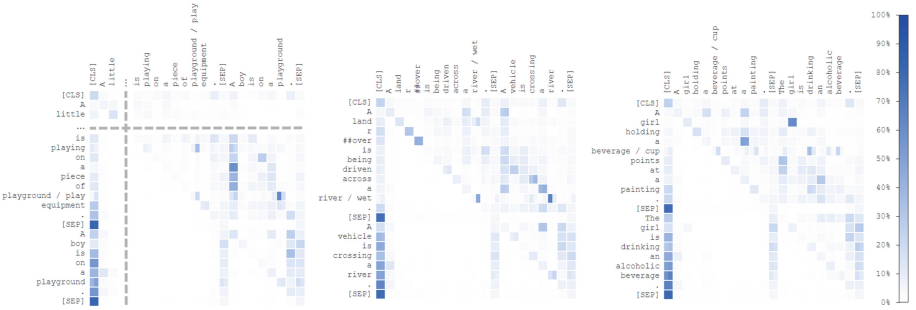

Fig. 4. Comparison of attention weights on the 3rd head of the 2nd layer of the model, before and after small changes to the inputs. From left to right: "*A little boy in a gray and white striped sweater and tan pants is playing on a piece of playground/play equipment.* (Color figure online)" and "*A boy is on a playground.*", "*A land rover is being driven across a river/wet.*" and "*A vehicle is crossing a river.*", "*A girl holding a beverage/cup points at a painting.*" and "*The girl is drinking an alcoholic beverage.*". (some irrelevant tokens were omitted to reduce the size of the figure)

5 Conclusion

In this work, a new tool was developed to visually analyse the attention weights of BERT-based models. This tool allows the visualization of those weights in different attention heads and even averages of heads for a single input or to compare the differences caused by changes in the input. This tool may prove useful in future studies of BERT-based models for interpreting the model behavior, particularly in the context of adversarial attacks. Some interesting use cases of this tool were explored, exposing patterns that may be further analysed in subsequent studies.

In future work, this tool may be extended to other NLP tasks and to other Transformer-based models. Another possibility is to implement alternative ways to visualize the weights, such as bipartite graphs or the total attention given to each token by all the others.

Acknowledgements. This research is supported by the Calouste Gulbenkian Foundation and by LIACC (FCT/UID/CEC/0027/2020), funded by Fundação para a Ciência e a Tecnologia (FCT).

References

1. Chowdhery, A. et al.: PaLM: Scaling Language Modeling with Pathways. arXiv:2204.02311 (2022)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate. In: Bengio, Y., LeCun, Y. (eds.) 3rd Int. Conf. on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conf. Track Proceedings (2015)

3. Bekoulis, G., Papagiannopoulou, C., Deligiannis, N.: A Review on Fact Extraction and Verification. ACM Comput. Surv. **55**(1) (nov 2021). https://doi.org/10.1145/3485127
4. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. In: Proceedings of 2015 Conference on Empirical Methods in Natural Language Processing, pp. 632–642. ACL, Lisbon, Portugal (Sep 2015). https://doi.org/10.18653/v1/D15-1075
5. Branco, R., Branco, A., António Rodrigues, J., Silva, J.R.: Shortcutted Commonsense: Data Spuriousness in Deep Learning of Commonsense Reasoning. In: Proceedings of 2021 Conference on Empirical Methods in Natural Language Processing, pp. 1504–1521. ACL (Nov 2021). https://doi.org/10.18653/v1/2021.emnlp-main.113
6. Buhrmester, V., Münch, D., Arens, M.: Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: a survey. Mach. Learn. Knowl. Extract. **3**(4), 966–989 (2021). https://doi.org/10.3390/make3040048
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proc. 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186. ACL, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423
8. Du, M., et al.: owards Interpreting and Mitigating Shortcut Learning Behavior of NLU models. In: Proc. 2021 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 915–929. ACL (Jun 2021). https://doi.org/10.18653/v1/2021.naacl-main.71
9. Feng, S., Wallace, E., Grissom II, A., Iyyer, M., Rodriguez, P., Boyd-Graber, J.: Pathologies of Neural Models Make Interpretations Difficult. In: Proc. 2018 Conf. on Empirical Methods in Natural Language Processing, pp. 3719–3728. ACL, Brussels, Belgium (Oct-Nov 2018). https://doi.org/10.18653/v1/D18-1407
10. Galassi, A., Lippi, M., Torroni, P.: Attention in Natural Language Processing. IEEE Trans. Neural Netw. Learn. Syst. **32**(10), 4291–4308 (10 2021). https://doi.org/10.1109/tnnls.2020.3019893
11. Garg, S., Ramakrishnan, G.: BAE: BERT-based Adversarial Examples for Text Classification. In: Proc. 2020 Conf. on Empirical Methods in Natural Language Processing (EMNLP), pp. 6174–6181. ACL (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.498
12. Geirhos, R., et al.: Shortcut learning in deep neural networks. Nature Mach. Intell. **2**(11), 665–673 (11 2020). https://doi.org/10.1038/s42256-020-00257-z
13. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and Harnessing Adversarial Examples. In: Bengio, Y., LeCun, Y. (eds.) 3rd Int. Conf. on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings (2015)
14. Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S., Smith, N.A.: Annotation Artifacts in Natural Language Inference Data. In: Proc. 2018 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pp. 107–112. ACL, New Orleans, Louisiana (Jun 2018). https://doi.org/10.18653/v1/N18-2017
15. Han, X., Wallace, B.C., Tsvetkov, Y.: Explaining Black Box Predictions and Unveiling Data Artifacts through Influence Functions. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 5553–5563. ACL (Jul 2020). 10.18653/v1/2020.acl-main.492

16. Jain, S., Wallace, B.C.: Attention is not Explanation. In: Proceedings 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 3543–3556. ACL, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1357
17. Jin, D., Jin, Z., Zhou, J.T., Szolovits, P.: Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 8018–8025 (Apr 2020). https://doi.org/10.1609/aaai.v34i05.6311
18. Koh, P.W., Liang, P.: Understanding Black-Box Predictions via Influence Functions. In: Proceedings of the 34th International Conference on Machine Learning - Volume 70, pp. 1885–1894. JMLR.org (2017)
19. Kovaleva, O., Romanov, A., Rogers, A., Rumshisky, A.: Revealing the Dark Secrets of BERT. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th Int. J. Conf. on Natural Language Processing (EMNLP-IJCNLP), pp. 4365–4374. ACL, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1445
20. Kuleshov, V., Thakoor, S., Lau, T., Ermon, S.: Adversarial Examples for Natural Language Classification Problems (2018). https://openreview.net/forum?id=r1QZ3zbAZ
21. Lee, J., Shin, J.H., Kim, J.S.: Interactive Visualization and Manipulation of Attention-based Neural Machine Translation. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 121–126. ACL, Copenhagen, Denmark (Sep 2017). https://doi.org/10.18653/v1/D17-2021
22. Lei, D., Chen, X., Zhao, J.: Opening the black box of deep learning. arXiv:1805.08355 (2018)
23. Li, L., Ma, R., Guo, Q., Xue, X., Qiu, X.: BERT-ATTACK: Adversarial Attack Against BERT Using BERT. In: Proceedings 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 6193–6202. ACL (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.500
24. MacCartney, B., Manning, C.D.: Modeling Semantic Containment and Exclusion in Natural Language Inference. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pp. 521–528. Coling 2008 Organizing Committee, Manchester, UK (Aug 2008)
25. Mnih, V., Heess, N., Graves, A., Kavukcuoglu, K.: Recurrent Models of Visual Attention. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - vol. 2, pp.. 2204–2212. NIPS'14, MIT Press, Cambridge, MA, USA (2014)
26. Morris, J., Lifland, E., Yoo, J.Y., Grigsby, J., Jin, D., Qi, Y.: TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In: Proceedings 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 119–126. ACL (Oct 2020). https://doi.org/10.18653/v1/2020.emnlp-demos.16
27. Niven, T., Kao, H.Y.: Probing Neural Network Comprehension of Natural Language Arguments. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4658–4664. ACL, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1459
28. Peldszus, A., Stede, M.: Joint prediction in MST-style discourse parsing for argumentation mining. In: Proceedings of the 2015 Conference. on Empirical Methods in Natural Language Processing, pp. 938–948. ACL, Lisbon, Portugal (Sep 2015). https://doi.org/10.18653/v1/D15-1110

29. Ren, S., Deng, Y., He, K., Che, W.: Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1085–1097. ACL, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1103
30. Ribeiro, M.T., Wu, T., Guestrin, C., Singh, S.: Beyond accuracy: Behavioral testing of NLP models with CheckList. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4902–4912. ACL (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.442
31. Rocha, G., Stab, C., Lopes Cardoso, H., Gurevych, I.: Cross-lingual argumentative relation identification: from English to Portuguese. In: Proceedings of the 5th Workshop on Argument Mining, pp. 144–154. ACL, Brussels, Belgium (Nov 2018). https://doi.org/10.18653/v1/W18-5217
32. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. ArXiv abs/1910.01108 (2019)
33. Serrano, S., Smith, N.A.: Is Attention Interpretable? In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 2931–2951. ACL, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1282
34. Strobelt, H., Gehrmann, S., Behrisch, M., Perer, A., Pfister, H., Rush, A.M.: Seq2seq-Vis: a visual debugging tool for sequence-to-sequence models. IEEE Trans. Visual Comput. Graph. **25**(1), 353–363 (2019). https://doi.org/10.1109/TVCG.2018.2865044
35. Brown, T., et al.: Language Models are Few-Shot Learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) In: Advances in Neural Information Processing Systems. vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020)
36. Thorne, J., Vlachos, A., Christodoulopoulos, C., Mittal, A.: FEVER: a Large-scale Dataset for Fact Extraction and VERification. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 809–819. ACL, New Orleans, Louisiana (Jun 2018). https://doi.org/10.18653/v1/N18-1074
37. Vaswani, A., et al.: Attention is All You Need. In: Proc. Int. Conf. on Neural Information Processing Systems, pp. 6000–6010. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
38. Vig, J.: A Multiscale Visualization of Attention in the Transformer Model. In: Proceedings 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 37–42. ACL, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-3007
39. Wagner, R.A., Fischer, M.J.: The String-to-String Correction Problem. J. ACM **21**(1), 168–173 (1 1974). https://doi.org/10.1145/321796.321811
40. Wiegreffe, S., Pinter, Y.: Attention is not not Explanation. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Journal Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 11–20. ACL, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1002
41. Wolf, T., et al.: Transformers: State-of-the-Art Natural Language Processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45. ACL (Oct 2020). https://doi.org/10.18653/v1/2020.emnlp-demos.6