Towards Trustworthy Transformer Models in Machine Learning

Ahmad Rashid, Siqing Huo, Temiloluwa P. Femi-Gege and Shaikh S. A. Shimon

Abstract— Trustworthiness in Machine Learning (ML) is one of the most important factor for domain experts with low ML expertise to deploy ML solutions in critical environments such as Healthcare and Control Systems. Our work focuses on increasing reliability and understanding of state-of-the art (SOTA) Natural language Processing (NLP) models in order to deploy them to aforementioned critical environments for general domain experts with low ML expertise. Studies have shown that transformers models (current SOTA for NLP) are vulnerable to adversarial attacks. Existing visualization techniques for transformer models do not support side-by-side model or text sequence comparison to compare model behavior changes under adversarial attack. In this work, we present visualization tool which can help compare attention patterns of different transformer models, compare between original and adversarial examples and provide a mechnism for filtering attention heads. Our aim is to increase transformer model interpretability for domain experts and help them understand how the model behavior changes with slight perturbations of data and also help ML experts to take appropriate measures while training to reduce this vulnerability.

Index Terms—Machine Learning, Transformers, Visualization, Robustness, Reliability

1 Introduction

Recent advances in Machine Learning (ML) have inspired new applications and technologies such as medicine, advertising, drug discovery, self-driving and more. Machine learning platforms have seen an increased deployment to critical application sectors such as control systems or healthcare to create predictive models trained with ground truth data. The end-users of such ML platforms have domain expertise in the respective application sectors, but do not have expertise in ML domain and treat ML platform as black-boxes.Deploying ML platforms in these critical application sectors require a high-degree of testing and verification. However, recent studies have shown that lay-users trust on such black box prediction models depend on understanding the inner working of the ML model [7] [18] [23], not just the reported accuracy score of said black-box models. Trustworthy ML studies data privacy, data deletion, algorithm robustness, algorithm fairness and interpretability among other disciplines. Our work will focus on robustness and interpretability of natural language processing (NLP) predictive models.

Currently, NLP systems are mostly built using Deep Neural Networks (DNNs), which, have achieved state-of-the-art (SOTA) results on diverse applications such as machine translation, speech recognition, natural language understanding, summarization etc. However, it has been observed across various disciplines of ML that DNN classifiers are vulnerable to adversarial examples - small, label preserving perturbations of data. In machine translation, it was shown that replacing a word by its synonym can sometimes lead to a nonsensical translation in commercial systems [3]. Another study demonstrated that adding a small amount of character level noise or natural noise [2] can significantly reduce the performance SOTA translation systems. In natural language inference (NLI), [19] demonstrated that statistical models learnt syntactical heuristics that are data set specific and not general. The common theme between all these works is that they demonstrate that DNNs tend to learn surface level patterns or ‘shortcuts’ [8] in the data. Figure 1 shows the effect of changing a few characters in a paragraph can change the prediction of an NLP classifier in the medical domain.

In this work, we focus on the interpretability and robustness of the Transformer model [27], a SOTA DNN model in NLP, when evaluated on adversarial examples. Transformer models process the entire input data at once and derive context between text tokens through a mechanism called attention. The capability of processing the entire input data at once compared to other models processing one token at a time allows Transformer models to be massively parallelizable.

Whereas recent works have explored the effect of adversarial examples on transformers [13], most of the works only measure the effect on summary statistics such as accuracy and F1 score. They have no way of identifying the cause of the vulnerability and are unable to prescribe any solution. In this work, we aim to focus on improving transformer visualizations to visualize model layers under the influence of adversarial examples. We focus on visualization of attention layers to identify patterns which can help ML practitioners, non-technical domain experts and linguists understand how the transformer NLP model behavior changes with respect to adversarial attacks, and help them guide ML practitioners to improve the NLP model in question. We achieve this by providing side-by-side visual comparison of the attention layers between a text sequence and it’s adversarial example pair for a Transformer model. Moreover, we also provide side-by-side comparison of two different models and come up with mechanisms which help guide the user which attention module to look at among multiple attention layers. To the best of our knowledge, existing visualization solutions for transformer models do not provide such a visual comparison toolkit for comparing models or side-by-side sequence comparisons for attention based models. Our work can help users identify the reasons for vulnerability to adversarial attacks and provide feedback on how to train ML models to be more robust.

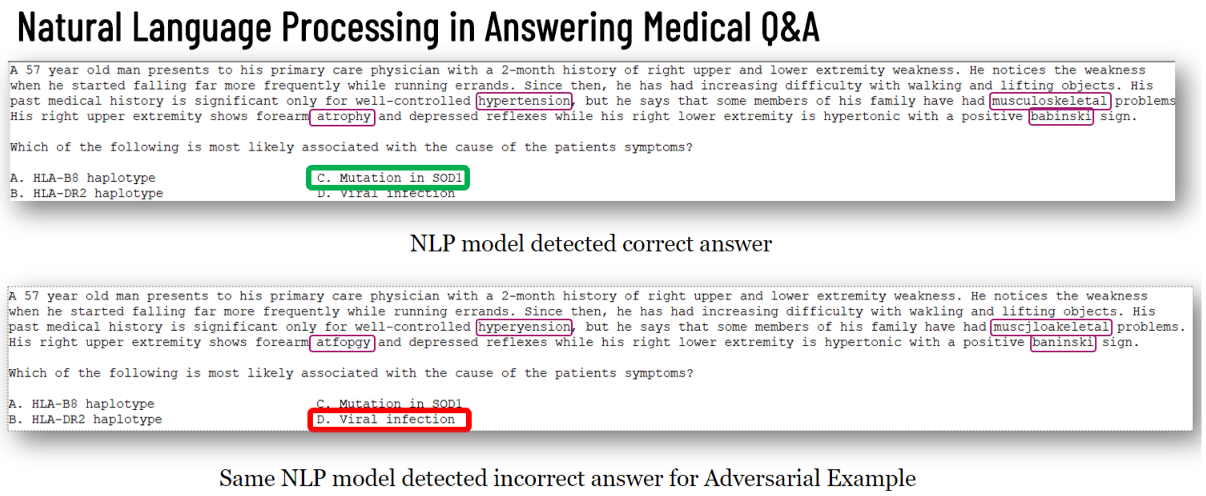
2 Related Works

2.1 Pre-trained Language Models

Attention is a mechanism to measure how different elements in two sequences are related to each other in terms of an attention or affinity score. Although attention was proposed in NLP to improve the memorization of long sequence by recurrent neural networks, Vaswani *et al.* [27] proposed a self-attention based architecture called the transformer and demonstrated that it was state-of-the-art (SOTA) on machine translation. Devlin *et al.* [6] applied transformers to language modeling (LM) and proposed Bidirectional Encoder Representation from Transformers (BERT), a masked language modeling (MLM) based architecture trained on 16 GBs of data. Traditional LMs are left to right and autoregressive whereas MLMs randomly mask a few tokens of the input sequence and train to predict them. BERT demonstrated that

- *Ahmad Rashid is with University of Waterloo and Vector Institute. E-mail: a9rashid@uwaterloo.ca.*
- *Siqing Huo is with University of Waterloo. E-mail: s2huo@uwaterloo.ca*
- *Temiloluwa P. Femi-Gege is with University of Waterloo. E-mail: tpfemige@uwaterloo.ca*
- *Shaikh S. A. Shimon is with University of Waterloo. E-mail: ssarefin@uwaterloo.ca*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx



Natural Language Processing in Answering Medical Q&A

```
A 57 year old man presents to his primary care physician with a 2-month history of right upper and lower extremity weakness. He notices the weakness
when he started falling far more frequently while running errands. Since then, he has had increasing difficulty with walking and lifting objects. His
past medical history is significant only for well-controlled hypertension, but he says that some members of his family have had musculoskeletal problems
His right upper extremity shows forearm atrophy and depressed reflexes while his right lower extremity is hypertonic with a positive babinski sign.

Which of the following is most likely associated with the cause of the patients symptoms?

A. HLA-B8 haplotype                 C. Mutation in SOD1
B. HLA-DR2 haplotype                D. Viral infection
```

NLP model detected correct answer

```
A 57 year old man presents to his primary care physician with a 2-month history of right upper and lower extremity weakness. He notices the weakness
when he started falling far more frequently while running errands. Since then, he has had increasing difficulty with wakling and lifting objects. His
past medical history is significant only for well-controlled hyperyension, but he says that some members of his family have had muscjloakeletal problems.
His right upper extremity shows forearm atfopgy and depressed reflexes while his right lower extremity is hypertonic with a positive baninski sign.

Which of the following is most likely associated with the cause of the patients symptoms?

A. HLA-B8 haplotype                 C. Mutation in SOD1
B. HLA-DR2 haplotype                D. Viral infection
```

Same NLP model detected incorrect answer for Adversarial Example

Fig. 1. Demonstrating the effect of adversarial examples on NLP classifier predicting the cause of a patient's symptoms

MLMs learn a better language representation and can be fine-tuned on different tasks, such as question answering, natural language inference, sentiment analysis etc. and achieve SOTA performance. This pre-train (LM on a large corpus) then fine-tune (on the target data and application) is now the dominant paradigm in NLP and the underlying models are referred to as pre-trained language models (PLMs). Current PLMs are composed of hundreds of billions of parameters and are trained on terabytes of data.

2.2 Adversarial Examples

In machine learning - *adversarial examples* are small data perturbations which are indiscernible to humans but can confuse a neural network classifier. This concept was first identified in computer vision [11, 16, 17]. The traditional approach to counter adversarial example based attacks is to add gradient-based perturbation on continuous input spaces [11, 16].

Adversarial examples have recently been extended into domains outside of computer vision as well, and NLP is one of such domain. Jin *et al.* proposed *TextFooler* [13] tool to measure robustness of NLP models against adversarial text attacks. This tool is capable of generating adversarial text examples, as well as replacing the most semantically important words based on model output. Others [9, 10, 22, 31] have looked into introducing adversarial examples as data augmentation in the training phase of NLP models to improve generalization and robustness. Text - unlike images is discrete, and perturbing the gradient of word embeddings can lead to nonsensical changes. Ribiero *et al.* explored rule-based methods to generate semantically meaningful adversarial text examples for probing NLP model robustness. Hybrid models (models based but condition on rules) such as Textfooler are most commonly used.

Several visualization techniques have been developed to allow users to investigate the robustness of ML models against adversarial examples. DetectorDetective [28] visualizes feature maps extracted by the selected module for benign and adversarial images side-by-side. Our design also shows side-by-side comparisons of models' internals in response to benign and adversarial inputs. *Bluff* [5] is another tool that compares activation pathway traversal for both benign and adversarial inputs. Bluff also aims to help users understand the roles of neurons in complex deep neural models which lack semantic meanings. When hovering over a neuron in *Bluff*, dataset examples (which activate the neuron highly) are shown, as well as the feature visualization for the corresponding neuron. Our work will also provide users more context through interactions to help them understand the complex internals of the models.

Visualizations of adversarial examples have been mainly studied in the context of images, general deep neural networks, and neuron activations. Our work will built on these visual, side-by-side comparisons of model's behavior with respect to benign and adversarial examples, and tailor the solution toward text and the Transformer model.

2.3 NLP Visualization

On the other hand, existing visualizations for NLP are not tailored for investigating the robustness of the models against adversarial examples. However, these general purpose visualization techniques do provide useful insights. ACTIVIS [14] enables subset-level visualization and allows users to customize subsets by applying any function to any combination of features. Texts are not as structured as tabular data, thus the functionality of customizing function to generate subset can be very helpful in producing useful subsets. We aim to support subset-level visualization and give user more freedom in defining subsets. SEQ2SEQ-VIS [26] facilitates debugging sequence-to-sequence models. It offers the ability to explore neighborhood of encoder states and decoder states, to visualize attention and probability scores of likely next words, and to visualize the top K considered options at each prediction step as a beam search tree. Users can explore the model's behavior by changing words at each step, or altering the assigned attention values. These interactions are very useful for probing the model's behavior and can be extended to exploring how an adversarial example can succeed.

2.4 Transformer Visualization

Following the success of BERT and subsequent PLMs, a number of works have focussed exclusively on visualizing transformer based PLMs. BertViz [29] builds on existing visualizations, such as matrix heatmaps, to view attention. It is capable of helping visualize models such as GPT-2 [21] which is a decoder-only model and BERT which is an encoder-only model. Bertviz has three views - *Attention-head view*, *Model view* and *Neuron view* incorporated in the visualization system. They attempt to visualize the computation of vectors using colour to show magnitude of dot products. The tool adds interactions that allow users to visualize the computed attention at various layers. However the visualization creates a lot of cognitive load when figuring out what the color and magnitude in each position of the vectors means. Also, the Bertviz tool can not load other transformer models - hence not supporting side-by-side transformer model comparisons. Also, it does not support side by side comparison of attention heads for multiple examples (for example - a text sequence and it's paired adversarial text sequence).

Aken *et al.* proposed VisBERT [1] tool designed to visualize internal state of BERT model for Question-Answering tasks. It visually identifies distinct phases in BERT transformation to offer insights about failed predictions. However, VisBERT does not provide any visual

True Sample
Premise: What does he do for a living ?
Hypothesis: What is his current occupation ?
Label: Entailment Prediction: Entailment

Adversarial Sample
Premise: What does he do for a living ?
Hypothesis: What is his current job ?
Label: Entailment Prediction: Contradiction

Fig. 2. Highlighting the differences between in domain and adversarial samples. The model makes an error on the TextFooler samples.

information on attention heads in different layers, and different transformer models can not be compared side by side. Also, similar to Bertviz - it lacks the capability of supporting side by side comparison of multiple text sequences.

Hoover *et al.* developed ExBERT [12] - a transformer model and corpus agnostic visualization tool to provide insights to contextual representation meaning by aggregating annotations of matching similar contexts. Although ExBERT tool is transformer model agnostic - like VisBERT and Bertvis it lacks the capability of comparing multiple transformer models , or multiple text sequences side by side. Also - the tool has no way of loading other fine-tuned transformer models except for preloaded BERT and GPT-2.

AttViz [25] online toolkit offers visual inspection of distribution of attention values across token sequence. It also supports integrating existing transformer libraries in PyTorch. Similar to previously mentioned transformer visualization tools - it also lacks the capability to compare multiple models or text sequences side-by-side. Also, the underlying infrastructure is tightly coupled with BERT-base transformer model, and other transformer models can not be swapped in.

Our work focuses on addressing this gap of comparing multiple BERT-based transformer models side by side, as well as comparing a text sequence and it's paired adversarial sequence against the same transformer model to visualize the changes in the attention head and neuron activity. We also incorporated the finding in [4] that attention heads attend to specific syntactic information of the input sentence to provide users more context. This visualization will help domain experts better understand NLP model behavior in the presence of adversarial examples. Although our solution is focused on NLP transformer model - it works well to support any NLP ML model with attention mechanism.

3 METHODOLOGY

To demonstrate our visualization on a concrete NLP task, we chose the important language understanding problem of NLI. In this problem typically we are given a pair of sentences, the first is the premise and the second is the hypothesis. The problem is to classify whether premise entails the hypothesis, contradicts it or is neutral. These problems are important for chatbots and dialogue systems.

3.1 Model and Dataset

Our base transformer is BERT which is a pre-trained language model based on the transformer architecture. We will take the BERT-base model (12 layers, 12 attention heads) from Google's official github repository [1].

The dataset we evaluate on is the MNLI dataset [30] a widely reported on, crowd-sourced, multi-domain NLI dataset. Figure 2 gives shoes an adversarial example where changing a single word (occupation) to its synonym (job) can lead to a change in prediction by a machine learning model.

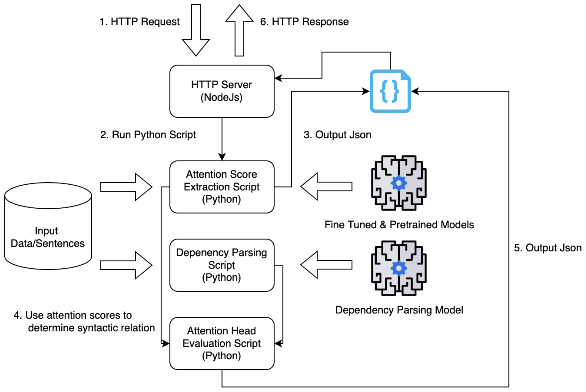

1. HTTP Request
6. HTTP Response
HTTP Server (NodeJs)
2. Run Python Script
3. Output Json
{}
Attention Score Extraction Script (Python)
Input Data/Sentences
Fine Tuned & Pretrained Models
Depencency Parsing Script (Python)
Dependency Parsing Model
5. Output Json
4. Use attention scores to determine syntactic relation
Attention Head Evaluation Script (Python)

Fig. 3. Overview of the Pipeline

We fine-tune a pre-trained BERT model on the MNLI dataset and then evaluate it on the given evaluation (or benign) set and the adversarial set. The adversarial examples are generated by a model based system called Textfooler [13]. The authors have shared 1000 samples that they generated for the MNLI dataset for BERT. We used the provided adversarial samples instead of generating them ourselves. As an additional step we manually verified the quality of the adversarial examples and then filtered out the ones which changed the label, semantics or were grammatically incorrect.

3.2 Pipeline

We present the overall architecture in the pipeline diagram in Figure 3. Building on the work done in [4] we extract the attention head scores for each model and instance which will be chosen by users. The BERT-base model, which we experimented on, has 12 layers and 12 attention heads which leads to 144 different attention heads. This can lead to a visual overload and no guidance as to where to start looking at in order to make sense of the prediction. Prior work [4] has demonstrated BERT has some understanding of syntactical relations of words.

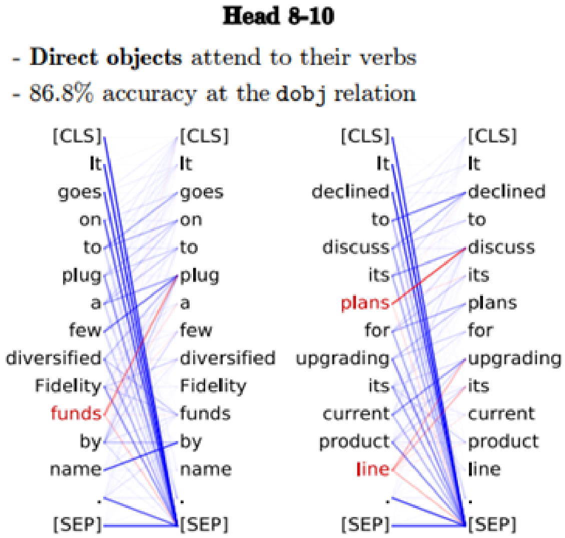


Fig. 4. Syntax relationship learnt by Attention head.

Figure 4 shows that layer 8 attention head 10 for a pre-trained model learns direct object relationships with an 86.6% accuracy. One of the main novelty of our work is that we classify which syntax information

[1] https://github.com/google-research/bert/.

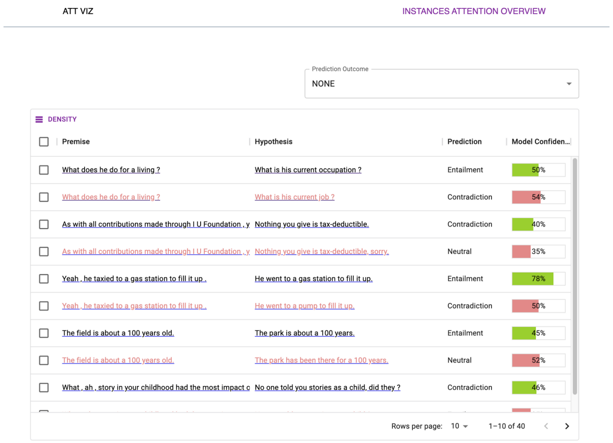
each attention head learns and use this information to select which attention head to analyze.The extracted scores for each attention head are passed into a classifier which outputs the syntactic relation which the attention head has learned.

To generate the data which we provide to our visualization, we developed a backend that consists of various scripts that are executed for a baseline pre-trained model and a fine-tuned model. Our datasets were also separated into 2 groups, adversarial examples and benign examples. We first used a script to extract the self attention scores for a given input sequence from each list. The output of this script is an object array containing the original sentence, tokenized version of the sentence, a flag indicating whether it is adversarial or benign, a pair id which facilitates associating the sentence with its corresponding adversarial/benign sample, and a self attention scores matrix. In order to generate information about the syntactic relation a head focuses on, we pass our input sentences into a dependency parser [20] which identifies the true syntactic relations for each pair of tokens. The syntactic relation labels are compared with the attention scores for an attention head and serve as the true labels that help evaluate what syntactic relation a head focuses on. The outputs from both scripts are served to the frontend to be plotted for users. Our backend is built on NodeJs while the frontend is a React application.

3.3 Visualization and Interaction

Most existing transformer visualizations [1, 15, 24, 29] show only a single instance at a given time. We will initially give a multi-instance snapshot and allow the user to explore a given instance further.

The main views of our system are divided into Instance Selection view (Figure 5), Attention Head Overview (Figure 6) and Attention Head instance View (Figure 8). Our proposed solution allows us to compare between multiple transformer models side by side. In addition, we provide the user the ability to compare adversarial example attention head patterns with benign example attention head patterns between layers. Although our base transformer model will be the BERT-base model, we want to have the option of allowing the users to upload their own preferred models to evaluate on our visual analytics toolkit for visually evaluating their own transformer models against adversarial examples.



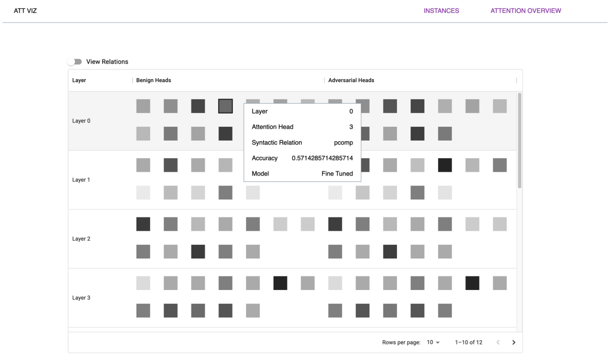
☐	Premise	Hypothesis	Prediction	Model Confiden...
☐	What does he do for a living ?	What is his current occupation ?	Entailment	50%
☐	What does he do for a living ?	What is his current job ?	Contradiction	54%
☐	As with all contributions made through I U Foundation , y	Nothing you give is tax-deductible.	Contradiction	40%
☐	As with all contributions made through I U Foundation , y	Nothing you give is tax-deductible, sorry.	Neutral	35%
☐	Yeah , he taxied to a gas station to fill it up .	He went to a gas station to fill it up.	Entailment	78%
☐	Yeah , he taxied to a gas station to fill it up .	He went to a pump to fill it up.	Contradiction	50%
☐	The field is about a 100 years old.	The park is about a 100 years.	Entailment	45%
☐	The field is about a 100 years old.	The park has been there for a 100 years.	Neutral	52%
☐	What , ah , story in your childhood had the most impact c	No one told you stories as a child, did they ?	Contradiction	46%

Fig. 5. Instance Selection View. Samples in red are adversarial

3.3.1 Instance Selection view

The Instance selection view (Figure 5) provides a table listing of all instances against the fine tuned model being evaluated. This view allows users to view the models confidence scores for the various data instances. Furthermore, the confidence scores are represented using horizontal bars where the size of the bar corresponds to the confidence level, meanwhile the colour of the bar represents whether the model predicted correctly for the provided instance. Users can sort the model's columns by the confidence score, they are also able to view the models prediction for each instance in the data. Selecting the check boxes in the first column of the table will narrow down the instances shown to only the adversarial pair of the selected example. Users can also filter the table by the prediction outcome by selecting from a dropdown with options to see examples that passed or failed.

Clicking on an instance navigates user to the Attention Head Overview view (Figure 6) where users can view attention head details for each model in question.



Layer	0
Attention Head	3
Syntactic Relation	pcomp
Accuracy	0.5714285714285714
Model	Fine Tuned

Fig. 6. Attention Head Overview

3.3.2 Attention Head Overview

This particular view provides a summary of of attention heads in each layer; and also provides a side by side comparison between attention heads of benign and adversarial examples. The view starts off by representing each attention head as black with various shades which represent how accurate the head extracts the syntactic relation in the provided sentence. Users are provided with the option to view the colored version of the overview. The colored version (as shown in Figure 7) provides a clearer view of the various syntactic relations that each attention head focuses on. The various colors represent the aspect of the language that the attention heads have learned. One quick observation from analyzing the results showed that the adversarial and benign example attention heads start to look similar as we go deeper down the layers.

In addition to using colors and shades of the attention head squares, numerical information is also rendered on a tooltip which users can view on-hover over an attention head. Hovering over a specific attention head in a model shows a tooltip that contains information about that particular attention head in a model, the layer, Syntactic relations, and how well the attention head performs at identifying the syntactic relation.

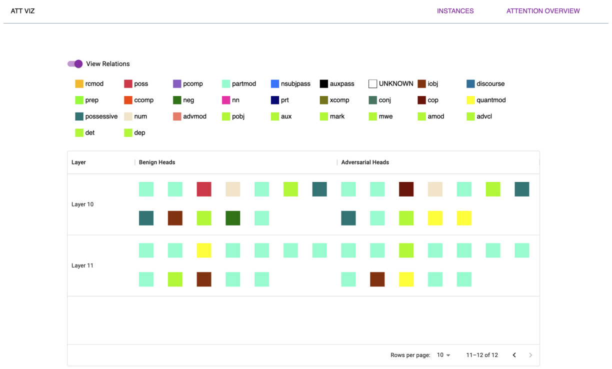


Fig. 7. Attention Head Overview - Showing Syntactic Relations

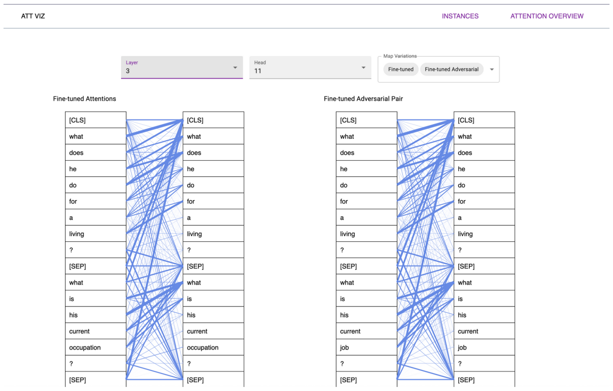


Fig. 8. Attention Head Instance View

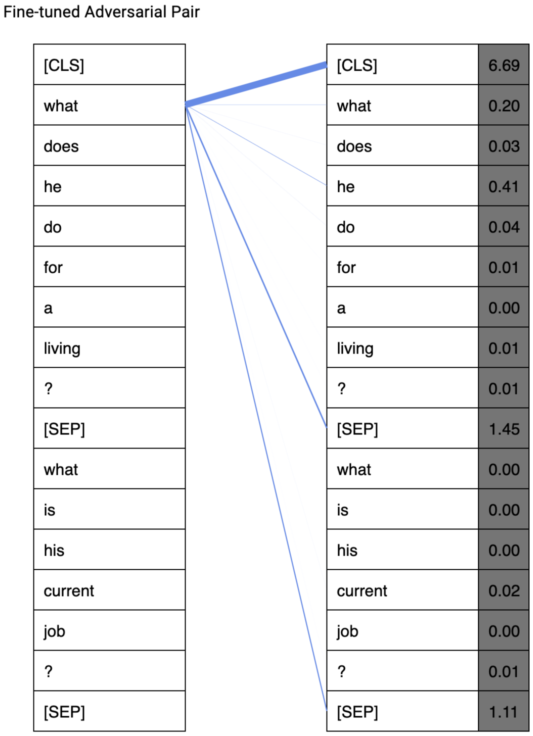


Fig. 9. Attention Head Instance View Interaction

3.3.3 Attention Head Instance View

When a user clicks on a specific attention head, they are led to another view where they are presented with the attention map for the selected instance. In this view, the users can drill down to view the attention scores for each tokens in the instance. The user can also verify whether or not the detected syntactic relation identified is correct from this view.

Within this view, users can adjust the layer they want to analyze, the attention head number, the model as well as the adversarial examples map for the selected instance. The number of attention maps a user can visualize at once is limited to 2 to try reducing clutter.

Figure 9 - When a user hovers over a token, they are presented with the attention scores for that token. The number of displayed links are also filtered down to only show links for the token being inspected. We also highlight the token with the max score to make it easier for the user to identify them.

Our aim is that our visualizations can help model designers identify the cause of vulnerability to adversarial examples and provide actionable insights which can help them improve their training algorithms. Our improvements over existing transformer visualization include a view showing multiple instances and their confidence scores, ability to compare models and a less cluttered interface.

4 Use Case

We gave our application to Alan (fictional name) who works at a large technology company which deploys PLMs for a number of different products. They frequently face the issue that models when deployed *in the wild* have significantly lower accuracy than when evaluated on clean data in-house. Alan sometimes uses adversarial examples as a proxy to test general robustness of the model.

Alan takes our model and initially likes that unlike prior work he can look at multiple examples and select ones where the model makes a mistake on the adversarial example. He selects an adversarial example which leads the model to make a mistake. He selects the same example that we have introduced in Figure 2. Next he looks at the Attention Head Overview and is curious about the different shades and the colors that he sees. He knows that the colors represent the syntactical relationships and the greyscale shades the accuracy of a particular attention head. Based on knowledge of these models Alan thinks that an attention head which can act as a participle modifier can give him some insight. On layer 11 (Figure 10) we can observe that attention head 10 is a participle modifier for both the benign and adversarial example. He selects this attention head to further analyze the attention scores.

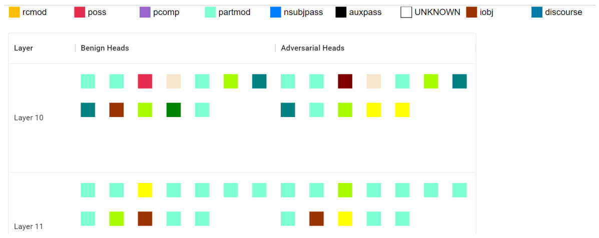


Fig. 10. Filtering attention head based on one that detects a partmod (participle modifier) relationship for both benign and adversarial sentence.

When he observes the attention (Figure 11) and hovers over the word that is changed (job to occupation) he observes that the attention score to the premise word living (synonym of job and occupation for this example) is lower in the adversarial example. On the other hand it has a stronger attention to the sentence separator. Alan observes that this is a recurring pattern and the model overfits to particular words. One remedy is to use data augmentation where the augmented data has synonyms of the nouns and verbs in the sentence.

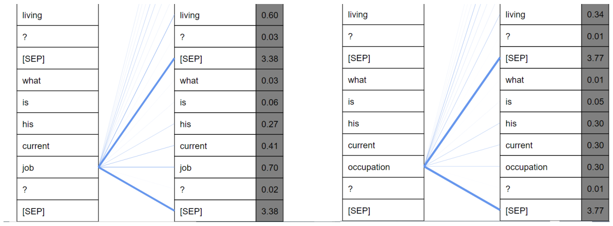


Fig. 11. Analyzing the difference in attention patterns between benign and adversarial example

Another interesting observation that Alan has is when comparing pre-trained and fine-tuned models even though only some of the parameters change, the attention patterns are completely re-arranged.

5 Discussion

The main contribution of our work is it helps domain experts such as linguists understand what transformers learn and help ML researchers gain insights of how to make the NLP model more robust. Users can compare models with different level of fine tuning and choose an appropriate level of fine tuning (whose internals appear to be more robust to adversarial examples). We provide the ability to compare different transformer models and also models based on benign and adversarial examples.

Although filtering based on syntactic patterns is useful in preliminary experiments, syntactic information is distributed over different attention heads. This information is helpful for linguists or NLP experts but to users from a different domain it is still difficult to understand where to look at. The tool itself is modular and can integrate other information such as semantic properties to filter attention heads. However going over the literature syntax looked the best option. One possible workaround is to filter out all the attention heads which achieve lower accuracy than a predefined threshold (such as 80%). This is left for future work.

Another challenge in this work is although looking at attention patterns can lead to some understanding of why an NLP model behaves in a certain way, it is difficult to convert it into modeling insight. Our user Alan could come up with a training strategy because in this case we could see changes due to a synonym. But in other adversarial examples it might not be straightforward to come up with an idea to modify training.

6 Future Work

One of the problems we noticed is that self-attention scores matrix is huge for any practical NLP model, and its size grows exponentially with the length of sentences as well. Since we need to generate, save and load these attention matrices, we have limited the number of data instances to be displayed. In the future, we can explore database technologies to make it possible to store massive amount of structured data and also to make querying it more efficient.

Another problem is that we used a trained syntactic parser as ground truth labels in our Attention Head Overview, but the prediction outputs of this model can be inaccurate. Although it would be more accurate to use human annotators to manually label the input data, our approach requires less work from the users as they do not have to supply labeled dataset. However, in the future we can add an option which allows users to supply their own labels which they deem more accurate. We can let users download our predicted labels and use it as a starting point.

It is difficult to compare different sized models as there is no one-to-one attention head interpretation. Our work currently requires the base transformers for the models under comparison to have the same structure. In the future we can explore ways to lift this requirement and allows meaningful comparisons of models based on transformers of different sizes.

The information learned by attention heads is usually distributed over multiple attention heads. Thus, looking at each attention head by itself is perhaps not ideal. In the future we can explore intelligent ways of automatically aggregating attention heads and let users customize the aggregation as well.

Acknowledgments

- Ahmad Rashid: Machine Learning code and writeup, use cases, discussion and integration
- Siqing Huo: Machine Learning Code and writeup, future works, integration and final editing.
- Temiloluwa P. Femi-Gege: Visualization design, code and writeup
- Shaikh S. A. Shimon: Abstract, Introduction and literature review

References

[1] B. v. Aken, B. Winter, A. Löser, and F. A. Gers. Visbert: Hidden-state visualizations for transformers. In *Companion Proceedings of the Web Conference 2020*, pp. 207–211, 2020.

[2] Y. Belinkov and Y. Bisk. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*, 2018.

[3] Y. Cheng, L. Jiang, and W. Macherey. Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4324–4333, 2019.

[4] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning. What does BERT look at? an analysis of bert's attention. *CoRR*, abs/1906.04341, 2019.

[5] N. Das, H. Park, Z. J. Wang, F. Hohman, R. Firstman, E. Rogers, and D. H. Chau. Bluff: Interactively deciphering adversarial attacks on deep neural networks. *CoRR*, abs/2009.02608, 2020.

[6] J. Devlin, M.-W. 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, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

[7] F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning, 2017. doi: 10.48550/ARXIV.1702.08608

[8] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[9] A. Ghaddar, P. Langlais, A. Rashid, and M. Rezagholizadeh. Context-aware adversarial training for name regularity bias in named entity recognition. *Transactions of the Association for Computational Linguistics*, 9:586–604, 2021.

[10] A. Ghaddar, P. Langlais, M. Rezagholizadeh, and A. Rashid. End-to-end self-debiasing framework for robust NLU training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 1923–1929. Association for Computational Linguistics, Online, Aug. 2021. doi: 10.18653/v1/2021.findings-acl.168

[11] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2015.

[12] B. Hoover, H. Strobelt, and S. Gehrmann. exBERT: A visual analysis tool to explore learned representations in Transformer models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 187–196. Association for Computational Linguistics, Online, July 2020.

[13] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits. Is bert really robust? natural language attack on text classification and entailment. *arXiv preprint arXiv:1907.11932*, 2019.

[14] M. Kahng, P. Y. Andrews, A. Kalro, and D. H. Chau. Activis: Visual exploration of industry-scale deep neural network models. *CoRR*, abs/1704.01942, 2017.

[15] G. Kobayashi, T. Kuribayashi, S. Yokoi, and K. Inui. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7057–7075. Association for Computational Linguistics, Online, Nov. 2020. doi: 10.18653/v1/2020.emnlp-main.574

[16] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *CoRR*, abs/1607.02533, 2016.

[17] R. Labaca-Castro, L. Muñoz-González, F. Pendlebury, G. D. Rodosek, F. Pierazzi, and L. Cavallaro. Universal adversarial perturbations for malware, 2021.

[18] Z. C. Lipton. The mythos of model interpretability. *CoRR*, abs/1606.03490, 2016.

[19] T. McCoy, E. Pavlick, and T. Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3428–3448, 2019.

[20] K. Mrini, F. Dernoncourt, T. Bui, W. Chang, and N. Nakashole. Rethinking self-attention: An interpretable self-attentive encoder-decoder parser. *CoRR*, abs/1911.03875, 2019.

[21] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[22] A. Rashid, V. Lioutas, and M. Rezagholizadeh. MATE-KD: Masked adversarial TExt, a companion to knowledge distillation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1062–1071. Association for Computational Linguistics, Online, Aug. 2021. doi: 10.18653/v1/2021.acl-long.86

[23] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should I trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938, 2016.

[24] B. Škrlj, N. Eržen, S. Sheehan, S. Luz, M. Robnik-Šikonja, and S. Pollak. Attviz: Online exploration of self-attention for transparent neural language modeling. *arXiv preprint arXiv:2005.05716*, 2020.

[25] B. Škrlj, S. Sheehan, N. Eržen, M. Robnik-Šikonja, S. Luz, and S. Pollak. Exploring neural language models via analysis of local and global self-attention spaces. In *Proceedings of the EACL Hackashop on News*

Media Content Analysis and Automated Report Generation, pp. 76–83. Association for Computational Linguistics, Online, Apr. 2021.

[26] 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, 2018.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[28] S. Vellaichamy, M. Hull, Z. J. Wang, N. Das, S. Peng, H. Park, and D. H. P. Chau. Detectordetective: Investigating the effects of adversarial examples on object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21484–21491, June 2022.

[29] J. Vig. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 37–42. Association for Computational Linguistics, Florence, Italy, July 2019. doi: 10.18653/v1/P19-3007

[30] A. Williams, N. Nangia, and S. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. 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. 1112–1122. Association for Computational Linguistics, New Orleans, Louisiana, June 2018. doi: 10.18653/v1/N18-1101

[31] C. Zhu, Y. Cheng, Z. Gan, S. Sun, T. Goldstein, and J. Liu. Freelb: Enhanced adversarial training for natural language understanding, 2020.