Understanding How BERT Learns to Identify Edits

Samuel Stevens
Ohio State University
stevens.994@osu.edu

Yu Su
Ohio State University
su.809@osu.edu

Abstract

Pre-trained transformer language models such as BERT are ubiquitous in NLP research, leading to work on understanding how and why these models work. Attention mechanisms have been proposed as a means of interpretability with varying conclusions. We propose applying BERT-based models to a sequence classification task and using the data set's labeling schema to measure each model's interpretability. We find that classification performance scores do not always correlate with interpretability. Despite this, BERT's attention weights are interpretable for over 70% of examples.

1 Introduction

Pre-trained transformer models such as BERT (Devlin et al., 2019) are currently ubiquitous within natural language processing (NLP) research and have demonstrated improvements in topics from sentiment analysis to semantic parsing (Guo et al., 2019). The widespread development and use of such models has led to an increased effort to interpret such models' decisions (Clark et al., 2019; Vashishth et al., 2019; Vig, 2019). As defined in Doshi-Velez and Kim (2017), model interpretability is "the ability [of a model] to explain or present in understandable terms to a human". Intuitively, a more interpretable model is easier to understand, debug and improve.

Interpreting modern pre-trained transformer models is difficult. First, modern deep learning models have hundreds of millions of parameters, and scale only continues to increase (Brown et al., 2020; Raffel et al., 2020). Understanding the impact of a single parameter is nearly impossible because these models are densely connected. Combined with the sheer number of parameters, manual analysis is infeasible. Secondly, while both pre-training and fine-tuning are required for state-of-the-art performance, effort has focused on alternative pre-training methods (Beltagy et al., 2019; Liu et al., 2019; Clark et al., 2020).

Previous work uses BERT's self-attention mechanism to interpret the model's predictions (Clark et al., 2019; Kovaleva et al., 2019; Vig and Belinkov, 2019). However, a body of work (Jain and Wallace, 2019; Vashishth et al., 2019) shows that models' attention mechanisms cannot be interpreted on single-sequence classification tasks.

We apply BERT and two BERT-based models (Liu et al.'s RoBERTa and Beltagy et al.'s SciBERT) to an existing sentence classification task proposed in Daudaravicius et al. (2016). We compare BERT-based models' performances with previous baselines and then use methods presented in Vashishth et al. (2019) and DeYoung et al. (2020) to evaluate BERT's interpretability in single-sequence classification tasks. We find that fine-tuning can teach BERT to recognize previously unknown patterns in natural language and that BERT is more interpretable than the attention-based models analyzed in Jain and Wallace (2019) and Vashishth et al. (2019). To summarize, the key contributions of this paper are:

- Applying pre-trained transformer models to a sequence classification task and demonstrating notable improvements.
- Quantifying BERT's interpretability on said sequence classification task using attention.
- Comparing the impacts of pre-training methods on interpretability.

2 Related Work

There is a body of work demonstrating that attention mechanisms are not faithful explanations of a model's prediction (Jain and Wallace, 2019; Serrano and Smith, 2019), especially in single-sequence classification tasks (Vashishth et al., 2019). These studies focus on attention mechanisms in LSTMs and hierarchical attention networks.

However, BERT is based on the transformer's self-attention mechanism (Vaswani et al., 2017). Vashishth et al. (2019) found that "altering weights in self-attention based models does have a substantial [negative] effect on the performance." Indeed, there are multiple studies demonstrating that BERT's attention maps are interpretable. Clark et al. (2019) found that particular attention heads within BERT learn syntactic relations such as "direct objects of verbs, determiners of nouns, objects of prepositions and objects of possessive pronouns." Vig and Belinkov (2019) inspected attention maps from both BERT and GPT-2 (Radford et al., 2019) and found that no probing is necessary to identify dependency relations.

Besides interpreting BERT's attention for error analysis and model understanding, there is increased interest in NLP models with interpretability as a goal. DeYoung et al. (2020) introduced a new data set and evaluation metric aimed at helping researchers evaluate models' explanations for predictions. BERT-based models are evaluated, but directly interpreting BERT's attention is not considered.

3 Methodology

We use the AESW task presented by Daudaravicius et al. (2016): predict if a sentence in an academic paper needs editing. Daudaravicius et al. created labeled examples from academic papers professionally edited at VTeX[1] and share training, validation and test sets. An example can be seen in Table 1. The original AESW paper contains more details.

We evaluate three different pre-trained transformer models, all based on BERT: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and SciBERT (Beltagy et al., 2019). BERT is the first pre-trained transformer model. RoBERTa has the same architecture as BERT but is pre-trained with more data (13GB vs 160GB) and demonstrates improvements over BERT on most benchmarks. SciBERT also has the same architecture as BERT but is pre-trained on 1.14M papers from `semanticscholar.org` and demonstrated state-of-the-art performance on a wide range of scientific domain NLP tasks at the time of writing.

We fine-tune each model on the AESW training set, and use validation loss to tune hyperparameters.[2] Code for reproducing our results is available at https://github.com/samuelstevens/bert-edits.

[1] https://www.vtex.lt

AESW	`<sentence sid="383.3">Stiction <del>non-linearity</del> <ins>nonlinearity</ins> thus results in non-smooth and non-convex objective maps.</sentence>`
Readable	Stiction ~~non-linearity~~**nonlinearity** thus results in non-smooth and non-convex objective maps.

Figure 1: An example sentence provided by AESW. In the following sections, we use the term "applying an edit" to mean removing the ~~strikeout~~ and adding the **bold**.

3.1 Comparison with AESW Task

We compare values for the top three models presented in Daudaravicius et al. (2016) with pre-trained transformer models in Table 1. We find that all pre-trained transformer models outperform previous traditional and neural models.

4 Discussion

To interpret pre-trained transformer models after task-specific fine-tuning, we inspect the attention maps of the initial token[3] (because it is used for classification) in the last layer (because it is the most task-specific (Kovaleva et al., 2019)). We find two trends:

1. Fine-tuning pre-trained transformer models teaches them previously unknown patterns.
2. BERT's final attention layer is more interpretable than SciBERT's or RoBERTa's.

We qualitatively analyze both trends in more depth in the following sections.

4.1 Novel Patterns

We find that pre-trained transformer models learn context-sensitive, domain-specific patterns without explicit instruction. One example pattern is avoiding contractions in academic writing (Osmond,

[2] Appendix A contains more details.

[3] We refer to all initial tokens as `[CLS]` and final tokens as `[SEP]` in the diagrams below to make comparisons between the models simpler. We also convert token-level attention to word-level attention using the procedure described in Clark et al. (2019). Appendix B contains more details.

Model	Dev Set			Test Set		
	Prec.	Rec.	F1	Prec.	Rec.	F1
CNN+LSTM	-	-	-	0.544	0.741	0.628
CNN	-	-	-	0.503	**0.779**	0.611
SVM	-	-	-	0.448	0.728	0.555
BERT$_{\text{base}}$	0.690	**0.622**	0.654	0.704	0.633	0.666
RoBERTa$_{\text{base}}$	**0.716**	0.614	**0.661**	**0.726**	0.622	**0.670**
SciBERT$_{\text{base}}$	0.705	0.617	0.658	0.715	0.627	0.668

Table 1: Performance for each model. Dev set results are not available for models from Daudaravicius et al. (2016).

2015). After fine-tuning, all three models correctly identify sentences with contractions as "need edit". Since RoBERTa and BERT are not pre-trained on formal academic text (unlike SciBERT), there is no preexisting inclination to attend to contractions. Despite this, the models heavily attend to the contractions in their last layer, as seen in Figure 2.

To further show that the models learn that contractions are an error rather than tokens such as `'ll` and `'s`, we construct adversarial examples where such tokens are not contractions and inspect the models' predictions and attention maps. As seen in Figure 3, the models do not attend to `'s` and predict "no edit," demonstrating that they learn context-specific representations of novel patterns. This is consistent with the conclusion drawn in Kovaleva et al. (2019): the last two layers of BERT undergo the largest changes in fine-tuning.

4.2 Interpreting

Fine-tuning pre-trained models adjusts model parameters to task-specific patterns. Attention mechanisms (Bahdanau et al., 2016) provide insight into the patterns a model identifies. Clark et al. (2019) showed that BERT's attention is interpretable before fine-tuning. We are interested in whether attention is interpretable after fine-tuning.

Qualitatively, we see that all three models attend to relevant words, as shown in Figure 2. Furthermore, this is a result of fine-tuning, as shown in Figure 4. However, some evidence shows that attention is not relevant to a model's predictions (Jain and Wallace, 2019). To check if BERT's self-attention mechanism is interpretable, we manually annotate 50 examples, following the method described in Vashishth et al. (2019).

We select 50 sentences that BERT, RoBERTa and SciBERT all correctly identify as "need edit." 25 sentences contain a single spelling error, and 25 contain only text that is deleted (typically incorrect

	BERT	SciBERT	RoBERTa
Spelling	72	68	60
Deleted	80	76	68

Table 2: % of manually annotated sentences where the model's top 3 attended-to words include the proposed edit's most relevant word for the edit.

punctuation or redundant explanation). For each sentence and each model, we mark the three words most heavily attended to by the `[CLS]` in the last layer. We annotate whether the top three words predicted by each model are useful in predicting that the sentence needs editing. Table 2 shows our results.

We find that BERT's attention in the last layer more accurately identifies words relevant to predicting "need edit", despite lower F1 scores on the classification task. We hypothesize that SciBERT's lower interpretability in the final layer is due in part to its scientific writing pre-training. It could be encoding the representation of an incorrect sentence in earlier layers, then attending to `[SEP]` as a no-op in the later layers, as proposed in Clark et al. (2019).

Encouraged by these results, we perform a larger-scale, automated analysis of how well pre-trained transformer models attend to relevant words in the next section.

5 Quantifying Interpretability

To create a large set of edits for which interpretability can be automatically evaluated, we find either sentences with only removed text (*deleted words*) or sentences with a misspelled word that is replaced (*spelling error*).[4] These two criteria produce a set of edits where the most relevant words are those that have been deleted. These criteria also avoid

[4] We use `https://github.com/hermitdave/FrequencyWords/` for an English dictionary.

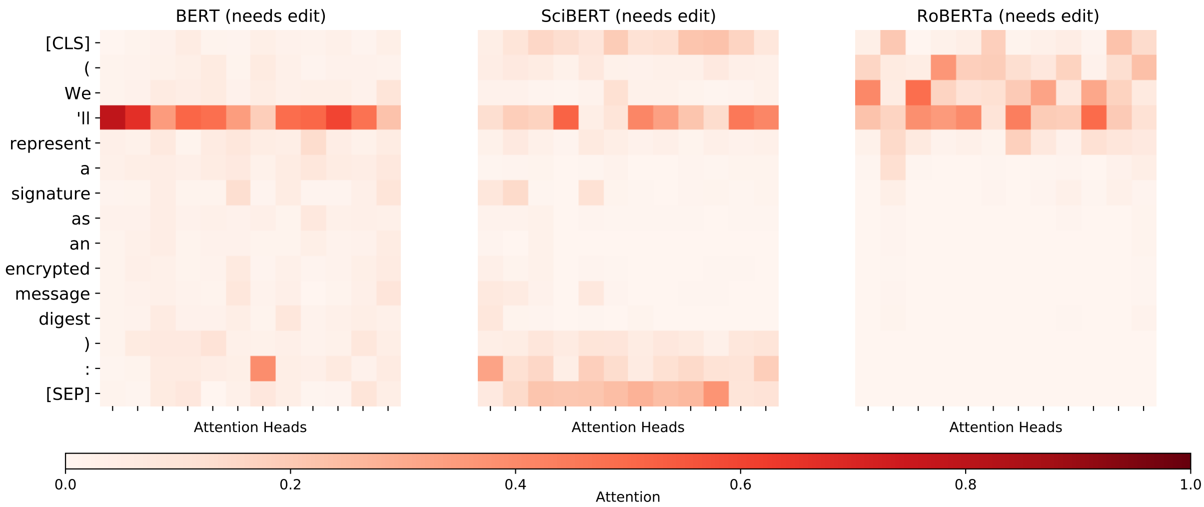

Figure 2: `[CLS]` attention maps for “(We~~’ll~~ **will** represent a signature as an encrypted message digest):” in the last layer. Notice that all three models attend heavily to `'ll`.

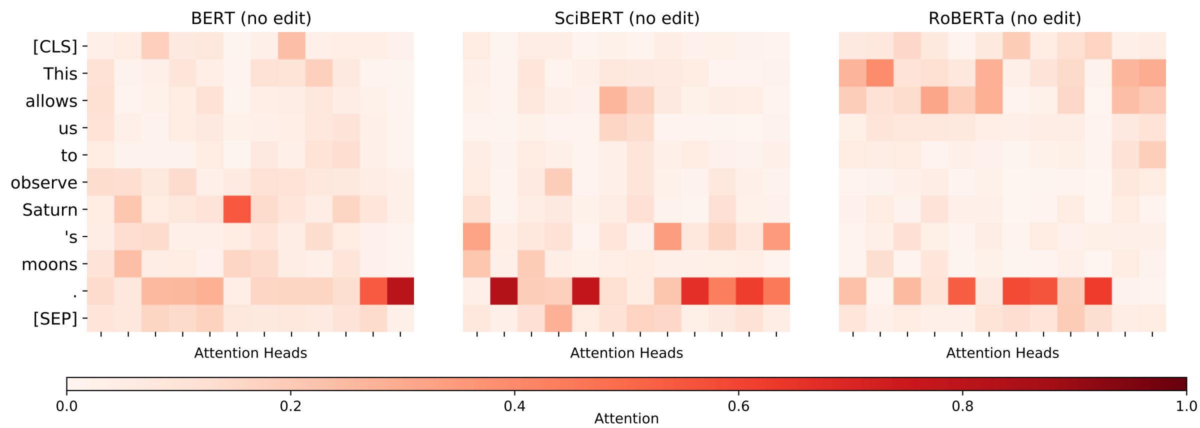

Figure 3: `[CLS]` attention maps for “This allows us to observe Saturn’s moons” in the last layer. Notice that all three models predict “no edit” and do not attend to `'s`.

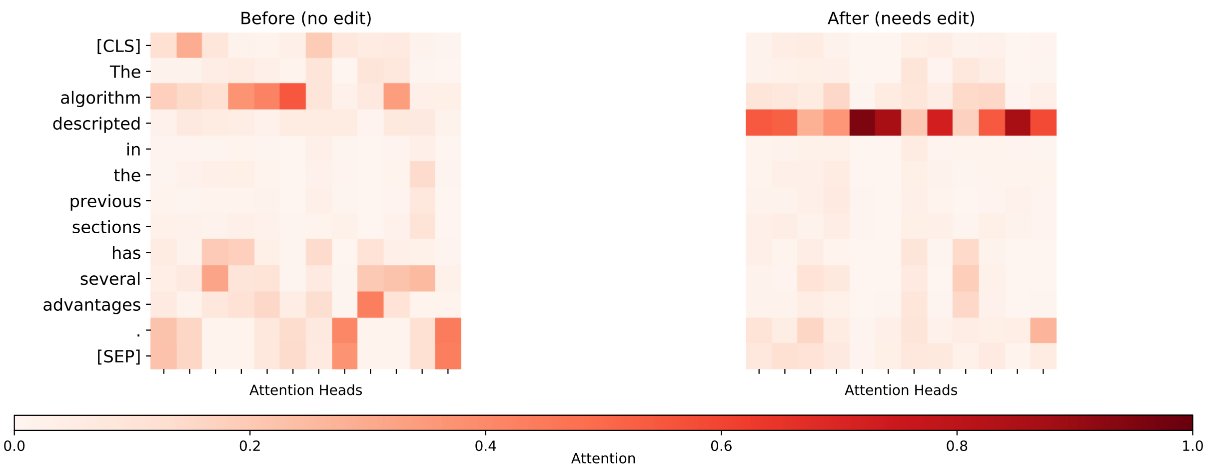

Figure 4: Comparison of BERT’s `[CLS]` attention map for “The algorithm ~~descripted~~**described** in the previous sections has several advantages” before and after fine-tuning on the AESW task.

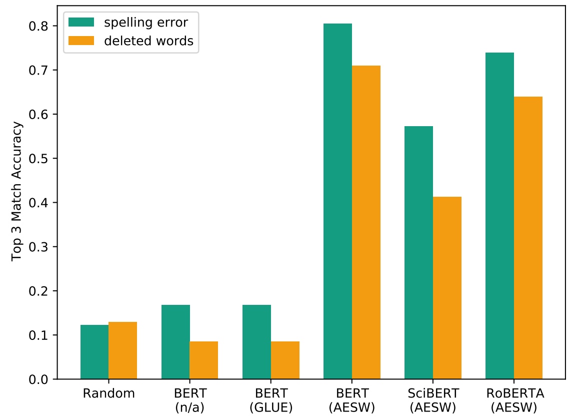

Figure 5: Comparison of BERT, SciBERT and RoBERTa's ability to predict the most important word in a sentence with three baselines: random, BERT with no fine-tuning and BERT fine-tuned on GLUE. "Top 3 Match Accuracy" (y-axis) is % of examples where the target words were a subset of the top 3 predicted words.

edits with insertions (because it is not clear which tokens are relevant in the original sentence) and edits where context is required to identify erroneous tokens. For example, many edits are simply removing a comma from a sentence. We would not expect the models to attend heavily to the comma, because the surrounding context is why the comma should be removed.

For each edit, let n represent the number of deleted words. To choose n important words from `[CLS]`'s attention, we use two strategies for each model:

1. Take n words corresponding to the n largest attention weights (*max*)
2. Take the mean attention across all 12 heads for each word, then choose n words corresponding to the n largest mean attention weights (*mean*)

For each example sentence, we calculate:

1. Whether a model correctly finds the exact n words deleted (*Exact*)
2. The Jaccard similarity of the n target words and the model's n predictions (*Avg Jac.*)
3. Whether the Jaccard similarity is > 0.5, the metric for interpretability presented in DeYoung et al. (2020) (*Jac. Acc.*)
4. Whether the n target words are a subset of the top 3 predicted words (*Top 3*)

Top 3 match accuracy for each model using the *mean* strategy can be seen in Figure 5. Appendix C contains a table with all of our results.

We find that BERT's results are consistent with our manual study. However, RoBERTa demonstrates more interpretability than SciBERT. Again, we hypothesize that SciBERT's pre-training scheme is more relevant to the task, so rather than updating the later layers' parameters during fine-tuning, SciBERT encodes the relevant information in earlier layers. The stark difference between the baselines and all three models demonstrate that BERT's attention in the last layer after fine-tuning is interpretable.

6 Conclusion

In this paper, we apply three BERT-based models to a sentence classification task, then quantify their interpretability through a small-scale manual study before expanding to a larger-scale automated study. We find that BERT's final attention layer is clearly interpretable by both human annotators and simple automated metrics.

Future work might expand the subset of examples that can be automatically annotated in order to further understand BERT's interpretability on different classes of edits. Additionally, more work is needed to understand the impacts of in-domain pre-training on model interpretability.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural machine translation by jointly learning to align and translate.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. 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), pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Steven Bird. 2002. Nltk: The natural language toolkit. In In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario

Amodei. 2020. Language models are few-shot learners.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In ICLR.

Vidas Daudaravicius, Rafael E. Banchs, Elena Volodina, and Courtney Napoles. 2016. A report on the automatic evaluation of scientific writing shared task. In Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications, pages 53–62, San Diego, CA. Association for Computational Linguistics.

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

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4443–4458, Online. Association for Computational Linguistics.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning.

Jiaqi Guo, Zecheng Zhan, Yan Gao, Yan Xiao, Jian-Guang Lou, Ting Liu, and Dongmei Zhang. 2019. Towards complex text-to-SQL in cross-domain database with intermediate representation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4524–4535, Florence, Italy. Association for Computational Linguistics.

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

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. 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), pages 4365–4374, Hong Kong, China. Association for Computational Linguistics.

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

Alex Osmond. 2015. Academic writing and grammar for students. Sage.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, and Manaal Faruqui. 2019. Attention interpretability across nlp tasks.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 37–42, Florence, Italy. Association for Computational Linguistics.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. In Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 63–76, Florence, Italy. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing. ArXiv, abs/1910.03771.

A Model and Training Details

We add four special tokens (_MATH_, _MATHDISP_, _CITE_ and _REF_) found in the AESW data (Table 3 contains an example of each) to the model vocabulary, fine-tuning the word representations during training. We train all models for a maximum of 30 epochs with a patience of 5 on a single Tesla P100 GPU.

All models (BERT[5], SciBERT[6], RoBERTA[7]) are based on their HuggingFace (Wolf et al., 2019) implementations.

We list all the key hyperparameters and tuning bounds for reproducibility in Table 4. Additionally, https://github.com/samuelstevens/bert-edits contains code and instructions for reproducing results.

Token	LaTeX Example
MATH	`$\beta_{2}$`
MATHDISP	`$$ 2 + 3 $$`
CITE	`\cite{google2018}`
REF	`\ref{tab:results}`

Table 3: Special tokens found in the original AESW data that should not be split further into bytes/tokens.

B Converting Attention

We convert token-level attention to word-level attention to allow for a direct comparison between BERT, SciBERT and RoBERTa, which all have different vocabularies (and different tokenization schemes in the case of RoBERTa).

We use NTLK (Bird, 2002) for word tokenization because we want to split contractions and punctuation into separate tokens.

As per Clark et al. (2019), for attention *to* a split-up word, we sum the attention weights over its token. For attention *from* a split-up word, we take the mean of the attention weights over its tokens.

C Interpretability Results

For completeness, all model-strategy combinations and their results are listed in Tables 5 and 6. Jac. Acc. refers to the matching criteria presented in DeYoung et al. (2020), where a prediction of words is considered a match if the Jaccard similarity of the prediction and target words is greater than 0.5.

[5] https://huggingface.co/transformers/v3.0.2/model_doc/bert.html#bertforsequenceclassification

[6] https://github.com/allenai/scibert#pytorch-huggingface-models

[7] https://huggingface.co/transformers/v3.0.2/model_doc/roberta.html#robertaforsequenceclassification

Model	Hyperparameters	Hyperparameter bounds
$\text{BERT}_{\text{base}}$	learning rate: 1×10^{-6} batch size: 32 model: bert-base-uncased vocab size: 30526 (normally 30522)	learning rate: (2×10^{-7}, 1×10^{-6}, 2×10^{-5}, 1×10^{-4})
SciBERT	learning rate: 1×10^{-6} batch size: 32 model: allenai/scibert_scivocab_uncased vocab size: 31094 (normally 31090)	learning rate: (1×10^{-6})
$\text{RoBERTa}_{\text{base}}$	learning rate: 1×10^{-6} batch size: 32 model: roberta-base vocab size: 50269 (normally 50265)	learning rate: (1×10^{-6})

Table 4: Hyperparameter options for each model. Note that each model had 4 special tokens added to the vocabulary. BERT was fine-tuned first. Because of compute limitations, RobERTa and SciBERT were both fine-tuned using the same hyperparameters as the optimal BERT configuration (learning rate of (1×10^{-6})).

Model Dataset	Strategy	Spelling Exact	 Jac. Acc.	 Avg Jac.	 Top 3	Deletes Exact	 Jac. Acc.	 Avg Jac.	 Top 3
Random	n/a	**0.038**	**0.038**	**0.040**	0.123	**0.043**	**0.045**	**0.053**	**0.130**
	n/a	**0.038**	**0.038**	**0.040**	0.123	**0.043**	**0.045**	**0.053**	**0.130**
BERT	mean	0.030	0.030	0.035	**0.168**	0.013	0.013	0.018	0.085
n/a	max	0.018	0.018	0.022	0.117	0.013	0.013	0.018	0.102
BERT	mean	0.030	0.030	0.035	**0.168**	0.013	0.013	0.018	0.085
GLUE	max	0.018	0.018	0.022	0.117	0.013	0.013	0.018	0.102

Table 5: Baseline interpretability.

Model Dataset	Strategy	Spelling Exact	 Jac. Acc.	 Avg Jac.	 Top 3	Deletes Exact	 Jac. Acc.	 Avg Jac.	 Top 3
BERT	mean	**0.667**	**0.668**	**0.683**	**0.805**	**0.562**	**0.568**	**0.585**	**0.709**
AESW	max	0.632	0.634	0.648	0.767	0.535	0.541	0.559	0.690
RoBERTA	mean	0.457	0.459	0.471	0.739	0.315	0.318	0.334	0.639
AESW	max	0.414	0.416	0.426	0.708	0.248	0.251	0.267	0.584
SciBERT	mean	0.228	0.229	0.238	0.573	0.124	0.127	0.141	0.413
AESW	max	0.180	0.182	0.194	0.528	0.110	0.112	0.127	0.422

Table 6: Comparing the three models' interpretability.