Faithful, Simple NLP Explanations with FRESH

Arthur Chen
Stanford University
Department of Statistics
aulchen@stanford.edu

Abstract

Large language models (LLM) have made tremendous advances in recent years, and with it, a desire to explain LLM output. One method of producing an explanation is with a *rationale*, a subset of the input that explains the output. Rationales can be introduced as additions to the input to improve accuracy [Zaidan et al., 2007], or as replacements to the input to improve explainability [Lei et al., 2016, Jain et al., 2020].

I implement Faithful Rationale Extraction from Saliency tHresholding (FRESH) [Jain et al., 2020], an explainability method that uses automatically-extracted rationales to train a text classification model. I implement FRESH and compare it against SHAP [Strumbelj and Kononenko, 2010, Lundberg and Lee, 2017]. I find that FRESH sacrifices accuracy for speed and explainability, making it suitable for when explainability is critical, or when compute resources are scarce. I also find that FRESH and SHAP have issues when explaining text that the model classifies incorrectly.

1 Background

With the rise of black box models, explainability in machine learning has become a major area of research. However, many competing explanation methods exist.

1.1 Existing Methods

I summarize three sample explanation methods that are useful background: attention scores [Bahdanau et al., 2014], SHAP values [Strumbelj and Kononenko, 2010, Lundberg and Lee, 2017] and rationale extraction [Lei et al., 2016].

1.1.1 Attention Scores

Attention scores are quite straightforward as a model explanation method: given a model that uses attention, use those values as an explanation. However, the interpretation of attention weights is controversial [Jain and Wallace, 2019, Wiegreffe and Pinter, 2019, Serrano and Smith, 2019]. Attention scores require no additional computations, as they are computed when making predictions with the model.

1.1.2 SHAP

SHAP is a model-agnostic method of explaining data by using Shapley values. Given a function $f(x)$ to be explained at point $x \in \mathbb{R}^d$, Shapley values additively divide the value of $f(x)$ into parts, each associated with feature x_i, such that all of these parts sum up to $f(x)$. It can be shown that under some reasonable axioms, Shapley values are the unique way of dividing up the value $f(x)$. In NLP classification tasks, x is text to be be classified, and $f(x)$ is often the logit of the predicted class.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

Algorithmically, SHAP consists of various methods to estimate the Shapley values. Conceptually, SHAP constructs a linear model g_x that takes zero-one vectors of length d as input, where each element indicates whether feature x_i is present in the explanation or not. g_x is constructed such that $g_x(\vec{1}) = f(x)$, and the Shapley values are the coefficients on the zero-one elements. As opposed to attention scores, Shapley values require additional computations to estimate g_x.

1.1.3 Rationale Extraction

One method of producing explanations in NLP is to generate rationales. Using the definition in Lei et al. [2016], Strout et al. [2019], and Jain et al. [2020], a *rationale* is a subset z of the original text input x. Conceptually, a good rationale is one such that $f(z) = f(x)$. That is, z consists of the 'informative' parts of x that the model uses to make its prediction. Lei et al. [2016] also make the rationales human interpretable by making them short and contiguous (i.e. extracting phrases and sentences, rather than isolated random words).

Lei et al. [2016]'s model for classifying text uses two modules, jointly trained. The **encoder** module f is a standard classification model, such as an RNN. The **generator** module gen generates a conditional probability distribution $p_{z|x}$ over rationales z. The sampled rationale $z(x)$ is then used as the input for the encoder f. During training, the encoder and generator are jointly trained. The loss function then ensures that the rationales z predict the output through f, and that the rationales are short and contiguous.

Although the rationale is conceptually simple, it's algorithmically difficult to compute. Training the model requires reinforcement learning, which is outside of scope for this course. As such, I make no attempt to replicate Lei et al. [2016].

1.2 Explanation Evaluation: Faithfulness and Plausibility

Many concepts for evaluating the quality of an explanation exist. Previous researchers [Rudin, 2019, Lindardatos et al., 2020, Jacovi and Goldberg, 2020] identify two conceptual ways to assess the quality of an explanation: *plausibility* and *faithfulness*. An explanation is plausible if it is understandable by humans, while an explanation is faithful if it accurately reflects the true reasoning process of the model.

Attention scores may or may not be faithful. Jain and Wallace [2019] find that for a given model architecture, input x, and prediction $f(x)$, multiple possible attention weightings α result in the same $f(x)$. However, Wiegreffe and Pinter [2019] argue that since attention weights are indeed learned during training, they are indeed what the model looks at to produce output, and are thus faithful.

SHAP is not faithful, which can be seen from theoretical and conceptual points of view. Theoretically, instead of explaining f itself, SHAP constructs a separate model g_x. Although g_x is linear and thus interpretable, g_x is not f: the explanation is not the model [Rudin, 2019]. Conceptually, SHAP is model and data agnostic, so it can't be exposed to the true inner workings of a model.

Rationale extraction is faithful, with caveats. The encoder module only sees the rationale; however, the generator sees the entire input. If we only look at the encoder, the rationale is faithful by construction, as the the encoder only has access to the input through the rationale. However, if we consider the rationale extractor as a whole, including the generator, the method is not faithful. Words in x that are not in the rationale z still affect the model output $f(z)$ by changing the conditional distribution that the rationales are sampled from.

2 Motivation

Faithful Rationale Extraction from Saliency tHresholding (FRESH) [Jain et al., 2020] is a proposed method for NLP classification tasks that aims to retain the faithfulness and conceptual simplicity of rationale extraction while reducing the computational complexity. To reduce computations, we skip training the generator function, as in rationale extraction. Instead, the authors propose to use a saliency map like attention to find important features, then use a heuristic to select a rationale z. The final prediction module f then has to predict the output using z.

Conceptually, instead of training a complicated function to choose z such that $f(z) \approx f(x)$, we use a computationally-cheap heuristic to pick z, and hope that accuracy doesn't decrease too much. I assess

Input	[CLS]	i	like	pie	.	[SEP]
Head 1	.1	.3	.3	.2	.1	0
Head 2	0	.1	0	.3	0	.6
Importances		**.20**	**.15**	**.25**	**.05**	

Figure 1: Feature importances using attentions from the [CLS] token at the last layer. Note that punctuation gets importances, while [CLS] and [SEP] do not.

the produced explanations on four dimensions: accuracy, computational simplicity, faithfulness, and plausibility.

3 Methods

FRESH consists of three modules: the **support** module `supp`, the rational **extractor** module `ext`, and the **classifier** module `pred`. `supp` is any arbitrary method that produces a feature importance saliency map on the input tokens x. `ext` is a method for binarizing those feature importances into two groups, 'keep' and 'ignore.' `pred` is the final trained prediction module. Conceptually, `supp` and `ext` create a rationale z, and `pred` creates the final model prediction $f(z)$.

3.1 The supp Feature Importance Module

`sup` is an arbitrary feature importance map, such as attention weights or SHAP. I follow the lead of Jain et al. [2020] and use attention scores generated by vanilla BERT [Devlin et al., 2019]. For my purposes, BERT is a pretrained transformer-based encoder. When BERT tokenizes input, it pads the input with a [CLS] token at the start and a [SEP] token at the end of the input. After the input is tokenized and transformed, each token is represented with a 768 dimensional embedding. The embedding of the initial [CLS] token represents the information of the entire sentence, and can be used for downstream classification tasks [Alammar, 2021].

For `sup`, I take the feature importances to be the attention scores from the [CLS] token to all of the other (non [CLS] or [SEP]) input tokens at the final attention block. Since BERT has twelve attention heads, I average attention over all of the heads.

For example, consider Figure 1 and the input sequence x =`I like pie.`. In this simplified illustration with two attention heads, the feature importances for the non-[CLS] and [SEP] tokens are just average attentions from [CLS] to the tokens.

3.2 The ext Extractor Module

For `ext`, I use two heuristics initially suggested in Jain et al. [2020]: Top-k tokens and Contiguous-k tokens. For rationale length k, Top-k tokens selects the top k tokens with the highest importance scores, while Contiguous-k takes the contiguous k-length sequence with the highest summed feature importances. In the example in Figure 1 with $k = 2$, the top-k extractor will pick the tokens `i pie`, while the contiguous-k extractor will pick the tokens `like pie`. The rationale is then padded with [CLS] and [SEP] tokens to prepare for input into another BERT extractor.

The BERT tokenizer splits words into its subparts and represents each subpart as a token. For example, `pies` would be separated into two tokens, `pie` and `-s`. The original FRESH paper ensures that these tokens are not separated in the `ext` module. For simplicity, I ignore this.

Jain et al. [2020] weren't very clear on whether location embeddings were preserved between the `sup` and `ext`. I assume that location embeddings are not preserved. Thus, even though `like pie` correspond to locations 2 and 3 in the the original input, in the extracted rationale, they correspond to locations 1 and 2.

3.3 The pred Predictor Module

Conceptually, `pred` is trained end-to-end given z. Jain et al. [2020] fine-tuned a BERT model with a classification layer on top of the [CLS] token. As I have no compute resources besides a laptop, I do the same, but freeze the weights of vanilla BERT. This is equivalent to feeding the rationale into vanilla BERT, taking the 768 dimensional embedding of the [CLS] token, and feeding it into a logistic regression.

3.4 Vanilla BERT

To quantify the accuracy loss from looking the rationale z instead of the full input x, I let BERT look at the entire input sequence to extract features before feeding the extracted features into a logistic regression. I dub this 'vanilla BERT.' The accuracy of 'vanilla BERT' is an upper-bound on model performance with this architecture, and this is the accuracy that would be achieved by non-rationale methods such as attention weights or SHAP. Since I don't have the compute resources or knowledge to reimplement rationale extraction [Lei et al., 2016], I will not do so here.

3.5 Pretrained BERT

For SHAP predictions, I used the SHAP package in Python to explain predictions from a pretrained BERT model on this task [Morris et al., 2020]. Ideally I would have explained the predictions from vanilla BERT, but I could not get the SHAP package to produce explanations for a custom function.

4 Task and Data

The data consists of the Large Movie Review Dataset [Maas et al., 2011], as converted into CSV on Kaggle [N, 2018]. The data consists of a movie review in text, and a positive/negative sentiment. Positive sentiment corresponds to a review score of ≥ 7 out of 10, while negative corresponds to a score of ≤ 4 out of 10. The task is to classify the movie reviews as positive or negative.

Due to the time it takes to extract features, I take the first 2000 entries, as ordered by the Kaggle CSV, to be my train dataset, and entries 25000 through 26000 as my test dataset, leading to a 66-33 train/test split. The train dataset contains 1005 positive reviews, for a roughly 50-50 positive/negative split. The test dataset contains 541 positive reviews, for a roughly 55-45 positive/negative split. The train data ranges in token length from 24 to 1797 tokens, while the test data ranges in token length from 35 to 1323 tokens. The average token length of train and test sets were 286 and 299, respectively.

One limitation of BERT is that it can not process text longer than 512 tokens. I simply truncate input that are longer than 512 tokens. There were 240 train reviews and 133 test reviews that were truncated, meaning that roughly one-eighth of the reviews were truncated. After truncation, the train and test reviews have average token length of 259 and 268, respectively.

5 Metrics

I assess FRESH on four metrics: accuracy, computational simplicity, faithfulness, and plausibility. I expect accuracy to decrease by showing the model f the rationale z instead of the full input x, so I make some attempt to quantify this. Faithfulness is assessed conceptually, while computational simplicity is assessed informally. For plausibility, a full study of plausibility would require a research study and is outside of scope. Instead, I pick a few example observations and look at the plausibility of each, as measured by my intuition of whether the explanation makes sense or not.

6 Results

6.1 Accuracy

Results are listed in Table 1. We see that using the Top-150 and Contiguous-150 heuristics lead to a small drop in accuracy, while using the Top-50 and Contiguous-50 heuristics lead to a larger decrease in accuracy.

Table 1: Model prediction accuracies

Model	k=150	k=50
Vanilla BERT	.780	
Top k	.748	.660
Contiguous k	.759	.677

Somewhat surprisingly, we see that for a given value of k, the Contiguous-k heuristic outperforms the Top-k heuristic. This differs from Jain et al. [2020], where no method dominated. We can gain intuition on this result by looking at the extracted rationales listed in Figure 2. The Top-50 rationale seems to be picking up punctuation marks and isolated words. While in this case Top-50 extracts a continuous phrase ('sad to see such good actors buried by this disaster'), other examples, such as the one in Figure 3 show that Top-50 can have trouble selecting meaningful phrases. By contrast, by construction the Contiguous-50 heuristic always picks up contiguous phrases and sentences.

These $k = 150$ heuristics correspond to roughly one-half of the post-truncation average sequence length, while the $k = 50$ heuristics correspond to approximately one-sixth. In Jain et al. [2020], the authors used set k to approximately one-fifth of the input sequence. The authors got similar accuracy drops to my results on some datasets, but much smaller accuracy drops on other datasets. I have no idea whether my results come from the particular dataset I chose, or my lack of model fine-tuning.

6.2 Computational Simplicity

It took about an hour to generate 2,000 rationales using FRESH, for a rate of approximately 3 rationales a minute. By contrast, it took about ten minutes to generate one explanation using SHAP. FRESH is clearly much faster than SHAP. By contrast, attention scores are essentially instantaneous.

6.3 Faithfulness

As discussed above, SHAP is not faithful. FRESH is faithful in the same way that rationale extraction is faithful. Looking at the predictor, FRESH is faithful, but if we consider the model as a whole, FRESH is not faithful. Even if a word is outside of the rationale, changing a word in x might change the attention scores in such a way that we produce a different z. Thus in general, the output of $f(z)$ depends on all of x.

However, FRESH still has a stronger notion of faithfulness than attention scores. Serrano and Smith [2019] find that in a text classification setting, zeroing out the attention scores of random words can change a prediction model's output. Thus, the attentions on features marked 'unimportant' can still affect the output. By contrast, FRESH is protected against this form of attention score manipulation: reducing the attentions of words outside of the rationale does not alter the rationale. Thus although FRESH is not completely faithful, it is more faithful than attention scores.

6.4 Plausibility

I look at three examples of explanations: one where all models predicted the sentiment correctly, one where only the pretrained BERT did, and one where only FRESH did. For SHAP, blue indicates that a word increases the predicted class, while red indicates that the word decreases it. For FRESH, blue indicates that the word is in the rationale. For all examples, I use $k = 50$ for FRESH and pretrained BERT for SHAP.

6.4.1 All Correct

For the review in Figure 2, all methods correctly predicted the sentiment, negative. SHAP is nonspecific - SHAP explains that every single word contributes to a negative sentiment. This is somewhat plausible, since the entire review is quite negative, but I would prefer if the explanation were more concise. Top 50 has a problem where it picks up random words and punctuation marks, but it does include a phrase ('sad to see such good actors buried by this disaster.') that plausibly explains negative sentiment. Contiguous 50 correctly picks up sentences that have negative sentiment.

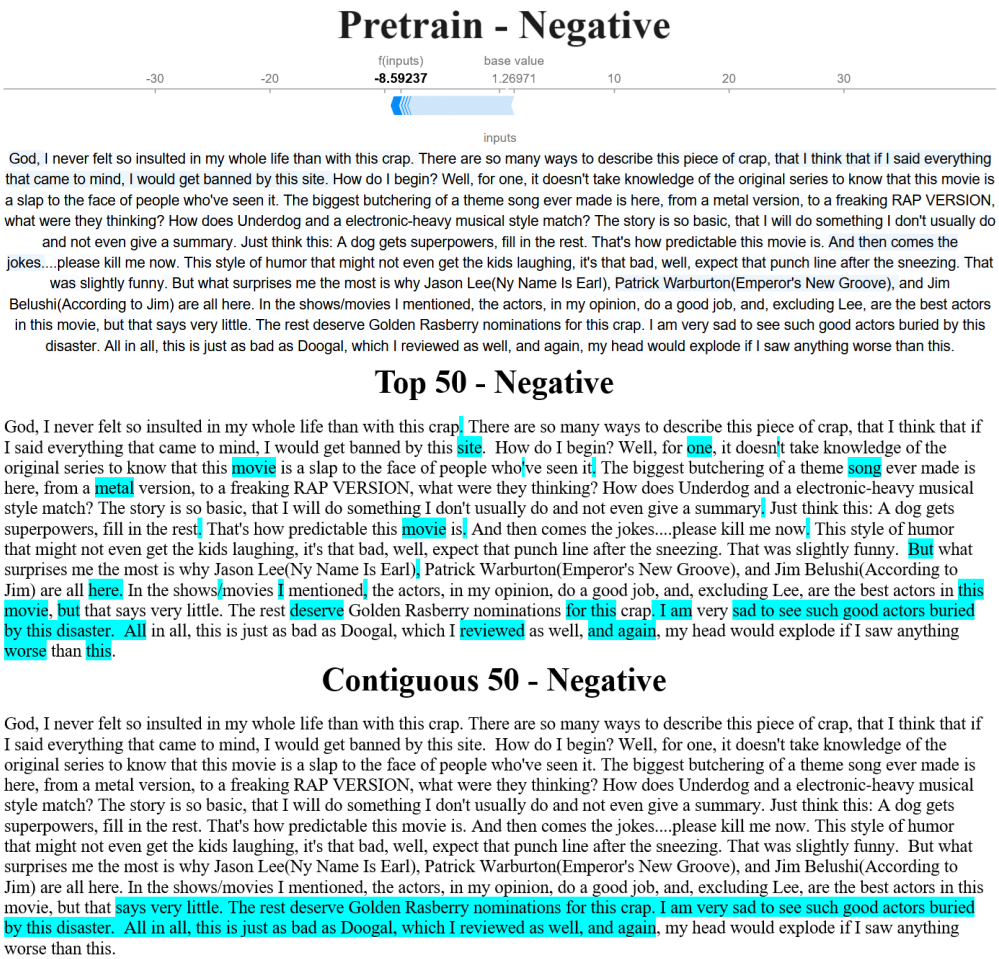

Figure 2: Explanations produced by FRESH and SHAP. All models correctly predicted negative sentiment.

We thus see an advantage that FRESH has over SHAP: the enforcing of conciseness in the explanations. Whereas SHAP can use the entire input as an explanation, FRESH is forced to use only a subset of the input, which can help interpretability.

6.4.2 Pretrain Correct

For the review in Figure 3, only the pretrained model correctly predicted the sentiment, negative, while FRESH predicted positive sentiment. SHAP's explanation for the negative sentiment for the pretrained model is generally accurate - the word 'bad' is the largest factor encouraging a negative review. Similarly, the phrase 'if you haven't seen get shorty, you will probably like be cool' is positive sentiment, while the phrase 'if you liked get shorty, you will leave the theater wanting to kill the director and screenwriter' is mostly negative sentiment. However, there are still strange parts of the explanation. According to SHAP, the phrase '10 minutes of the movies' is indicative of a negative sentiment, while 'screenwriter' indicates positive sentiment.

The FRESH rationales, by contrast, do little to explain model behavior. For the Top 50 heuristic, the only phrase extracted is 'wait for the DVD,' and the predicted sentiment of the rationale is positive. I have no idea how 'wait for the DVD' is a positive-sentiment phrase. The Contiguous 50 heuristic is a bit better, as it manages to extract the last three sentences, which seem to properly summarize the review. However, I don't know why FRESH thought that the last three sentences result in positive sentiment. In conclusion, FRESH does not do a good job of explaining an incorrect prediction.

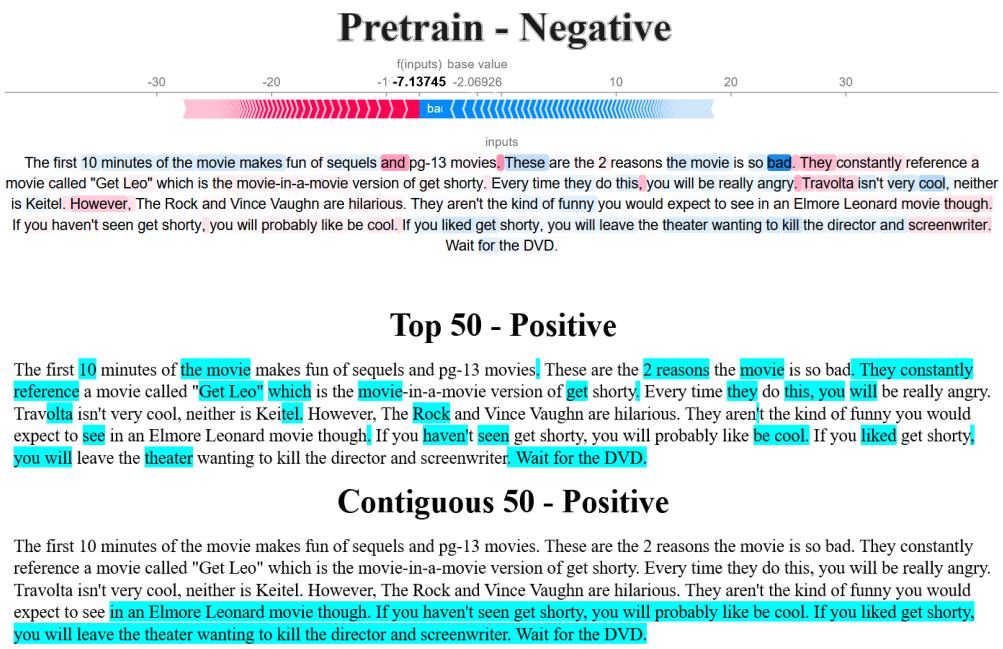

Figure 3: Pretrain correctly predicted negative sentiment, while FRESH incorrectly predicted positive sentiment.

6.4.3 FRESH Correct

Nor does SHAP do much better at explaining incorrect predictions. For the review in Figure 4, the FRESH models correctly predicted positive sentiment, while the pretrained predicted negative sentiment. The FRESH rationales are plausible. Top 50 picks up the phrases 'I love them,' 'I consider this the gem of my collection,' and 'If bad movies are your, definitely get this one,' which are positive phrases. Similarly, Contiguous 50 picks up those phrases, along with 'this is loads of fun.' In this case, the rationales produced by FRESH are plausible explanations for a positive prediction.

In contrast, the SHAP explanation is plausible, but for a completely opposite class. SHAP picks up the opening two sentences as driving the prediction. This would be a plausible explanation, if the predicted sentiment were positive. Perhaps 'so bad it's good' is confusing the pretrained model, but I have no idea now 'simply beautiful' can be construed to have negative sentiment. In short, the SHAP explanation is implausible, since it doesn't explain a negative sentiment prediction.

7 Discussion

The big advantages of FRESH are its faithfulness, computational simplicity, and conceptual simplicity. For faithfulness, FRESH is more faithful than attention scores or SHAP, while retaining the faithfulness of rationale extraction. For computational simplicity, FRESH is more computationally complex than attention scores, but much faster than SHAP, and is implementable with my knowledge, unlike rationale extraction. Finally, the produced rationales were roughly as plausible as SHAP's.

All these properties come at a cost: reduced accuracy. When looking at roughly 20% of the average input, we lose about one-eighth of our accuracy as compared to attention scores and SHAP, two methods that can see the full input when making predictions. In their paper, Jain et al. [2020] generally found FRESH to have comparable or superior accuracy to rationale extraction, but replicating their result is out of scope.

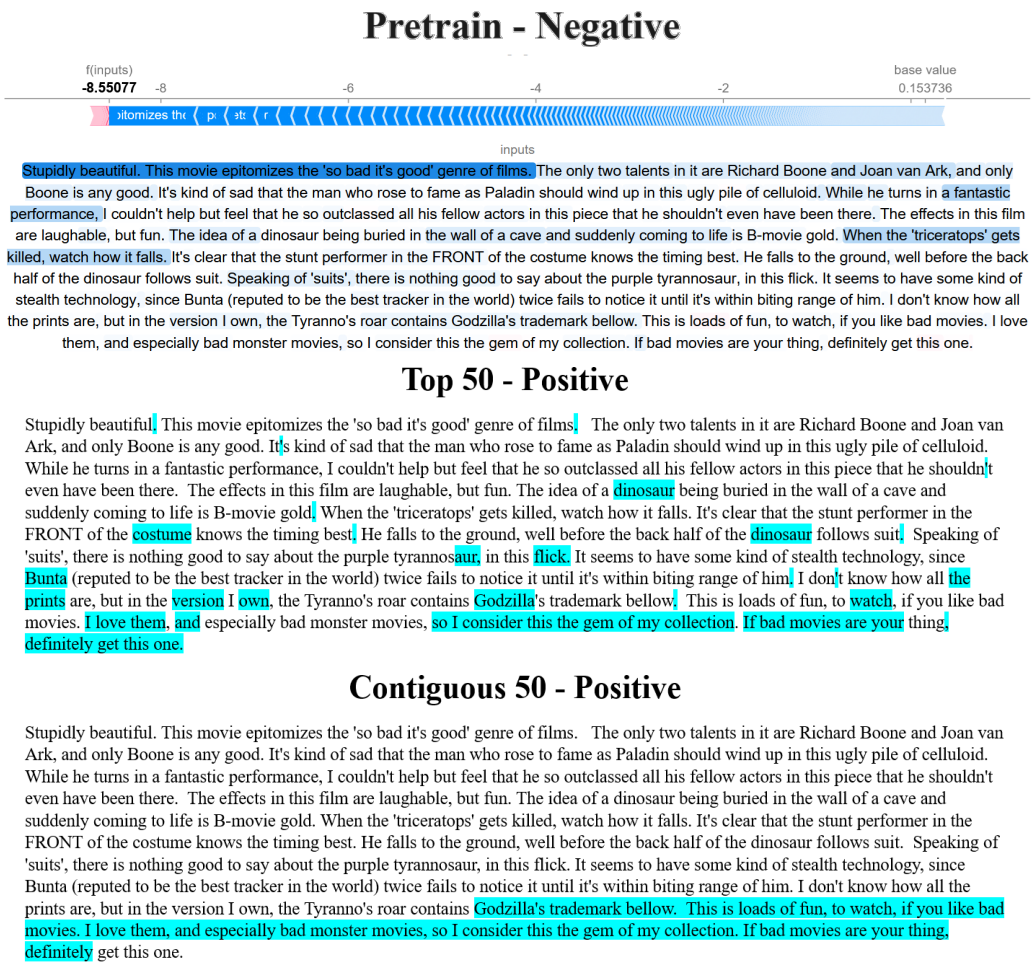

Figure 4: Pretrain correctly predicted negative sentiment, while FRESH incorrectly predicted positive sentiment.

8 Conclusion and Future Work

FRESH is a proposed method for generating explanations in NLP tasks that combines many of the good features of previous methods, at the cost of reduced prediction accuracy. FRESH is most practically useful in NLP tasks that require faithfulness and are willing to sacrifice accuracy to achieve it. The most obvious example would be an automated job-hiring software that reads resumes and cover letters, as we have to ensure that the model uses no protected characteristics to make its decision. As part of the rationale generating module, we can simply tell the model to give words related to protected characteristics an importance of zero, ensuring that the predictor module never sees them.

Three obvious extensions to the current work present themselves. First, given additional time, I would compare FRESH's explanations with those generated by attention scores, and see if the increase in faithfulness is worth the extra setup. Computationally, given compute resources, I would fine-tune the BERT model used for classification, rather than freezing the parameters, as in practice a model would be fine-tuned before it's deployed. I would also use these resources to generate more SHAP explanations to get a better sense for SHAP's plausibility. On a token level, given that the Top k extractor seems to pick up isolated punctuation marks, I would see if I can ignore attention scores on punctuation marks. With slightly more time, I clean up the code, and also figure out how to combine parts of a word into a single token.

References

Jay Alammar. The illustrated bert, elmo, and co. (how nlp cracked transfer learning). 2021. URL http://jalammar.github.io/illustrated-bert/.

Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. 2014. doi: https://doi.org/10.48550/arXiv.1409.0473.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2019. doi: https://doi.org/10.48550/arXiv.1810.04805. URL https://arxiv.org/abs/1810.04805.

Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020. doi: 10.18653/v1/2020.acl-main.386.

Sarthak Jain and Byron Wallace. Attention is not explanation. 2019. doi: https://doi.org/10.48550/arXiv.1902.10186.

Sarthak Jain, Sarah Wiegreffe, Yuval Pinter, and Byron Wallace. Learning to faithfully rationalize by construction. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020. doi: 10.18653/v1/2020.acl-main.409. URL https://aclanthology.org/2020.acl-main.409/.

Tao Lei, Regina Barzilay, and Tommi Jaakola. Rationalizing neural predictions. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016. doi: 10.18653/v1/D16-1011.

Pantelis Lindardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 2020. doi: https://doi.org/10.3390/e23010018. URL https://www.mdpi.com/1099-4300/23/1/18.

Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017. doi: https://dl.acm.org/doi/10.5555/3295222.3295230. URL https://arxiv.org/abs/1705.07874.

Andrew Maas, Raymond Daly, Peter Pham, Dan Huang, Andrew Ng, and Christopher Potts. Learning word vectors for sentiment analysis. *The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*, 2011. URL https://ai.stanford.edu/~amaas/data/sentiment/.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. 2020. doi: https://doi.org/10.48550/arXiv.2005.05909. URL https://huggingface.co/textattack/bert-base-uncased-imdb.

Lakshmipathi N. Imdb dataset of 50k movie reviews. 2018. URL https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 2019. doi: https://doi.org/10.1038/s42256-019-0048-x. URL https://www.nature.com/articles/s42256-019-0048-x#citeas.

Sofia Serrano and Noah Smith. Is attention interpretable? *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019. doi: 10.18653/v1/P19-1282.

Julia Strout, Ye Zhang, and Yamond J. Mooney. Do human rationales improve machine explanations? *Proceedings of the Second BlackboxNLP Workshop at ACL*, 2019. doi: https://doi.org/10.48550/arXiv.1905.13714.

Erik Strumbelj and Igor Kononenko. An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 2010. doi: https://dl.acm.org/doi/10.5555/1756006.1756007.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019. doi: https://doi.org/10.48550/arXiv.1908.04626.

Omar Zaidan, Jason Eisner, and Christine Piatko. Using "annotator rationales" to improve machine learning for text categorization. *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, 2007. URL `https://aclanthology.org/N07-1033.pdf`.