Improving the Faithfulness of Attention-based Explanations with Task-specific Information for Text Classification

George Chrysostomou **Nikolaos Aletras**
Department of Computer Science, University of Sheffield
United Kingdom
{gchrysostomou1, n.aletras}@sheffield.ac.uk

Abstract

Neural network architectures in natural language processing often use attention mechanisms to produce probability distributions over input token representations. Attention has empirically been demonstrated to improve performance in various tasks, while its weights have been extensively used as explanations for model predictions. Recent studies (Jain and Wallace, 2019; Serrano and Smith, 2019; Wiegreffe and Pinter, 2019) have showed that it cannot generally be considered as a faithful explanation (Jacovi and Goldberg, 2020) across encoders and tasks. In this paper, we seek to improve the faithfulness of attention-based explanations for text classification. We achieve this by proposing a new family of Task-Scaling (TaSc) mechanisms that learn task-specific non-contextualised information to scale the original attention weights. Evaluation tests for explanation faithfulness, show that the three proposed variants of TaSc improve attention-based explanations across two attention mechanisms, five encoders and five text classification datasets without sacrificing predictive performance. Finally, we demonstrate that TaSc consistently provides more faithful attention-based explanations compared to three widely-used interpretability techniques.[1]

1 Introduction

Natural Language Processing (NLP) approaches for text classification are often underpinned by large neural network models (Cho et al., 2014; Devlin et al., 2019). Despite the high accuracy and efficiency of these models in dealing with large amounts of data, an important problem is their increased complexity that makes them opaque and hard to interpret by humans which usually treat them as *black boxes* (Zhang et al., 2018; Linzen et al., 2019).

Attention mechanisms (Bahdanau et al., 2015) produce a probability distribution over the input to compute a vector representation of the entire token sequence as the weighted sum of its constituent vectors. A common practice is to provide explanations for a given prediction and qualitative model analysis by assigning importance to input tokens using scores provided by attention mechanisms (Chen et al., 2017; Wang et al., 2016; Jain et al., 2020; Sun and Lu, 2020) as a mean towards model interpretability (Lipton, 2016; Miller, 2019).

A faithful explanation is one that accurately represents the true reasoning behind a model's prediction (Jacovi and Goldberg, 2020). A series of recent studies illustrate that explanations obtained by attention weights do not always provide faithful explanations (Serrano and Smith, 2019) while different text encoders can affect attention interpretability, e.g. results can differ when using a recurrent or non-recurrent encoder (Wiegreffe and Pinter, 2019).

A limitation of attention as an indicator of input importance is that it refers to the word in context due to information mixing in the model (Tutek and Snajder, 2020). Motivated by this, we aim to improve the effectiveness of neural models in providing more *faithful* attention-based explanations for text classification, by introducing non-contextualised information in the model. Our contributions are as follows:

- We introduce three Task-Scaling (TaSc) mechanisms (§4), a family of encoder-independent components that learn task-specific non-contextualised importance scores for each word in the vocabulary to scale the original attention weights which can be easily ported to any neural architecture;

[1]Code is available at: https://github.com/GChrysostomou/tasc.git

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 477–488
August 1–6, 2021.

- We show that TaSc variants offer more robust, consistent and faithful attention-based explanations compared to using vanilla attention in a set of standard interpretability benchmarks, without sacrificing predictive performance (§6);
- We demonstrate that attention-based explanations with TaSc consistently outperform explanations obtained from two gradient-based and a word-erasure explanation approaches (§7).

2 Related Work

2.1 Model Interpretability

Explanations for neural networks can be obtained by identifying which parts of the input are important for a given prediction. One way is to use sparse linear meta-models that are easier to interpret (Ribeiro et al., 2016; Lundberg and Lee, 2017; Nguyen, 2018). Another way is to calculate the difference in a model's prediction between keeping and omitting an input token (Robnik-Šikonja and Kononenko, 2008; Li et al., 2016b; Nguyen, 2018). Input importance is also measured using the gradients computed with respect to the input (Kindermans et al., 2016; Li et al., 2016a; Arras et al., 2016; Sundararajan et al., 2017). Chen and Ji (2020) propose learning a variational word mask to improve model interpretability. Finally, extracting a short snippet from the original input text (rationale) and using it to make a prediction has been recently proposed (Lei et al., 2016; Bastings et al., 2019; Treviso and Martins, 2020; Jain et al., 2020; Chalkidis et al., 2021).

Nguyen (2018) and Atanasova et al. (2020) compare explanations produced by different approaches, showing that in most cases gradient-based approaches outperform sparse linear meta-models.

2.2 Attention as Explanation

Attention weights have been extensively used to interpret model predictions in NLP; i.e. (Cho et al., 2014; Xu et al., 2015; Barbieri et al., 2018; Ghaeini et al., 2018). However, the hypothesis that attention should be used as explanation had not been explicitly studied until recently.

Jain and Wallace (2019) first explored the effectiveness of attention explanations. They show that adversary attention distributions can yield equivalent predictions with the original attention distribution, suggesting that attention weights do not offer robust explanations. In contrast to Jain and Wallace (2019), Wiegreffe and Pinter (2019) and Vashishth et al. (2019) demonstrate that attention weights can in certain cases provide robust explanations. Pruthi et al. (2020) also investigate the ability of attention weights to provide plausible explanations. They test this through manipulating the attention mechanism by penalising words a priori known to be relevant to the task, showing that the predictive performance remain relatively unaffected. Sen et al. (2020) assess the plausibility of attention weights by correlating them with manually annotated explanation heat-maps, where plausibility refers to how convincing an explanation is to humans (Jacovi and Goldberg, 2020). However, Jacovi and Goldberg (2020) and Grimsley et al. (2020) suggest caution with interpreting the results of these experiments as they do not test the faithfulness of explanations (e.g. an explanation can be non-plausible but faithful or vice-versa).

Serrano and Smith (2019) test the faithfulness of attention-based explanations by removing tokens to observe how fast a *decision flip* happens. Results show that gradient attention-based rankings (i.e. combining an attention weight with its gradient) better predict word importance for model predictions, compared to just using the attention weights. Tutek and Snajder (2020) propose a method to improve the faithfulness of attention explanations when using recurrent encoders by introducing a word-level objective to sequence classification tasks. Focusing also on recurrent-encoders, Mohankumar et al. (2020) introduce a modification to recurrent encoders to reduce repetitive information across different words in the input to improve faithfulness of explanations.

To the best of our knowledge, no previous work has attempted to improve the faithfulness of attention-based explanations across different encoders for text classification by inducing task-specific information to the attention weights.

3 Neural Text Classification Models

In a typical neural model with attention for text classification; one-hot-encoded tokens $x_i \in \mathrm{R}^{|\mathrm{V}|}$ are first mapped to embeddings $\mathbf{e}_i \in \mathrm{R}^{\mathrm{d}}$, where $i \in [1, ..., t]$ denotes the position in the sequence, t the sequence length, $|V|$ the vocabulary size and d the dimensionality of the embeddings. The embeddings $\mathbf{e}_i$ are then passed to an encoder to produce hidden representations $\mathbf{h}_i = Enc(\mathbf{e}_i)$, where

$\mathbf{h}_i \in \mathrm{R}^{\mathrm{N}}$, with N the size of the hidden representation. A vector representation $\mathbf{c}$ for the entire text sequence $x_1, ..., x_t$ is subsequently obtained as the sum of $\mathbf{h}_i$ weighted by attention scores α_i:

$$\mathbf{c} = \sum_i \mathbf{c}_i, \quad \mathbf{c}_i = \mathbf{h}_i \alpha_i, \quad \mathbf{c} \in \mathrm{R}^{\mathrm{N}} \tag{1}$$

Vector $\mathbf{c}$ is finally passed to the output, a fully-connected linear layer followed by a softmax activation function.

3.1 Encoders

To obtain representations $\mathbf{h}_i$, we consider the following recurrent, non-recurrent and Transformer (Vaswani et al., 2017) encoders, $Enc(.)$, as in (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019): (i) bidirectional Long Short-Term Memory (**LSTM**; Hochreiter and Schmidhuber (1997)); (ii) bidirectional Gated Recurrent Unit (**GRU**; Cho et al. (2014)); (iii) Convolutional Neural Network (**CNN**; LeCun et al. (1999)); (iv) Multi-Layer Perceptron (**MLP**); (v) **BERT**[2] (Devlin et al., 2019).

3.2 Attention Mechanisms

Attention scores (a_i) are computed by passing the representations ($\mathbf{h}_i$) obtained from the encoder to the attention mechanism which usually consists of a similarity function ϕ followed by softmax:

$$a_i = \frac{\exp(\phi(\mathbf{h_i}, \mathbf{q}))}{\sum_{k=1}^{t} \exp(\phi(\mathbf{q}, \mathbf{h_k}))} \tag{2}$$

where $\mathbf{q} \in R^N$ is a trainable self-attention vector similar to Yang et al. (2016).

Following Jain and Wallace (2019), we consider two self-attention similarity functions: (i) **Additive Attention (Tanh;** Bahdanau et al. (2015)**):**

$$\phi(h_i, \mathbf{q}) = \mathbf{q}^T \tanh(W\mathbf{h}_i) \tag{3}$$

where W is a trainable model parameter; and (ii) **Scaled Dot-Product (Dot;** Vaswani et al. (2017)**):**

$$\phi(h_i, \mathbf{q}) = \frac{\mathbf{h}_i^T \mathbf{q}}{\sqrt{N}} \tag{4}$$

4 Task-Scaling (TaSc) Mechanisms

Attention indicates how well inputs around a position i correspond to the output (Bahdanau et al., 2015). For example, in a bidirectional recurrent encoder each token representation $\mathbf{h}_i$ contains information from the whole sequence so the attention weights actually refer to the input word *in context* and not individually (Tutek and Snajder, 2020).

Inspired by the simple and highly interpretable bag-of-words models, which assign a single weight for each word type (word in a vocabulary), we hypothesise that by scaling each input word's contextualised representation $\mathbf{c}_i$ (see Eq. 1) by its attention score *and* and a non-contextualised word type scalar score, we can improve attention-based explanations. The intuition is that by having a less contextualised sequence representation $\mathbf{c}$ we can reduce information mixing for attention.

For that purpose, we introduce the non-contextualised word type score s_{x_i} in Eq. 1 to enrich the text representation $\mathbf{c}$, such that:

$$\mathbf{c} = \sum_i \mathbf{h}_i \alpha_i s_{x_i}, \quad \mathbf{c} \in \mathrm{R}^{\mathrm{N}} \tag{5}$$

We compute s_{x_i} by proposing three Task-Scaling (**TaSc**) mechanisms.[3]

4.1 Linear TaSc (Lin-TaSc)

We first introduce Linear TaSc (Lin-TaSc), the simplest method in the family of TaSc mechanisms that estimates a scalar weight for each word in the vocabulary by introducing a new vector $\mathbf{u} \in \mathrm{R}^{|\mathrm{V}|}$. Given the input sequence $\mathbf{x} = [x_1, \ldots, x_t]$ representing one-hot-encodings of the tokens, we perform a look up on $\mathbf{u}$ to obtain the scalar weights of words in the sequence. $\mathbf{u}$ is randomly initialised and updated partially at each training iteration, because naturally each input sequence contains only a small subset of the vocabulary words.

We then obtain a task-scaled embedding $\hat{\mathbf{e}}_i$ for a token i in the input by multiplying the original token embedding with its word type weight u_i:

$$\hat{\mathbf{e}}_i = u_i \mathbf{e}_i \tag{6}$$

The intuition is that the embedding vector $\mathbf{e}_i$ was trained on general corpora and is a non-contextualised "generic" representation of input x_i. As such the score u_i will scale $\mathbf{e}_i$ to the task. We subsequently compute context-independent scores s_{x_i} for each token in the sequence, by summing all elements of its corresponding task-scaled embedding $\hat{\mathbf{e}}_i$; $s_{x_i} = \sum^d \hat{\mathbf{e}}_i$ in a similar way that token embeddings are averaged in the top-layers of a

[2] We use BERT to obtain $\mathbf{h}_i$ with an attention mechanism on top for consistency with the other encoders

[3] Number of parameters for each proposed mechanism in Appendix B.

neural architecture. We opted to sum-up and not average, because we want to retain large and small values from the task-scaled embedding vector $\hat{\mathbf{e}}_i$ (Atanasova et al., 2020).[4]

As the attention scores pertain to the word in context (Tutek and Snajder, 2020), we also expect the score s_{x_i} to pertain to the word without the contextualised information. That way, we complement attention which results into a richer sequence representation $\mathbf{c}$.

4.2 Feature-wise TaSc (Feat-TaSc)

Lin-TaSc assigns equal weighting to all the dimensions of the word embedding $\mathbf{e}_i$ (see Eq. 6), but some of them might be more important than others. Inspired by the RETAIN mechanism (Choi et al., 2016), Feature-wise TaSc (Feat-TaSc) learns different weights for each embedding dimension to identify the most important of them. Compared to Lin-TaSc where $\mathbf{e}_i$ is scaled uniformly across all vector dimensions, with Feat-TaSc each dimension is scaled independently. To achieve this, we introduce a learnable matrix $\mathbf{U} \in \mathrm{R}^{|\mathrm{V}| \times \mathrm{d}}$. Similar to Lin-TaSc, given the input sequence $\mathbf{x}$, we perform a look up on $\mathbf{U}$ to obtain $\mathbf{U}_s = [\mathbf{u}_1, \ldots, \mathbf{u}_t]$. $\mathbf{U}$ is randomly initialised and updated partially at each training iteration. To obtain s_{x_i}, we perform a dot product between $\mathbf{u}_i$ and embedding vector $\mathbf{e}_i$; $s_{x_i} = \mathbf{u}_i \cdot \mathbf{e}_i$.

4.3 Convolutional TaSc (Conv-TaSc)

Lin-TaSc and Feat-TaSc weigh the original word embedding $\mathbf{e}_i$ but do not consider any interactions between embedding dimensions. Conv-TaSc addresses this limitation by extending Lin-TaSc.[5] We apply a CNN[6] with n channels over the scaled embedding $\hat{\mathbf{e}}_i$ from Lin-TaSc, keeping a single stride and a 1-dimensional kernel. This way, we ensure that input words remain context-independent. We then sum over the filtered scaled embedding $\hat{\mathbf{e}}_i^f$, to obtain the scores s_{x_i}; $s_{x_i} = \sum^d \hat{\mathbf{e}}_i^f$.[4]

[4] We also tried max and mean-pooling or using the u_i directly instead of s_i in early experimentation resulting in lower results.

[5] We only apply Conv-TaSc over Lin-TaSc to keep the mechanism relatively lightweight. Note that Feat-TaSc learns an extra matrix of equal size to the embedding matrix.

[6] See CNN configurations in Appendix A.

5 Evaluating Attention-based Interpretability

Jacovi and Goldberg (2020) propose that an appropriate measure of *faithfulness* of an explanation can be obtained through *erasure* (the most relevant parts of the input–according to the explanation–are removed). We therefore follow this evaluation approach similar to Serrano and Smith (2019), Atanasova et al. (2020) and Nguyen (2018).[7]

5.1 Attention-based Importance Metrics

We opt using the following three input importance metrics by Serrano and Smith (2019):[8]

- $\boldsymbol{\alpha}$: Importance rank corresponding to normalised attention scores.
- $\nabla\boldsymbol{\alpha}$: Provides a ranking by computing the gradient of the predicted label $\hat{y}$ with respect to each attention score α_i in descending order, such that $\nabla\alpha_i = \frac{\partial\hat{y}}{\partial\alpha_i}$.
- $\boldsymbol{\alpha}\nabla\boldsymbol{\alpha}$: Scales the attention scores α_i with their corresponding gradients $\nabla\alpha_i$.

5.2 Faithfulness Metrics

Decision Flip - Most Informative Token: The average percentage of decision flips (i.e. changes in model prediction) occurred in the test set by removing the token with highest importance.

Decision Flip - Fraction of Tokens: The average fraction of tokens required to be removed to cause a decision flip in the test set.

Note that we conduct all experiments at the input level (i.e. by removing the token from the input sequence instead of only removing its corresponding attention weight) as we consider the scores from importance metrics to pertain to the corresponding input token following related work (Arras et al., 2016, 2017; Nguyen, 2018; Vashishth et al., 2019; Grimsley et al., 2020; Atanasova et al., 2020).

6 Experiments and Results

6.1 Data

We use five datasets for text classification following Jain and Wallace (2019): (i) **SST** (Socher et al., 2013); (ii) **IMDB** (Maas et al., 2011); (iii) **ADR**

[7] Note that Jacovi and Goldberg (2020) argue that a human evaluation is not an appropriate method to test faithfulness.

[8] Serrano and Smith (2019) show that gradient-based attention ranking metrics ($\nabla\boldsymbol{\alpha}$, $\boldsymbol{\alpha}\nabla\boldsymbol{\alpha}$) are better in providing faithful explanations compared to just using attention ($\boldsymbol{\alpha}$).

Dataset	Av. $\|W\|$	$\|\mathbf{V}\|$	Splits Train/Dev/Test
SST	20	13,686	6,920 / 872 / 1,821
ADR	22	6,716	14,452 / 2,551 / 4,251
IMDB	185	12,147	17,212 / 4,304 / 4,363
AG	34	14,573	60,895 / 7,145 / 3,960
MIMIC	2,180	16,277	4,654 / 822 / 1,369

Table 1: Dataset statistics including average words per instance, vocabulary size and splits.

Tweets (Sarker et al., 2015); (iv) **AG** News;[9] and (v) **MIMIC** Anemia (Johnson et al., 2016). See Table 1 for detailed data statistics.

6.2 Predictive Performance

A prerequisite of interpretability is to obtain robust explanations without sacrificing predictive performance (Lipton, 2016). Table 2 shows the macro F1-scores of all models across datasets, encoders and attention mechanisms using the three TaSc variants (Lin-TaSc, Feat-TaSc and Conv-TaSc described in Section 4) and without TaSc (No-TaSc).[10]

In general, all TaSc models obtain comparable performance and in some cases outperform No-TaSc across datasets and attention mechanisms. However, our main aim is not to improve predictive performance but the faithfulness of attention-based explanations, which we illustrate below.

6.3 Decision Flip: Most Informative Token

Table 3 and Figure 1 present the mean average percentage of decision flips (higher is better) across attention mechanisms, encoders and datasets by removing the most informative token for TaSc variants and No-TaSc for all attention-based importance metrics (see Section 5).

In Table 3, we observe that TaSc variants are effective in identifying the single most important token, outperforming No-TaSc in 12 out of 18 cases across attention-based importance metrics. This suggests that the attention mechanisms benefit from the non-contextualised information encapsulated in TaSc when allocating importance to the input tokens. Models using Tanh without TaSc appear to produce on average a higher percentage of decision flips compared to those using the Dot mechanism. Using either of the TaSc variants improves both

Data	Enc()	No-TaSc		Lin-TaSc		Feat-TaSc		Conv-TaSc	
		Dot	Tanh	Dot	Tanh	Dot	Tanh	Dot	Tanh
SST	BERT	.91	.90	.89	.88	.85	.88	.91	**.91**
	LSTM	.76	.75	**.79**	**.79**	**.79**	**.80**	**.78**	**.77**
	GRU	.76	.77	**.79**	**.78**	**.80**	**.79**	**.77**	.77
	MLP	.76	.76	**.78**	**.78**	**.79**	**.78**	**.79**	**.79**
	CNN	.76	.74	**.80**	**.78**	**.80**	**.80**	**.78**	**.76**
ADR	BERT	.80	.79	.78	.77	.79	.76	.78	.77
	LSTM	.74	.73	**.75**	**.75**	.74	**.75**	.73	**.75**
	GRU	.74	.73	**.76**	**.75**	.74	**.76**	.74	**.75**
	MLP	.74	.68	**.75**	**.74**	**.75**	**.74**	**.75**	**.74**
	CNN	.73	.69	**.75**	**.74**	**.74**	**.75**	**.76**	**.75**
IMDB	BERT	.93	.93	.93	.92	.92	.92	.93	.93
	LSTM	.89	.89	.88	.88	.88	.89	.89	.89
	GRU	.89	.90	.88	.88	.89	.89	.89	.89
	MLP	.88	.88	.88	.88	.88	.88	**.89**	.88
	CNN	.88	.88	.88	.88	.88	.88	.88	**.89**
AG	BERT	.94	.94	.94	.94	.94	.94	.94	.94
	LSTM	.92	.93	.92	.92	.92	.92	.92	.92
	GRU	.92	.92	.92	.92	.92	.92	.92	.92
	MLP	.92	.92	.92	.92	.91	.91	.92	.92
	CNN	.92	.92	.92	.92	.92	.92	.92	.92
MIMIC	BERT[11]	.82	.84	.82	.83	**.83**	.83	**.83**	.83
	LSTM	.87	.89	.87	.87	**.88**	.88	**.88**	.88
	GRU	.87	.89	.87	.88	**.88**	.88	**.88**	.88
	MLP	.87	.87	.87	.86	.86	.86	.87	.86
	CNN	.88	.89	.88	.87	.87	.87	.88	.88

Table 2: F1-macro average scores (3 runs) across datasets, encoders and attention mechanisms for models with and without TaSc (No-TaSc). Underlined and **bold** values indicate comparable and better predictive performance by using TaSc respectively. Standard deviations do not exceed 0.01

	Att.	No-TaSc	Lin-TaSc	Feat-TaSc	Conv-TaSc
α	**Tanh**	**8.4**	7.3 (0.9)	6.5 (0.8)	5.4 (0.6)
	Dot	**5.4**	4.3 (0.8)	4.8 (0.9)	4.5 (0.8)
$\nabla\alpha$	**Tanh**	8.2	10.2 (1.2)	**11.2** (1.4)	10.4 (1.3)
	Dot	6.9	10.9 (1.6)	**12.2** (1.8)	11.1 (1.6)
$\alpha\nabla\alpha$	**Tanh**	11.7	**14.0** (1.2)	13.5 (1.1)	12.2 (1.0)
	Dot	8.2	11.8 (1.4)	**12.6** (1.5)	11.3 (1.4)

Table 3: Mean average *percentage of decision flips* across attention mechanisms occurred by removing the most informative token, using the three TaSc variants and No-TaSc (higher is better). **Bold** and underlined values denote best performing method row-wise and overall (for each attention mechanism). Relative improvement over No-TaSc in parenthesis (>1 TaSc is better than No-TaSc).

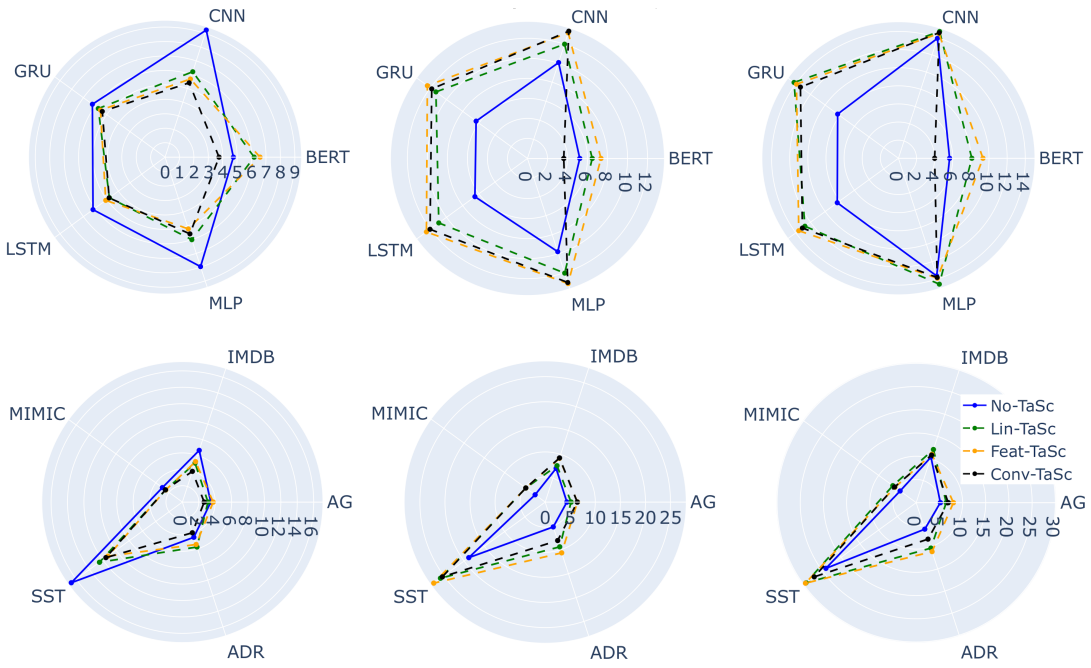
mechanisms, with Dot mechanism benefiting the most, making it comparable to Tanh. For example, Dot moves from 8.2% with No-TaSc to 11.8% with Lin-TaSc, which is closer to 14.0% achieved by Lin-TaSc with Tanh (for $\alpha\nabla\alpha$).

The first row of Figure 1 presents a comparison across encoders. TaSc variants achieve improved performance over No-TaSc across all encoder variants with $\nabla\alpha$ and $\alpha\nabla\alpha$. All TaSc variants yield comparable results with the exception

[9] https://di.unipi.it/~gulli/AG_corpus_of_news_articles.html

[10] For model hyper-parameters and prepossessing steps see Appendix A.

[11] Lower predictive performance is observed with BERT in MIMIC, as BERT accepts a maximum of 512 word pieces as input. See Appendix A.



(a) $\boldsymbol{\alpha}$ (b) $\boldsymbol{\nabla}\alpha$ (c) $\boldsymbol{\alpha}\nabla\alpha$

Figure 1: Mean average *percentage of decision flips* occurred by removing the most informative token, using the three TaSc variants and No-TaSc across encoders (first row) and datasets (second row), where lower is better.

of Conv-TaSc with BERT. Results further suggest that non-recurrent encoders (MLP, CNN) without TaSc outperform recurrent encoders (LSTM, GRU) and BERT which has the poorest performance. We hypothesise that this is due to the attention module becoming more important without feature contextualisation which is similar to findings of Serrano and Smith (2019) and Wiegreffe and Pinter (2019). However, we observe that using any of the TaSc variants across encoders results into improvements with LSTM and GRU becoming comparable to MLP and CNN. For example, BERT without TaSc improves from 5.7% to 8.0% (relative improvement 1.4x) and 9.3% (relative improvement 1.6x) using Lin-TaSc and Feat-TaSc respectively (for $\alpha\nabla\alpha$).

Observing results in the second row of Figure 1, we see that TaSc variants outperform No-TaSc in all datasets when using $\nabla\alpha$ and $\alpha\nabla\alpha$. This highlights the robustness of TaSc as improvements are irrespective of the dataset. In general, Lin-TaSc and Feat-TaSc perform equally well, however Lin-TaSc has the smaller number of parameters amongst the three variants. Similar to the findings of Serrano and Smith (2019) best results overall, irrespective of the use of TaSc, are obtained using $\alpha\nabla\alpha$ to rank importance.

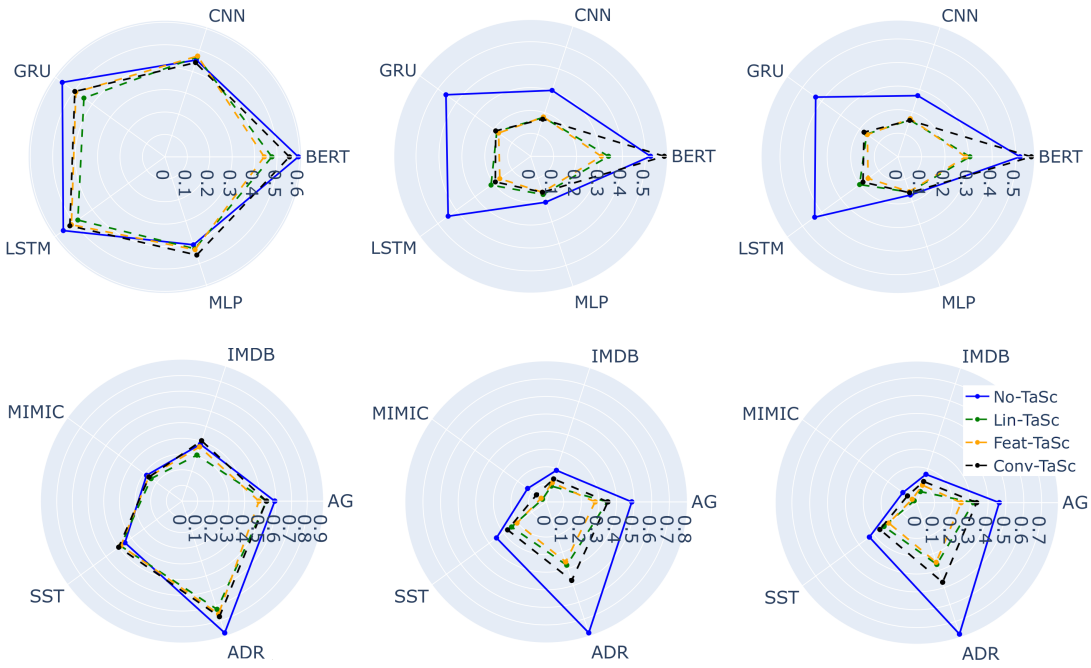
6.4 Decision Flip: Fraction of Tokens

Providing one token (i.e., the most informative) as an explanation is not always a realistic approach to assessing faithfulness. In our second experiment, we test TaSc by measuring the fraction of important tokens required to be removed to cause a decision flip (change model's prediction). Table 4 and Figure 2 show the mean average fraction of tokens required to be removed to cause a decision flip (lower is better) across attention mechanisms, encoders and datasets for all importance metrics.

	Att.	No-TaSc	Lin-TaSc	Feat-TaSc	Conv-TaSc
α	**Tanh**	.44	**.39** (0.9)	.42 (0.9)	.43 (1.0)
	Dot	.60	**.52** (0.9)	.53 (0.9)	.56 (0.9)
$\nabla\alpha$	**Tanh**	.36	.21 (0.6)	**.19** (0.5)	.26 (0.7)
	Dot	.42	**.22** (0.5)	**.22** (0.5)	.26 (0.6)
$\alpha\nabla\alpha$	**Tanh**	.32	**.17** (0.5)	.18 (0.5)	.24 (0.7)
	Dot	.41	**.21** (0.5)	**.21** (0.5)	.26 (0.6)

Table 4: Mean average *fraction of informative tokens* required to cause a decision flip across attention mechanisms, using the three TaSc variants and No-TaSc (lower is better). **Bold** and underlined values denote best performing method row-wise and overall (for each attention mechanism). Relative improvement over No-TaSc in parenthesis (<1 TaSc is better than No-TaSc).

In Table 4, we see that attention-based explanations from models trained with any of the TaSc mechanisms require on average a lower fraction of tokens to cause a decision flip compared to No-



(a) $\boldsymbol{\alpha}$ (b) $\boldsymbol{\nabla}\alpha$ (c) $\boldsymbol{\alpha}\nabla\alpha$

Figure 2: Mean average *fraction of tokens* required to cause a decision flip, using the three TaSc variants and No-TaSc across encoders (first row) and datasets (second row), where lower is better.

TaSc (in 17 out of 18 cases). Overall Lin-TaSc achieves higher or comparable relative improvements over Conv-TaSc and Feat-TaSc in 5 out of 6 times.

We present an across encoders comparison in the first row of Figure 2. All three TaSc variants obtain comparable performance with the exception of Conv-TaSc with BERT. We hypothesise that with BERT, Conv-TaSc fails to capture interactions between embedding dimensions due to perhaps higher contextualisation of BERT embeddings (i.e. contain more duplicate information). Similarly to the previous experiment results suggest that non-recurrent encoders (MLP and CNN) without TaSc outperform the remainder of encoders, with BERT having the worst performance. This strengthens our hypothesis that attention becomes more important to a model with reduced contextualisation. When using TaSc, performance across all encoders becomes comparable with the exception of BERT. For example, GRU improves from .43 with No-TaSc to .16 with Lin-TaSc, .17 with Feat-TaSc and .18 with Conv-TaSc (for $\alpha\nabla\alpha$).

The second row of Figure 2 presents results across datasets. All three TaSc mechanims manage to outperform vanilla attention. Lin-TaSc and Feat-TaSc perform comparably, with the first having a slight edge obtaining highest relative improvements in 3 out of 5 datasets with $\alpha\nabla\alpha$. For example in ADR, No-TaSc requires on average .77 of all tokens to be removed for a decision flip to occur compared to .34 obtained by Lin-TaSc (for $\alpha\nabla\alpha$). The benefits of TaSc become evident when considering longer sequences. For example in MIMIC, Lin-TaSc requires on average 44 tokens to cause a decision flip compared to 220 for No-TaSc.

6.5 Robustness Analysis

We also perform a detailed comparison between the best performing TaSc variant (Lin-TaSc) and vanilla attention (No-TaSc) across all test instances. Figure 3 shows box-plots with the median fraction of tokens required to be removed for causing a decision flip when ranking tokens by all three importance metrics. For brevity we present results for four cases.

We notice that the median fraction of tokens required to cause a decision flip for Lin-TaSc using α is higher compared to No-TaSc in certain cases. However, Lin-TaSc results in consistently lower medians (with substantially reduced variances) compared to No-TaSc using $\nabla\alpha$ and $\alpha\nabla\alpha$ which are more effective importance metrics. This is particularly visible in ADR using BERT, where the 25% and 75% percentiles are much closer to the median values, compared to No-TaSc. Reduced variances suggest that the explanation faithfulness across instances remains consistent.

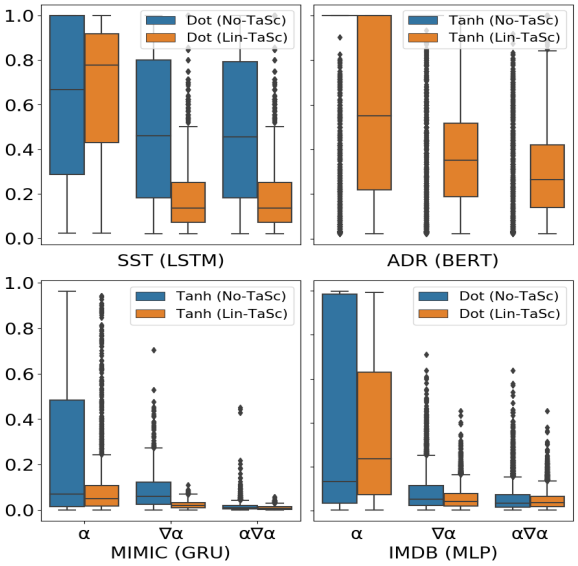


Figure 3: Box-plots of *fractions of tokens* removed across all test instances and importance metrics. ● denotes attention without TaSc; ● denotes attention with Lin-TaSc (lower and narrower is better).

7 Comparing TaSc with Non-attention Input Importance Metrics

We finally compare explanations provided by using Lin-TaSc and $\alpha\nabla\alpha$ to three standard non-attention input importance metrics without TaSc which are strong baselines for explainability (Nguyen, 2018; Atanasova et al., 2020).

Word Omission (WO) (Robnik-Šikonja and Kononenko, 2008; Nguyen, 2018): Ranking input words by computing the difference between the probabilities of the predicted class when including a word i and omitting it: $\text{WO}_i = p(\hat{y}|\mathbf{x}) - p(\hat{y}|\mathbf{x}_{\setminus x_i})$

InputXGrad ($\mathbf{x}\nabla\mathbf{x}$) (Kindermans et al., 2016; Atanasova et al., 2020): Ranking words by multiplying the gradient of the input by the input with respect to the predicted class: $\nabla x_i = \frac{\partial \hat{y}}{\partial x_i}$

Integrated Gradients (IG) (Sundararajan et al., 2017): Ranking words by computing the integral of the gradients taken along a straight path from a baseline input to the original input, where the baseline is the zero embedding vector.

Comparison Results Table 5 shows the results on decision flip (fraction of tokens removed) comparing the best performing attention-based importance metric ($\alpha\nabla\alpha$) with Lin-TaSc to Non-TaSc models with WO, $\mathbf{x}\nabla\mathbf{x}$ and IG importance metrics across all encoders and datasets.[12] We observe that using $\alpha\nabla\alpha$ with TaSc to rank word importance requires a lower fraction of tokens to cause a decision flip on average compared to WO, $\mathbf{x}\nabla\mathbf{x}$ and IG without TaSc. We outperform the other explanation approaches in 40 out of 50 cases, whilst obtaining comparable performance in other 5 cases. This demonstrates the efficacy of TaSc in providing more faithful attention-based explanations than strong baselines without TaSc (Nguyen, 2018; Atanasova et al., 2020). The improvements are particularly evident using BERT as an encoder. In IMDB, WO with Tanh requires on average .23 of the tokens to be removed for a decision flip compared to just .07 for $\alpha\nabla\alpha$ with TaSc.

We also observe that the attention-based importance metric ($\alpha\nabla\alpha$) with TaSc is a more robust explanation technique than non-attention based ones, obtaining lower variance in the fraction of tokens required to cause a decision flip across encoders. For example $\alpha\nabla\alpha$ with TaSc and Tanh requires a fraction of tokens in the range of .01-.05 compared to IG which requires .02-.43 in MIMIC, showing the consistency of our proposed approach.

Finally we observe that TaSc consistently improves non-attention based explanation approaches (WO, $\mathbf{x}\nabla\mathbf{x}$ and IG) requiring a lower fraction of tokens to be removed compared to Non-TaSc across encoders, datasets and attention mechanisms in the majority of cases (see full results in Appendix E).

8 Qualitative Analysis

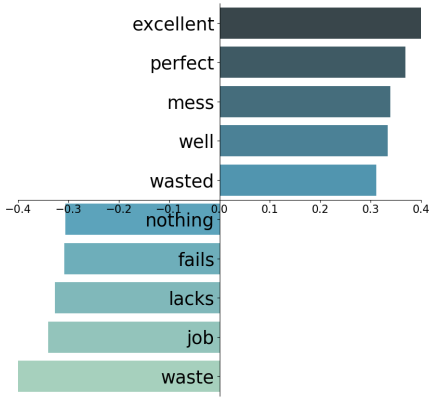
We finally examine qualitatively what type of information the parameter $\mathbf{u}$ from Lin-TaSc learns. Similar to a bag-of-words model, our initial hypothesis is that $\mathbf{u}$ will assign high scores to the words that are most relevant to the task. Figure 4 illustrates the 5 highest and lowest scored words from the IMDB and ADR datasets with a LSTM encoder and Dot attention and CNN encoder and Tanh attention respectively. For brevity we include two examples, however observations hold similar throughout other configurations (e.g. encoders, datasets) and when increasing the number of top-k words.

We first observe in 4a, that indeed words expressing sentiment are assigned with high scores (e.g. *excellent, waste, perfect*), either positive or negative. However, a positive or negative sign does

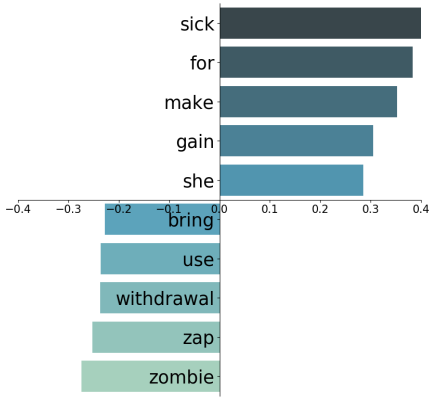
[12] We do not compare with LIME (Ribeiro et al., 2016) because WO and the gradient-based approaches outperform it (Nguyen, 2018; Atanasova et al., 2020).

Data	Enc()	Tanh Non-TaSc WO	Tanh Non-TaSc x∇x	Tanh Non-TaSc IG	Tanh TaSc α∇α	Dot Non-TaSc WO	Dot Non-TaSc x∇x	Dot Non-TaSc IG	Dot TaSc α∇α
SST	BERT	.29	.64	.51	**.22**	**.32**	.62	.49	.55
	LSTM	.25	.24	.20	**.19**	.21	.23	**.19**	**.19**
	GRU	.24	.22	.19	**.18**	.24	.25	.23	**.19**
	MLP	.36	.26	.24	**.18**	.22	.19	**.18**	**.18**
	CNN	.30	.25	.20	**.19**	.22	.20	**.18**	.19
ADR	BERT	.83	.91	.89	**.31**	.81	.90	.87	**.50**
	LSTM	.82	.81	.80	**.32**	.87	.88	.87	**.34**
	GRU	.84	.84	.84	**.35**	.79	.80	.80	**.38**
	MLP	.71	.63	.57	**.31**	.49	.43	**.39**	.40
	CNN	.80	.78	.78	**.37**	.77	.74	.74	**.36**
IMDB	BERT	.23	.69	.43	**.07**	.24	.72	.49	**.20**
	LSTM	.18	.12	.07	**.04**	.26	.09	.07	**.05**
	GRU	.18	.12	.07	**.04**	.27	.15	.08	**.05**
	MLP	.16	**.05**	**.05**	**.05**	.18	.07	.06	**.05**
	CNN	.21	.09	.07	**.05**	.27	.07	.06	**.05**
AG	BERT	.62	.78	.56	**.50**	.56	.76	**.60**	**.60**
	LSTM	.53	.51	**.30**	.38	.47	.52	**.35**	.46
	GRU	.45	.36	.31	**.20**	.54	.40	.30	**.22**
	MLP	.53	.24	.25	**.19**	.44	.25	.23	**.19**
	CNN	.55	.38	.28	**.20**	.53	.35	.25	**.21**
MIMIC	BERT	.24	.67	.43	**.03**	.21	.57	.26	**.05**
	LSTM	.35	.32	.12	**.01**	.28	.40	.30	**.01**
	GRU	.20	.24	.23	**.01**	.36	.18	.08	**.01**
	MLP	.40	.03	.22	**.01**	.13	.04	.03	**.02**
	CNN	.26	.15	.02	**.01**	.43	.09	**.02**	**.02**

Table 5: Average *fraction of tokens* required to cause a decision flip using the best performing attention-based ranking ($\alpha\nabla\alpha$) with TaSc, *Word omission* without TaSc (**WO**), *InputXGrad* without TaSc (∇**x**) and *Integrated Gradients* without TaSc (**IG**).



(a) IMDB - LSTM - Dot



(b) ADR - GRU - Tanh

Figure 4: Highest and lowest scored 5 words from learnable parameter **u** with LSTM encoder and Dot mechanism for the IMDB dataset.

not correspond to supporting the positive or negative class respectively. For example *withdrawal* in ADR can be considered relevant to positive class, yet it is negatively scored. Also *sick* can be considered a withdrawal symptom which is relevant to the negative class, yet it is positively scored. We speculate that this happens due to the complex non-linear relationships between the input words and the target classes learned by the model.

9 Conclusion

We introduced TaSc, a family of three encoder-independent mechanisms that induce context-independent task-specific information to attention. We conducted an extensive series of experiments showing the superiority of TaSc over vanilla attention on improving faithfulness of attention-based interpretability without sacrificing predictive performance. Finally, we showed that attention-based explanations with TaSc outperform other interpretability techniques. For future work, we will explore the effectiveness of TaSc in sequence-to-sequence tasks similar to Vashishth et al. (2019).

Acknowledgments

We would like to thank the anonymous reviewers for their constructive and detailed comments that helped to improve the paper. Nikolaos Aletras is supported by EPSRC grant EP/V055712/1, part of the European Commission CHIST-ERA programme, call 2019 XAI: Explainable Machine Learning-based Artificial Intelligence.

References

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2016. Explaining predictions of non-linear classifiers in

NLP. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 1–7.

Leila Arras, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2017. Explaining recurrent neural network predictions in sentiment analysis. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 159–168.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. A diagnostic study of explainability techniques for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3256–3274, Online. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*.

Francesco Barbieri, Luis Espinosa-Anke, Jose Camacho-Collados, Steven Schockaert, and Horacio Saggion. 2018. Interpretable emoji prediction via label-wise attention LSTMs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4766–4771.

Jasmijn Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2963–2977, Florence, Italy. Association for Computational Linguistics.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Ilias Chalkidis, Manos Fergadiotis, Dimitrios Tsarapatsanis, Nikolaos Aletras, Ion Androutsopoulos, and Prodromos Malakasiotis. 2021. Paragraph-level rationale extraction through regularization: A case study on European court of human rights cases. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 226–241, Online. Association for Computational Linguistics.

Hanjie Chen and Yangfeng Ji. 2020. Learning variational word masks to improve the interpretability of neural text classifiers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4236–4251, Online. Association for Computational Linguistics.

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 452–461.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. 2016. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3504–3512.

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.

Reza Ghaeini, Xiaoli Fern, and Prasad Tadepalli. 2018. Interpreting recurrent and attention-based neural models: a case study on natural language inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4952–4957, Brussels, Belgium. Association for Computational Linguistics.

Christopher Grimsley, Elijah Mayfield, and Julia R.S. Bursten. 2020. Why attention is not explanation: Surgical intervention and causal reasoning about neural models. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1780–1790, Marseille, France. European Language Resources Association.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–80.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online. 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.

Sarthak Jain, Sarah Wiegreffe, Yuval Pinter, and Byron C. Wallace. 2020. Learning to faithfully rationalize by construction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4459–4473, Online. Association for Computational Linguistics.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hervé Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *CoRR*, abs/1612.03651.

Pieter-Jan Kindermans, Kristof Schütt, Klaus-Robert Müller, and Sven Dähne. 2016. Investigating the influence of noise and distractors on the interpretation of neural networks. *arXiv preprint arXiv:1611.07270*.

Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. 1999. Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pages 319–345.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas. Association for Computational Linguistics.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016a. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016b. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

Tal Linzen, Grzegorz Chrupała, Yonatan Belinkov, and Dieuwke Hupkes, editors. 2019. *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Zachary C Lipton. 2016. The mythos of model interpretability. int. conf. In *Machine Learning: Workshop on Human Interpretability in Machine Learning*.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38.

Akash Kumar Mohankumar, Preksha Nema, Sharan Narasimhan, Mitesh M. Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. 2020. Towards transparent and explainable attention models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4206–4216, Online. Association for Computational Linguistics.

Dong Nguyen. 2018. Comparing automatic and human evaluation of local explanations for text classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1069–1078, New Orleans, Louisiana. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C. Lipton. 2020. Learning to deceive with attention-based explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4782–4793, Online. Association for Computational Linguistics.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50.

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 97–101, San Diego, California. Association for Computational Linguistics.

Marko Robnik-Šikonja and Igor Kononenko. 2008. Explaining classifications for individual instances. *IEEE Transactions on Knowledge and Data Engineering*, 20(5):589–600.

Abeed Sarker, Rachel Ginn, Azadeh Nikfarjam, Karen O'Connor, Karen Smith, Swetha Jayaraman, Tejaswi Upadhaya, and Graciela Gonzalez. 2015. Utilizing social media data for pharmacovigilance: a review. *Journal of Biomedical Informatics*, 54:202–212.

Cansu Sen, Thomas Hartvigsen, Biao Yin, Xiangnan Kong, and Elke Rundensteiner. 2020. Human attention maps for text classification: Do humans and neural networks focus on the same words? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4596–4608, Online. Association for Computational Linguistics.

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.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

Xiaobing Sun and Wei Lu. 2020. Understanding attention for text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3418–3428, Online. Association for Computational Linguistics.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org.

Marcos Treviso and André F. T. Martins. 2020. The explanation game: Towards prediction explainability through sparse communication. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 107–118, Online. Association for Computational Linguistics.

Martin Tutek and Jan Snajder. 2020. Staying true to your word: (how) can attention become explanation? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 131–142, Online. Association for Computational Linguistics.

Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, and Manaal Faruqui. 2019. Attention interpretability across NLP tasks. *arXiv preprint arXiv:1909.11218*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20.

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. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.

Zhongheng Zhang, Marcus W Beck, David A Winkler, Bin Huang, Wilbert Sibanda, Hemant Goyal, et al. 2018. Opening the black box of neural networks: methods for interpreting neural network models in clinical applications. *Annals of Translational Medicine*, 6(11).