The University Of Sheffield.

The University of Sheffield

Model Interpretability for Natural Language Processing Applications

A thesis submitted for the degree of Doctor of Philosophy

in the

Department of Computer Science

George Chrysostomou

Supervisor: Dr. Nikolaos Aletras

Declaration

I, George Chrysostomou, hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted for any other degree or qualification in this, or any other university. All work presented is my own and does not contain work as a result of collaboration with others, except as specified in the text and Acknowledgements.

Acknowledgments

I would first like to thank my supervisor Nikolaos Aletras, for the invaluable guidance through my studies. I am really happy I followed his advice before and during the PhD, as it allowed me to progress, develop and grow both as a person and as a researcher. Working with Nikos has made the three years of my studies very enjoyable.

I would also like to thank the panel committee members, Mauricio Alvarez Lopez and Heidi Christensen for their constructive feedback and advice through the project.

Also, I would like to thank all my colleagues and friends in room G35 for making working at the university a real pleasure. I could not wish for a better work environment and I sincerely hope they achieve all their dreams.

Finally, I would like to thank my family for their love and support. Their help, patience and advice has made this possible and I will always appreciate that. Whilst hesitant at first with continuing further studies, having made my decision they were there to support me through it all.

Abstract

This thesis focuses on model interpretability, an area concerned with understanding model predictions in Natural Language Processing (NLP) tasks. The increase in adoption of opaque models, such as BERT, leads to an increasing need for explaining their predictions. This is typically performed by extracting a sub-set of the input, that is indicative of the true reasoning behind the model's prediction (i.e. a faithful explanation or rationale).

Whilst there are multiple approaches in literature for extracting explanations (e.g. feature attribution methods), some faced criticism about how faithful they are. Furthermore, explanation faithfulness also depends on the model employed, where highly parametrised models have been shown to produce less faithful explanations. Previous research has also shown that there is no single best feature attribution method across models, tasks and even instances of the same dataset, whilst finding a rationale length is still an open problem. Additionally, a limitation of current evaluations for explanation faithfulness, is that they are performed on a held-out dataset coming from the same domain (i.e. the data they are evaluated on, are from the same distribution as the training data). However, we are not aware how faithfulness is impacted in out-of-domain settings.

The main aim of this thesis therefore, is to improve and evaluate the faithfulness of explanations in NLP applications. First, we improve the faithfulness of explanations extracted using attention mechanisms, a popular component used in neural NLP models. In a similar direction, we show improvements on the faithfulness of explanations from feature attribution approaches, when using large language models. We then address the problem of specifying a priori a feature scoring method, rationale length and type. Finally, we evaluate the faithfulness of explanations in out-of-domain settings, highlighting a problem when using popular faithfulness evaluation metrics.

Contents

List of Figures

List of Tables

Nomenclature

NLP	Natural Language Processing
LMs	Language Models
MLP	Multi-Layer Perceptron
RNN	Recurrent Neural Networks
CNN	Convolutional Neural Networks
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
BERT	Bidirectional Encoder Representations from Transformers
IG	Integrated Gradients
LIME	Local Interpretable Model-agnostic Explanations
FRESH	Faithful Rationale Extraction from Saliency tHresholding
ID	In-Domain
OOD	Out-Of-Domain
KL	Kullback–Leibler
JSD	Jensen–Shannon

Chapter 1

Introduction

Natural Language Processing (NLP) technologies are increasingly becoming more reliant on complex neural network models, with large pre-trained transformer-based language models (LMs) such as BERT (Devlin et al., 2019), currently dominating performance across language understanding benchmarks (Wang et al., 2018). Their high predictive capabilities and efficiency in dealing with large amounts of data, have expanded their application in safety-critical tasks, such as the medical domain (Johnson et al., 2016) and the justice system (Chalkidis et al., 2020a). However, a significant drawback of these models is their opaqueness in providing predictions due to their highly parameterised and interconnected architecture, earning them the title of "black boxes" (Zhang et al., 2018; Linzen et al., 2019). A growing concern as such is whether their use in safety-critical domains is responsible, safe or even ethical (Madsen et al., 2021b).

Pivotal to mitigating these concerns, is understanding model predictions by providing a justification (e.g. an explanation or rationale). This has led to increased interest by researchers towards model interpretability, an area concerned with defining and identifying approaches for understanding model predictions. The growth of this field within the NLP community is evident by the evolution towards what constitutes a high quality interpretation, with an increasing amount of methods and approaches developed to provide a justification by generating explanations or rationales (Ribeiro et al., 2016; Kindermans et al., 2016; Bastings and Filippova, 2020; Guerreiro and Martins, 2021).[1] Rationales are snippets, representing

[1]These terms are used interchangeably through this work

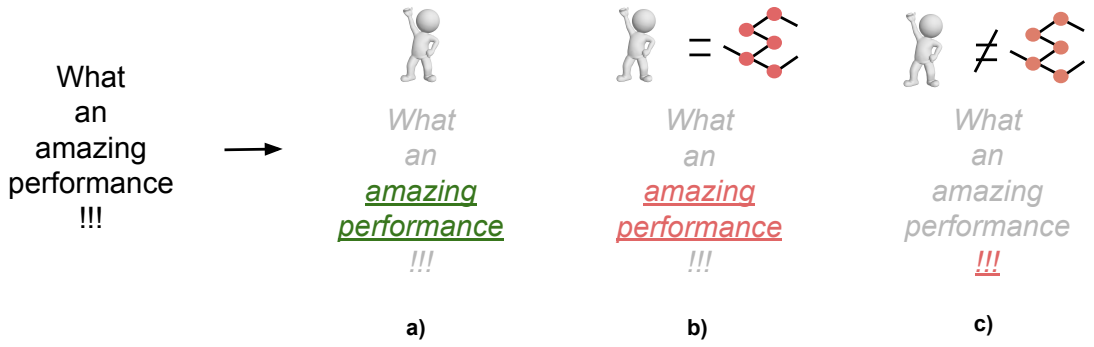
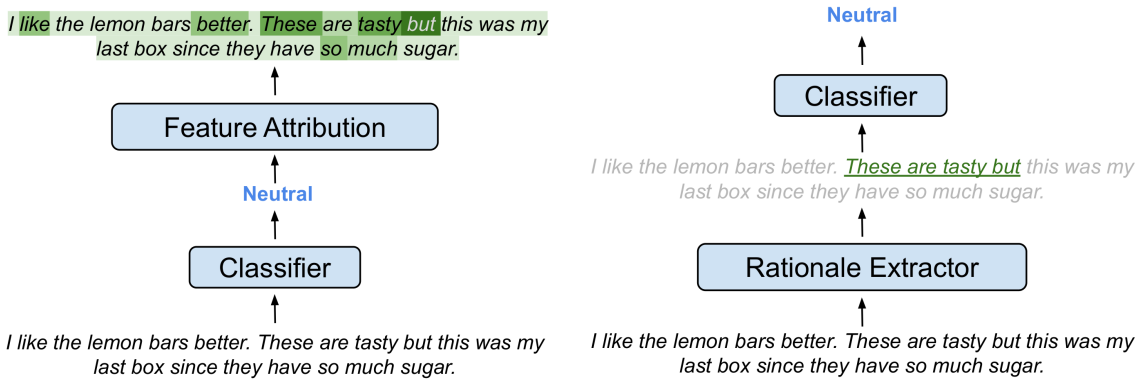


Figure 1.1: Demonstrative example of different rationale types: a) human rationale; b) plausible rationale; and c) faithful rationale. A plausible rationale is one which agrees with human rationales (a = b), whilst a faithful rationale does not necessarily entail that (a ≠ c).

the most important parts of the input that a model uses to make a prediction, and are such a subset of the input.

Two important categorisations of rationales are *plausibility* and *faithfulness* (Jacovi and Goldberg, 2020). A plausible rationale is one that is intuitive and aligns with human understanding. A faithful rationale is one that accurately describes the reasoning behind a model's prediction. These two properties do not correlate, such that a plausible rationale is not necessarily faithful and vice versa (Atanasova et al., 2020). However, this does not exclude that a faithful rationale can also be plausible. For clarity, this distinction can be demonstrated in an example of a positive sentiment prediction using a movie review snippet shown in Figure 1.1.

Through this example we can observe that using the faithful rationale, for a user it would not be intuitive that the model has selected "***!!!***" instead of the word "*amazing*". However this demonstrates that what we find intuitive is not often what the model uses to make a prediction. This can be also caused by a model attaching to dataset-specific artefacts or by spurious correlations learned during training (Glockner et al., 2020). In fact, plausible rationales seem to fit the description of model interpretability as given by Lipton (2016), which is "*a measure of how understandable by an observer is an explanation of why a model made a particular prediction for a single input*". However, Jacovi and Goldberg (2020) argue that faithfulness is an important property of rationales as it provides us with a better understanding of what reasoning the model followed, despite of how intuitive it is.



(a) Post-hoc explanation **(b)** Select-then-predict model

Figure 1.2: Examples of rationale extraction using two approaches. (a) uses a feature attribution approach to identify the most important subset of the input (post-hoc explanation). Darker coloured heatmaps indicate a higher importance score (i.e. more important for a model's prediction). (b) extracts rationales using inherently faithful, select-then-predict models. These models are inherently faithful, as the classifier is trained **only** on the rationale.

In literature there are two popular approaches for extracting rationales (see examples in Figure 1.2). The first (see example (a)) uses feature scoring methods to identify the most important segments of the input (Arras et al., 2017; Ribeiro et al., 2016; Jain and Wallace, 2019), i.e. post-hoc explanations. Feature scoring methods attribute importance to the input tokens, with highly scored tokens indicating that they have contributed more towards a model's predictions. The latter (see example (b)) consists of training inherently interpretable models (also referred to as select-then-predict), consisting of two modules: a generator and an encoder (Lei et al., 2016; Bastings et al., 2019; Jain et al., 2020; Treviso and Martins, 2020; Guerreiro and Martins, 2021). The generator is responsible for generating a rationale mask, whilst the encoder is trained on an end-task (e.g. text classification). The encoder is thus inherently faithful, as it is trained using only the rationales as input. A drawback of these models, is that they sacrifice predictive performance to achieve inherent interpretability, as in the large majority of cases they do not reach the predictive performance of a model trained on the full text (Guerreiro and Martins, 2021). Recent studies have used feature scoring methods as part of the generator in select-then-predict models (Jain et al., 2020; Treviso and Martins, 2020).

One prominent example of a feature scoring method are attention mechanisms (Bahdanau et al., 2015). Attention mechanisms produce a probability distribution over the input

by computing 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; Sun and Lu, 2020; Jain et al., 2020). However, the efficacy of attention mechanisms in producing faithful explanations has been recently been disputed. A series of studies showed that that explanations obtained by attention weights do not always provide faithful explanations (Jain and Wallace, 2019; 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). Evaluation in these studies was mostly conducted by comparing against another popular feature scoring method, the gradients of the prediction computed with respect to the input. Contradicting previous studies, more recent work has shown that attention mechanisms *can* produce faithful explanations, in certain cases even surpassing the previous "gold standard" gradient-based feature scoring method (Jain et al., 2020; Madsen et al., 2021a). 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 (Pascual et al., 2020; Tutek and Snajder, 2020).

LMs such as BERT can provide faithful explanations through feature scoring methods, particularly using attention (Jain et al., 2020), but still fall behind other simpler architectures, such as the multi-layer perceptron (Atanasova et al., 2020), possibly due to increased information mixing and higher contextualisation in the model (Brunner et al., 2020; Pascual et al., 2020). Multiple studies have attempted to improve the explainability of non transformer-based models, by guiding them through an auxiliary objective towards informative input importance distributions (e.g. human or adversarial priors) (Ross et al., 2017a; Liu and Avci, 2019; Moradi et al., 2021). In fact, auxiliary training objectives have also been used to assess the robustness of input importance distributions produced by attention mechanisms (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019). These show the efficacy of auxiliary objectives in impacting the faithfulness of explanations generated via feature scoring methods.

Atanasova et al. (2020) also show that there is no single best feature scoring method across models, datasets and even within the same dataset. Additionally, rationales are usually extracted using a pre-defined fixed length (i.e. the ratio of a rationale compared to the full

input sequence) across all instances in a dataset. Using a fixed length or type for different instances could result into shorter (i.e. not sufficient for explaining a model's prediction) or longer than needed rationales reducing rationale faithfulness, whilst finding the explanation length is an open problem (Zhang et al., 2021). Moreover to extract rationales, practitioners are currently required to make assumptions for the rationale parameters (i.e. feature scoring method, length and type), whilst different choice of parameters might substantially affect the faithfulness of rationales.

Extracting faithful explanations can help improve *trust* in a model's prediction. Currently, the faithfulness of both post-hoc explanations and rationales from select-then-predict models, has mostly been evaluated on in-domain settings (i.e. the train and test data come from the same distribution). However, when deploying models in real-world applications, inference might be performed on data from a different distribution, i.e. out-of-domain (Desai and Durrett, 2020; Ovadia et al., 2019). This can create implications when extracted explanations (either using post-hoc methods or through select-then-predict models) are used for assisting human decision making. Whilst we are aware of the limitations of current state-of-the-art models in out-of-domain predictive performance (Hendrycks et al., 2020), how faithful out-of-domain post-hoc explanations are has yet to be explored. Similarly, we are not aware how inherently faithful select-then-predict models generalize in out-of-domain settings.

1.1 Research Questions

The previous discussion shows that providing explanations for model predictions is a rapidly evolving field, critical for improving trust and understanding for otherwise opaque model predictions. Whilst there is growing number of approaches for extracting explanations and we are closer to a consensus of what constitutes a quality interpretation, there are still significant challenges to address. These challenges give rise to the following research questions, that we seek to address in this thesis:

1. Attention mechanisms have been criticised for their ability to produce faithful explanations. Additionally it has been shown that their performance is encoder-dependent. Is it possible to improve their faithfulness, regardless of the encoder? Background on

the attention mechanisms and previous studies on evaluating the faithfulness of their explanations is further discussed in Chapter 2;

2. Large pre-trained transformer models have been empirically shown to produce faithful explanations, however still fall behind simpler model architectures. Building on previous research that shows explanation faithfulness can be influenced by learning with auxiliary objectives, can we improve the faithfulness of their explanations by informing their learning through other, informative salient importance distributions?

3. Given all the recently developed tools for generating explanations, there is empirical proof that there is no single feature scoring method that produces the most faithful rationales across all instances in a dataset (Atanasova et al., 2020; Jacovi and Goldberg, 2020). This also applies for pre-defined fixed rationale lengths (Zhang et al., 2021) and therefore possibly for rationale types (i.e. an explanation consists of either independent words from the input sequence or a contiguous segment). Can we select these at instance-level rather than globally (across all instances in a dataset) in order to provide more faithful explanations?

4. Currently explanation faithfulness (for both post-hoc explanations and select-then-predict models) is evaluated in-domain. This raises the question: How is post-hoc explanation faithfulness and select-then-predict performance affected out-of-domain?

1.2 Main Contributions

This thesis makes the following contributions towards tackling the previously discussed research questions:

1. Proposes a new family of mechanisms to improve the faithfulness of attention-based explanations, by scaling the attention weights using a non-contextualised score. The motivation, methodology, experiments and empirical results are detailed in Chapter 3.1;

2. Introduces an auxiliary objective that guides the model towards an alternative informative salient distribution during training, to improve post-hoc explanation faithfulness

in large pre-trained transformer-based models. Details, experiments and results in Chapter 3.2;

3. A simple yet effective method is presented in Chapter 4, which allows the selections of instance-specific (1) feature scoring method; (2) rationale length and (3) rationale type. This helps mitigate the assumptions a practitioner needs to make during rationale extraction and shows that rationales extracted with the proposed method are more comprehensive and highly sufficient;

4. In Chapter 5 an extensive empirical study that examines how explanation faithfulness is affected when moving from in-domain to out-of-domain settings. We show that current metrics for post-hoc explanation faithfulness are misleading in out-of-domain settings, whilst select-then-predict models demonstrate comparable predictive performance in out-of-domain settings to full-text trained models.

1.3 Publications

Material from this thesis has been published in the following peer reviewed conferences:

- The work presented in Chapter 3.1 has been published at the Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL&IJCNLP 2021) (Chrysostomou and Aletras, 2021b);

- The work presented in Chapter 3.2 has been published at the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021) (Chrysostomou and Aletras, 2021a);

- The work presented in Chapter 4 has been published at the 36th AAAI Conference on Artificial Intelligence (AAAI-22).

- The work presented in Chapter 5 has been accepted at the Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)

1.4 Structure of Thesis

Chapter 2 begins with prerequisite knowledge on neural networks and then presents a detailed background on model interpretability, including a critical analysis of previous work. This includes how it is defined in literature and the definition adopted by this work. We then discuss approaches to extract explanations and finally how faithfulness is currently evaluated and improved.

Chapter 3 presents two novel approaches for improving the faithfulness of post-hoc explanations. More specifically, Chapter 3.1 presents the first contribution on improving attention-based explanations for text classification. Chapter 3.2 presents a novel auxiliary objective for improving the faithfulness of transformer-model based explanations. Both sections begin with the problem motivation, the methodology adopted, experimental setup and results.

Chapter 4 We motivate the adoption of instance-specific rationalisation in NLP and propose a novel method to achieve this. Followed by the description of the proposed approach, this chapter then includes the experiments and results that empirically show that rationales extracted with the proposed method are more faithful.

Chapter 5 first motivates the need for evaluating faithfulness of post-hoc explanations and select-then-predict models in out-of-domain settings. The experimental setup then covers the range of approaches adopted to extract explanations, followed by the results which highlight a problematic behaviour of current popular faithfulness evaluation metrics in out-of-domain settings.

Chapter 6 provides a summary of the contributions made throughout this thesis and the research questions tackled by them. Finally, based on these contributions we propose potential areas of future research.

Chapter 2

Background

The aim of this chapter is to provide a background understanding around the area of model interpretability and will identify and highlight limitations of previous work. It first begins by briefly describing some prerequisites, which are necessary for introducing and motivating the problem of model interpretability. These include popular NLP tasks and a brief background on different neural network architectures (Sections 2.2 and 2.3). Secondly, this chapter will focus on model interpretability, with a focus on generating explanations (or rationales) for model predictions. This includes: (1) what are explanations, how they are defined in literature and which definition is adopted by this work (Section 2.4); (2) how explanations are extracted (Section 2.5); (3) how they are evaluated (Sections 2.6 and 2.6.1) and finally (4) how previous research has attempted to improve them (Section 2.7).

2.1 Supervised NLP Task Setup

We begin by describing typical supervised NLP task setups that are used in this work. In supervised learning, models are trained using annotated labels provided with the data (Kotsiantis et al., 2007). In Table 2.1, we include demonstrative examples of popular supervised NLP tasks, which we describe in detail below.

A popular NLP task is text classification, where the goal is to "teach" a model $\mathcal{M}$ to predict the label $\hat{y}$ of a text sequence $\mathbf{x}$, such as predicting the sentiment of movie reviews

Task	**Input 1 (x)**	**Input 2 (q)**	**Labels**
Sentiment Analysis	*Great movie !!*	-	Positive Negative
Topic Classification	*Stocks are at a historic low.*	-	World Business Sports Politics
Question Answering	*Mary is home sleeping.*	*Where is Mary?*	Office Home Outside
Reading Comprehension	*Today the weather is much warmer than yesterday. However, today is rainy.*	*Yesterday was raining.*	True False

Table 2.1: Examples of supervised NLP tasks, which require either one or two text sequences as input and their possible labels.

(Socher et al., 2013) or the topic of news articles (Corso et al., 2005). In contemporary neural approaches, $\mathcal{M}$ is comprised of four components: (1) the embedding layer; (2) an encoder Enc(); (3) an attention mechanism (described in Section 2.2) and (4) a classification layer.

Therefore, a pipeline for text classification consists first of one-hot-encoded tokens $x_i \in \mathrm{R}^{|\mathrm{V}|}$ that are 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 embedding $\mathbf{e}_i$ is a vector representation of the token x_i which can be learned along-side the model during training. Alternatively, we can preload embeddings which have been trained on a large corpora (Mikolov et al., 2013; Pennington et al., 2014). The embeddings $\mathbf{e}_i$ are then passed to an encoder to produce contextualised hidden representations $\mathbf{h}_1, \ldots, \mathbf{h}_t = Enc(\mathbf{e}_1, \ldots, \mathbf{e}_t)$, where $\mathbf{h}_i \in \mathrm{R}^{\mathrm{N}}$, with N the size of the hidden representation. $Enc()$ can represent any suitable neural network architecture (described in Section 2.3). 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 (see Section 2.2 for more details).

$$\mathbf{c} = \sum_{i=1}^{t} \mathbf{h}_i \alpha_i, \quad \mathbf{c} \in \mathrm{R}^{\mathrm{N}} \tag{2.1}$$

The representation $\mathbf{c}$ is finally passed to the output layer, which is a fully-connected linear

layer, followed by an activation function.

Another task setting in NLP, is to use two text sequences to the model $\mathcal{M}$. Popular tasks with this setup are question answering, reading comprehension (Khashabi et al., 2018; DeYoung et al., 2020) and next sentence prediction (Devlin et al., 2019). For example, in question answering the model $\mathcal{M}$ usually presented with a context sequence $\mathbf{x}$ and a question sequence $\mathbf{q}$. The task is to find the answer to the question $\mathbf{q}$ in the context $\mathbf{x}$. Similar to text classification the context $\mathbf{x}$ and question $\mathbf{q}$ undergo the same procedures to obtain representations, $\mathbf{h}_i^x$ and $\mathbf{h}_i^q$. Both representations $[\mathbf{h}_i^x, \mathbf{h}_i^q]$ are then used to compute the attention scores α_i. To then reach a prediction, the same process is applied as in text classification. We obtain the context vector $\mathbf{c}$ for the entire context sequence $\{x_1, ..., x_t\}$ as the sum of $\mathbf{h}_i^x$ weighted by attention scores α_i. The vector $\mathbf{c}$ is then passed to a fully-connected linear layer, followed by an activation function return the prediction $\hat{y}$.

While there are other popular tasks in NLP, such as sequence-to-sequence tasks (e.g. machine translation or natural language generation), these have not been used for the experiments of this thesis. This work focuses only on binary or multi-class classification tasks, that require as input a text sequence (or two text sequences) and return an output distribution $\mathcal{Y}$ over classes, where the predicted label $\hat{y} \in \mathcal{Y}$.

2.2 Attention Mechanisms

We have previously described how the encoded representations $\mathbf{h}_i$ are weighted by attention scores α_i. In this section we are going to describe in detail the attention mechanisms that compute α_i.

Attention mechanisms were first introduced for aiding models in word alignments in neural machine translation (Bahdanau et al., 2015). However they were quickly adapted to various NLP tasks, such as text classification due demonstrating improved predictive performance (Yang et al., 2016). Additionally, attention mechanisms form the backbone of the transformer architecture (Vaswani et al., 2017), which is discussed in Section 2.3.4, arguably one of the most influential model architectures in contemporary NLP.

Attention mechanisms produce a probability distribution over the input tokens. As de-

scribed previously, they operate over the word representations resulting from the encoder. Attention weights (α_i), the elements of the distribution, are obtained by passing the contextualised representations ($\mathbf{h}_i$) to the attention mechanism, which usually consists of a similarity function $\phi()$ followed by a softmax activation function:

$$\alpha_i = \frac{\exp^{\phi(\mathbf{q},\mathbf{h_i})}}{\sum_{k=1}^{t} \exp^{\phi(\mathbf{q},\mathbf{h_k})}} \tag{2.2}$$

where $\mathbf{q} \in R^N$ is a query vector in a sequence to sequence or a question answering task. In a machine translation task for example the query vector would represent the decoder output state $\mathbf{s}_t$ corresponding to output word at position t, as the aim of the attention mechanism is to compute the alignment score of output word y_t and input word x_i. In a text classification task, $\mathbf{q}$ represents a trainable self-attention vector similar to Yang et al. (2016). There are multiple variants of similarity functions or scoring functions in literature. Below, we describe two popular scoring functions that are used in our experiments:

Tanh Attention (Tanh): was first introduced by Bahdanau et al. (2015). In this implementation (Yang et al., 2016), the similarity function is as follows:

$$\phi(\mathbf{s}_t, \mathbf{h}_i) = \mathbf{v}^T \tanh(\mathbf{W}[\mathbf{s}_t; \mathbf{h}_i]) \tag{2.3}$$

where $\mathbf{v}$ and W are trainable model parameters, $\mathbf{s}_t$ is the decoder network's hidden state output for the output word at position t and $\mathbf{h}_i$ the encoder output for the input word at position i. In text-classification, the similarity function can be adapted as (Jain and Wallace, 2019):

$$\phi(\mathbf{q}, \mathbf{h}_i) = \mathbf{q}^T \tanh(\mathbf{W}\mathbf{h}_i) \tag{2.4}$$

where the similarity between a learnable parameter $\mathbf{q}$ and the contextualised representation $\mathbf{h}_i$ of input word at position i is learned.

Scaled Dot-Product (Dot): is a similarity function first proposed by Luong et al. (2015), where similarity scores are computed by:

$$\phi(\mathbf{q}, h_i) = \frac{\mathbf{h}_i^T \mathbf{q}}{\sqrt{N}} \tag{2.5}$$

where the query vector $\mathbf{q}$ in a machine translation task is the decoder output state $\mathbf{s}_t$, and N is the dimension of $\mathbf{h}_i$. Vaswani et al. (2017) formed this adaptation from Luong et al. (2015), by adding the scaling factor $\frac{1}{\sqrt{N}}$.

2.3 Neural Text Encoders

In the previous section we presented attention mechanisms, which operate over the contextualised representations produced by the encoder ($\mathbf{h}_i$). This section will therefore describe popular neural text encoders, that map the embeddings ($\mathbf{e}_i$) to latent representations $\mathbf{h}_i$.

2.3.1 Multi-Layer Perceptron (MLP)

A neural network can be defined as an interconnected network of *nodes*, or *neurons*, with the processing memory of the network stored in the connections between the neurons, the *weights*, which are obtained by a *learning process* (Gurney, 2014). Neural networks bare their roots from linear regression models and more specifically to the perceptron. Linear regression learns a linear relationship between a response and explanatory variables (e.g. the text sequence $\mathbf{x}$ and a label y) (Montgomery et al., 2012). McCulloch and Pitts (1943) formulated a mathematical model as an attempt to mimic brain activity, by using interconnected nodes to project input to an output space. This model though lacked of a learning process. Building on top of this model, Rosenblatt (1958) conceived the perceptron with a learning mechanism by assigning weights to the connections rather than the inputs. McCulloch and Pitts (1943) then proved that with a perceptron they could model the OR/AND/NOT logical functions, as neuron activations above a certain threshold return a 1 and 0 otherwise. The perceptron can be formally defined as:

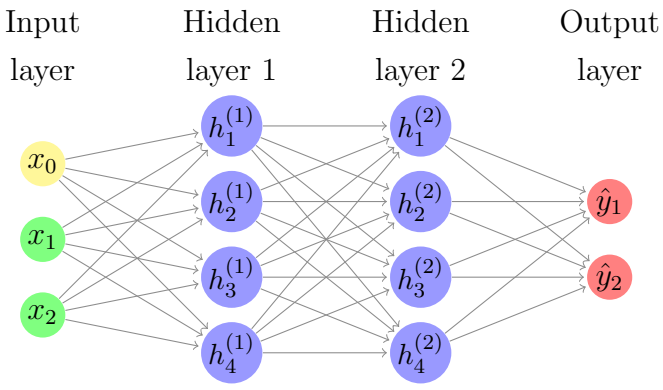
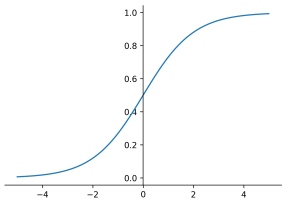


Figure 2.1: Illustration of the Multi-Layer Perceptron (MLP). Arrows represent the connections (weights) between the nodes.

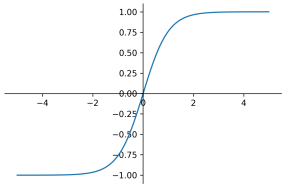
$$\hat{y} = \begin{cases} 1 & \text{if } \sum_{i=0}^{t} w_i x_i + b > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

where w_i is a weight multiplied by the input x at position i, added with the bias term b to produce a value $\hat{y}$. The learning mechanism is activated when the prediction does not match the expected label. The update process included negating the features that led to the incorrect prediction and adding the features that contribute towards the correct. The addition and negation can also be multiplied by a *learning rate*, which scales the effect of the weight update.

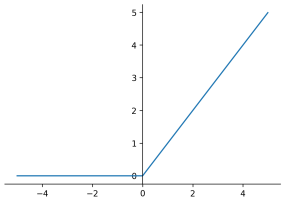
As the perceptron is a linear model, it cannot model the XOR problem (McCulloch and Pitts, 1943) or other non-linear problems. This limitation was tackled by the Multi-Layer Perceptron (MLP), or Feed-forward Neural Network, which comprises of multiple perceptron nodes stacked in layers, illustrated in Figure 2.1 (Minsky, 1968). Between the hidden layers and the output layer an activation function is introduced. Activation functions are non-linear functions that allow the model to compute non-linearly solvable problems. There are several activation functions common in literature, with the most popular being: the sigmoid , tanh and ReLU. The activation functions can be described by the following equations and can be



(a) sigmoid



(b) tanh



(c) ReLU

Figure 2.2: Visual illustration of the activation functions as described in equation 2.7 with input $x \in [-5, 5]$

visualised as in Figure 2.2:

$$\begin{aligned} sigmoid(x) &= \frac{1}{1 + \exp^{-x}} \\ \tanh(x) &= \frac{\exp^{x} - \exp^{-x}}{\exp^{x} + \exp^{-x}} \\ ReLU(x) &= max(0, x) \end{aligned} \tag{2.7}$$

Having multiple nodes stacked in a layer allows each node to learn different decision boundaries while the activation functions applied on the nodes allow modelling of non-linear problems. An MLP with a single layer and sigmoid activation can be viewed as a logistic regression model, whereby:

$$\hat{y} = \frac{1}{1 + exp^{-(Wx+b)}}$$

Similarly to the perceptron, the major problem in its initial formulation was the lack of a learning process. Whilst the training process of neural models is described in detail in Section 2.3.6, it is essential to introduce a key component of the learning procedure as several neural architectures were developed with the aim of improving it. Prior to updating the model parameters the contribution of each parameter with respect to the error of the output if first calculated. Their contribution is calculated via computing the derivative of the output with respect to the parameter. This is achieved through the *chain rule* which us to obtain the derivative of the composition of functions, by computing intermediary derivatives and multiplying them to obtain the target derivative.

2.3.2 Recurrent Neural Networks (RNNs)

With the need to incorporate model dependencies between tokens in the input sequence, Recurrent Neural Networks (RNN) were soon widely used in NLP (Goldberg, 2016). Unlike Feed-Forward Networks, which assume independence of the input tokens, RNNs can be conceptualised as Hidden Markov Models. As an example, considering a sequence $\mathbf{x}$ with T tokens, the RNN produces the hidden representations of the input in a recurrent manner, where the process at time step i in a single cell is:

$$\mathbf{h}_i = \mathbf{W}\mathbf{x}_i + \mathbf{U}\mathbf{h}_{i-1} + b \tag{2.8}$$

where $\mathbf{h}_i$ is the contextualised representation at time step i, $\mathbf{W}$, $\mathbf{U}$ are the parameters of the model, $\mathbf{h}_{i-1}$ is the hidden representation of the previous time step and b is the bias. Therefore each representation at time step t carries information from the previous token and is as such contextualised. It can therefore be well conceptualised as a Hidden Markov model where under ideal conditions the representation at time step i is dependent on the history up to the previous time step i - *1*.

LSTM: Standard RNN's have a fundamental flaw. In contrast to feed-forward networks which do not consider dependencies between tokens, the recursive nature of the gradient calculation in the backward step during training, causes the gradients to vanish due to increases in the multiplicative interactions in the chain rule (Hochreiter and Schmidhuber, 1997) (we provide details about the training and optimisation of neural networks in Section 2.3.6).

To address this issue, Hochreiter and Schmidhuber (1997) proposed Long-term Short Term Memory (LSTM) Networks, which at a single LSTM cell the operations are described

as:

$$
\begin{aligned}
\mathbf{f}_i &= \sigma_g(\mathbf{W}_f[\mathbf{h}_{i-1}, \mathbf{x}_i] + b_f) \\
\mathbf{i}_i &= \sigma_g(\mathbf{W}_i[\mathbf{h}_{i-1}, \mathbf{x}_i] + b_i) \\
\mathbf{o}_i &= \sigma_g(\mathbf{W}_o[\mathbf{h}_{i-1}, \mathbf{x}_i] + b_o) \\
\hat{\mathbf{c}}_i &= tanh(\mathbf{W}_c[\mathbf{h}_{i-1}, \mathbf{x}_i] + b_c) \\
\mathbf{c}_i &= \mathbf{f}_i \circ \mathbf{c}_{t-1} + \mathbf{i}_i \circ \hat{\mathbf{c}}_t \\
\mathbf{h}_i &= \mathbf{o}_i \circ tanh(\mathbf{c}_i)
\end{aligned}
\tag{2.9}
$$

where **f, i, o**, $\hat{\boldsymbol{c}}$ represent the forget, input, output and cell gates of the LSTM. The process starts with the forget gate $\boldsymbol{f}$, which controls how much information is passed on from the previous state. The input gate controls how much information to keep at the current state and the output gate the information relevant to the hidden representation. The strength of LSTM's considering gradients lies in the cell state, which carries and stores information going forward throughout the recurrency. During training, the element-wise multiplication and addition in the cell state "modify" the chain rule from being only a product in the case of RNNs, to products of additive interactions. This results into a stabilised gradient and subsequently yields better long-range dependencies than RNNs, but at the cost of an increased number of parameters and complexity.

GRU: Cho et al. (2014) proposed the Gated-Recurrent Units (GRU), an architecture that simplifies the LSTM using a smaller number of parameters. It is defined as follows:

$$
\begin{aligned}
\mathbf{z}_i &= \sigma_g(\mathbf{W}_z[\mathbf{h}_{i-1}, \mathbf{x}_i] + b_z) \\
\mathbf{r}_i &= \sigma_g(\mathbf{W}_r[\mathbf{h}_{i-1}, \mathbf{x}_i] + b_r) \\
\mathbf{n} &= tanh(\mathbf{W}_n[\mathbf{h}_{t-1} \circ \mathbf{r}, \mathbf{x}_i] + b_n) \\
\mathbf{h}_i &= (1 - \mathbf{z}_i) \circ \mathbf{n} + \mathbf{z} \circ \mathbf{h}_{i-1}
\end{aligned}
\tag{2.10}
$$

where **z** is the update gate and **r** the reset gate. While having similar properties with LSTMs, the latter have been proven to handle better longer range dependencies compared to GRUs (Weiss et al., 2018).

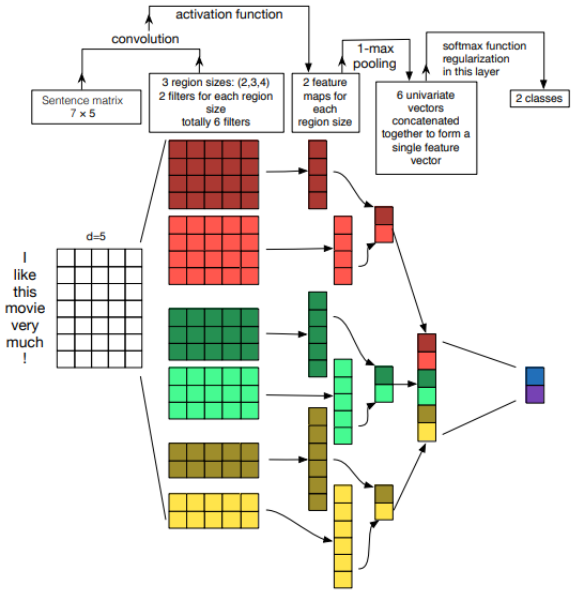


Figure 2.3: An illustration of a CNN pipeline used for an NLP task (Zhang and Wallace, 2017)

2.3.3 Convolutional Neural Networks (CNNs)

Convolutional neural networks (CNNs) were first introduced in the 1960's and function similarly to MLP's (LeCun et al., 1999). The core difference is the replacement of the matrix multiplications, of the previously mentioned neural architectures, with mathematical operations called convolutions. A convolution operates on two functions that produce an expression on how the functions affect each other. It is easier to understand the workings of CNNs in a computer vision task, where a filter or kernel is passed through a matrix of values or pixels (image) to produce a value. The kernel $\mathbf{W} \in \mathrm{R}^{\mathrm{k}\times\mathrm{l}}$ represents the learnable parameters of a CNN, where k and l are the height and width of the kernel. In NLP tasks, the sequence is represented as a matrix $\mathbf{S}$ with height T representing the sequence length and width d representing the embedding dimension. Since each row in $\mathbf{S}$ represents a distinct token, usually $\mathbf{W}$ has width $l = d$ (Zhang and Wallace, 2017). The convolutional operation can be described as:

$$o_i = \mathbf{W} \cdot A[i : i + k - 1] \tag{2.11}$$

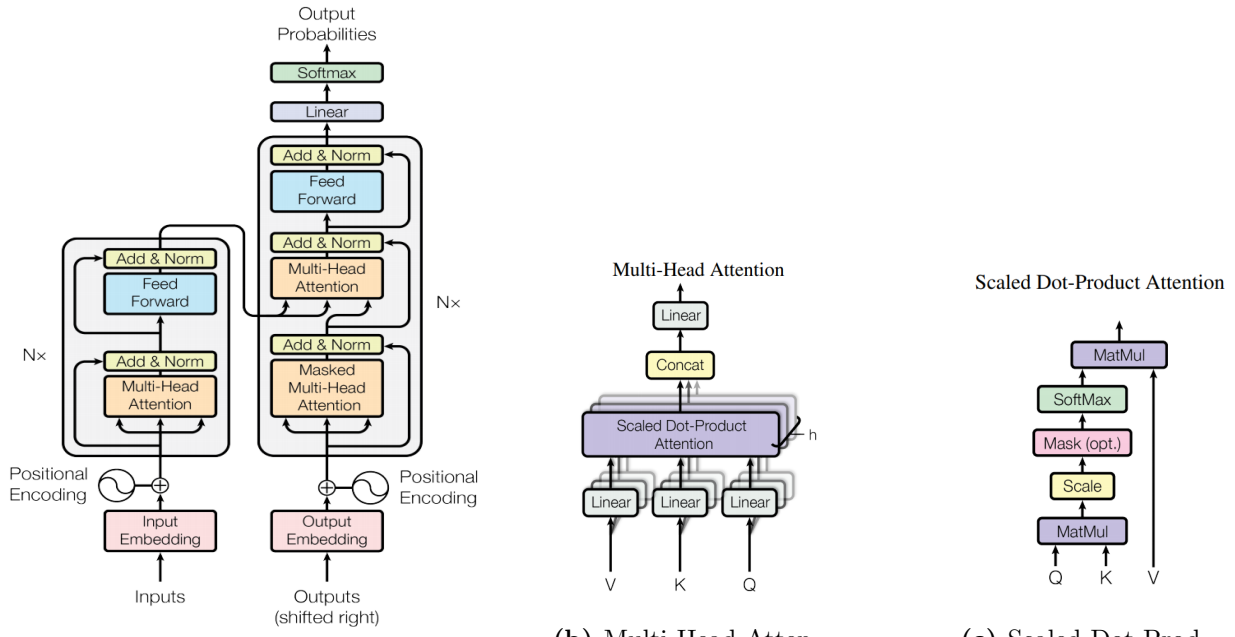
where $A[i : i+k-1]$ represents the sub-matrix the kernel covers at each stride and $\cdot$ is the dot

product (Zhang and Wallace, 2017). Multiple filters can be used for the same or different sized regions to learn complementary features. To produce a feature map c_i, or a represenation of the sequence, usually an activation function f is applied to the output sequence $\mathbf{o} \in \mathrm{R}^{\mathrm{s-k+1}}$, where $c_i = f(o_i)$. The second distinct feature of CNN's is pooling. Pooling works by reducing the dimensionality of the feature map, with the two common approaches being average pooling and max pooling. Pooling proves to be an effective technique in computer vision to allow the learned features to be translation and positional invariant. In a computer vision task this means that irrespective of the position of the features in an image, the CNN will be capable of detecting them due to the pooling technique. The pooling layer works much like the filter with a fixed width and height passing over the feature map and selecting either the average in the region of the filter or the max value. Figure 2.3 represents an illustration of a CNN applied to a binary text classification task.

2.3.4 Transformer

Vaswani et al. (2017) introduced the Transformer, a non-recurrent model architecture based on the attention mechanism, mainly for machine translation and sequence-to-sequence tasks. In recurrent encoders, each contextual representation h_i is mostly influenced from the surrounding context (Bahdanau et al., 2015). Though desirable for grammatical understanding, this reduces the ability of associating relevance with parts of the sequence that are further apart which the transformer architecture is capable of capturing.

The transformer architecture can be viewed in Figure 2.4a, whereby the component on the left can be described as an encoder and the component on the right as the decoder. The encoder consists of six stacked layers of multi-head attention an MLP . The multi-head attention seen in Figures 2.4b and 2.4c, is a key component in this architecture, where it computes the relevance of certain values $\mathbf{V}$, based on keys $\mathbf{K}$ and queries $\mathbf{Q}$. A single head attention is based on the scaled-dot attention mechanism described in Section 2.2. Using a single attention on the same sets of values, keys and queries would result in capturing only a single aspect of the input, thus the transformer uses multi-head attention, where each attention head focuses on different aspects of the sequence. The resulting similarity matrices from the attention process are concatenated and mapped through an MLP, to compute relationships between the different aspects.



(a) The Transformer architecture

(b) Multi-Head Attention

(c) Scaled-Dot Product Attention

Figure 2.4: The transformer architecture with a detailed view of Multi-head Attention and Scaled-Dot product attention

2.3.5 Pre-trained Transformer-based Language Models

The inception of the transformer architecture, paved the way for the development of models which would revolutionise the NLP landscape and dominate popular language understanding benchmarks (Wang et al., 2019). The most widely used model is BERT (Devlin et al., 2019), where its encoder is comprised of multiple layers of transformer networks. Crucial to its success is the learning component employed, the Masked Language Model (MLM) objective. MLM tasks a model in predicting all tokens in a sequence that have been masked ([MASK]) given their surrounding context. Devlin et al. (2019) adapted the transformer for learning text represenations, by pre-training on large corpora using the MLM as a primary objective. This pre-training results in the model learning rich text-representations due to the large number of training iterations (i.e. epochs) and size of the corpora it was trained on. Following pre-training, the model can be "fine-tuned" on a supervised end-task (e.g. text classification) by using only a handful of epochs. Soon after the introduction of BERT other variants were introduced, with most of them varying the pre-training objective (Lan et al., 2020; Sanh et al., 2020; Clark et al., 2020; Liu et al., 2019a; Yamaguchi et al., 2021) or by changing the

pre-training corpus to be domain specific (Beltagy et al., 2019; Chalkidis et al., 2020b).

2.3.6 Training Neural Networks

An ideal neural network would be initialised with a set of weights, that would yield for each case the appropriate prediction (Egmont-Petersen et al., 1994). However such weights cannot be known a priori. This is where the training, or learning phase, is applied to adapt the parameters of the model to produce predictions close to some target outputs. The training of neural networks often follows the same pipeline across architectures in supervised learning. First, the data is split into subsets for training, development and testing. The training subset is used for training the model with the development subset used for evaluating the predictive performance on unseen data, to find which set of parameters are most suitable for the model and task at hand. The testing subset is used for monitoring the model's predictive performance on unseen data. During training a prediction is produced by passing the input through the model, in what is often called a forward pass. The error is then computed between the prediction and the actual expected output.

Loss functions: The function used to compute the error, also referred to as loss, often varies with the selection of the metric heavily relying on the task at hand (Egmont-Petersen et al., 1994). A common metric used for regression tasks is the Mean-Absolute Error , which can be defined as:

$$\mathcal{L}(\hat{y}, y) = |\hat{y} - y| \tag{2.12}$$

where $\hat{y}$ and y are the models prediction and actual output respectively. This is a naive metric used for simple models, which was later adapted to the Mean-Squared Error (MSE). MSE simply squares the error shown in equation 2.12. This essentially penalises the model more when the error values are large. For classification tasks a commonly used error metric is Cross-Entropy Loss, which is described as:

$$\mathcal{L}(\hat{y}, y) = -\sum \hat{y} \log y \tag{2.13}$$

Cross-Entropy Loss penalises the model not only when it makes a false prediction, but penalises highly when it makes a false prediction highly. Another common metric used for classification is Negative Log-Likelihood Loss which is formed from the negative log value of the expected output, $loss(\hat{y}, y) = -\log y$.

Computing the error is followed by the learning phase, where the parameters are adjusted to reduce the error or loss. This is an optimisation problem and the first step to optimising is calculating the contribution of each parameter to the output error (LeCun et al., 1989).

Backpropagation: The contribution of each of the parameters can be calculated by backpropagation (LeCun et al., 1989). Backpropagation was first tested on feed forward neural networks and has since then become the predominant approach for training neural models (Werbos, 1974). Given that the functions used by a neural network to produce a prediction are differentiable, the chain rule is used to compute the derivatives of the error, computed from the loss functions, with respect to the network's parameters. The chain rule allows us to obtain the derivative of the composition of functions. As an example consider function $f(x) = g(z(x))$, where we assume that g is an activation function and $z(x)$ a linear transformation function. The chain rule allows us to compute effectively the derivatives $f'(x)$ by computing the intermediary derivatives such that $f'(x) = f'(g) \cdot g'(x)$. This allows for optimisation algorithms, described further on, to be used as learning processes to adjust the parameters with the most prominent one being stochastic gradient descent.

In the case of RNN's though the time dependency requires a slight variant of backpropagation, called Backpropagation-through time (BTT) (Werbos, 1990). This variant modifies the standard backpropagation approach to unfolds the equations through the time steps and assigns the propagated the error through each time step (LeCun et al., 1989). This results in an elongated chain of nodes equal to the time steps.

Optimisation: Following the computation of the gradients, and thus the contribution of each parameter, there are several commonly used optimisation algorithms to adapt the parameters. The basis for most optimisation algorithms in neural networks is Gradient Descent (Lemaréchal, 2012). Batch Gradient Descent (BGD) is performed on the entire dataset and is guaranteed to converge to the global minimum for convex error surfaces and

to a local-minimum for non-convex (Ruder, 2016). However it can be very time consuming for large datasets and is rarely used in literature, thus will not be described in this report. Stochastic Gradient Descent (SGD) is a variant of BGD, and in contrast to BGD performs an update for each instance by using the following rule:

$$\theta = \theta - \eta \cdot \nabla loss(\theta; x^i; y^i) \tag{2.14}$$

where θ represents the model parameters, η the learning rate and $\nabla J(\theta; x^i; y^i)$ the gradient of the loss. Compared to BGD, the fact that the update is performed for each point allows the learning process to progress much faster and allows for finding better local minima, but complicates convergence as it can also overshoot and not converge. By taking advantage of the benefits of, Mini-batch Gradient descent utilises the same principles seen in equation 2.14 but performs updates on several instances at once (Ruder, 2016). There are several variants and add-ons to the SGD to aid convergence with the most prominent ones varying the learning rate according to the learning process.

Various modifications have been introduced to stochastic gradient descent for better convergence. Such a variant is Adagrad, which modifies the learning rate η at each time by performing larger updates for infrequent parameters and smaller updates for frequent ones making it suitable for sparse data (Ruder, 2016). Adaptive Moment Estimation (Adam) also dynamically adjusts the learning rate η by storing an exponentially decaying estimates of the mean m_t and variance v_t of past gradients (Ruder, 2016). The Adam update rule is then computed by:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{v_t} + \epsilon} m_t \tag{2.15}$$

Regularisation: A common problem of neural models during training is overfitting. Overfitting is caused when a model can predict accurately the instances within the domain of the training data but performs badly on unseen data. To avoid this, as mentioned in this section previously, a development subset of the data is used to monitor performance and decide where to stop training. Additionally there are several techniques to regularise the network parameters and prevent overfitting by introducing a priori on the error function. The most common approaches in literature are l_1 and l_2 regularisations or weight decay (Mc Loone and Irwin, 2001) and dropout. They aim to penalising the generalisation error

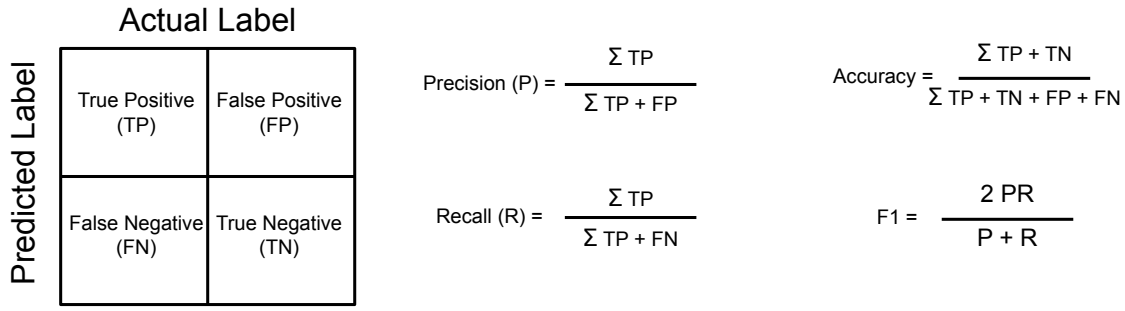


Figure 2.5: Popular metrics for evaluating model predictive performance.

but not the training error, by adding a term on the loss function. l_1 regularisation includes adding the sum of all model parameters, l_2 regularisation adds the sum of all squared model parameters. Usually the two terms are multiplied by a scalar quantity λ to adjust the level of regularisation. Finally, dropout controls overfitting by randomly omitting subsets of features at each training iteration (Hinton et al., 2012).

Evaluating Model Predictive Performance: Having trained a model on an end-task, we then need to evaluate the model's predictive performance on unseen instances during training. Two popular metrics used to evaluate predictive performance are Accuracy and the F1 score, where a higher score indicates better predictive performance. Accuracy, represents the ratio of the correctly predicted labels against the actual given labels. The F1 score represents the harmonic mean between presicion and recall. Precision is the ratio of true positives (correctly classified instances of target class) against the sum of true positives and true negatives (instances correctly not classified as target class). Similarly, recall is computed using the true positive ratio against the true positives and false negatives (instances wrongly classified as not the target class). We show these metrics in Figure 2.5. For multi-class tasks (i.e. not a binary task), we can compute the F1-macro or F1-micro for the overall performance. F1-macro computes the average of the F1 score of all classes. In contrast, F1-micro computes the weighted average (weighted by the number instances found under each class) of the F1 score of all classes.

Whilst neural models have been gaining ground in most NLP applications in contrast with linear classifiers, they carry several shortfalls. Larger neural architectures require more data, time and computational power to train in comparison to simpler classifiers and more

importantly they are more complex to understand. Compared to linear models, which offer intuitive approaches to explain a specific prediction, the interconnected grid of weights, non-linear activation functions and stacks of layers of neural models makes interpreting their decisions notoriously difficult (Weiss et al., 2018). Their enlarged complexity and highly parametrised architectures makes them hard to interpret, as such earning them the title of 'black boxes' (Zhang et al., 2018). Having provided pre-requisite knowledge in the previous sections, the remainder of this chapter will first motivate why model interpretability is important and discuss the challenges faced by previous studies.

2.4 Model Interpretability in NLP

As more 'black box' models penetrate safety-critical tasks, extracting explanations to understand model predictions is increasingly becoming more important in NLP. Doshi-Velez and Kim (2017) argue that model interpretability is of crucial importance for the following reasons:

- To gain scientific understanding, by obtaining explanations that can be converted into knowledge.
- To increase safety in scenarios where a prediction will lead to action in a high-risk environment, by providing explanations for the prediction.
- To improve the 'fairness' of algorithmic predictions thus reducing or ideally eliminating biases. For example when a hiring algorithm rejects a candidate based on his resume, the hiring agents need to make sure that this decision was not influenced by 'unethical' reasoning.
- To provide explanations and accountability, as required by law, for predictions affecting actions at a user-level (Goodman and Flaxman, 2017).

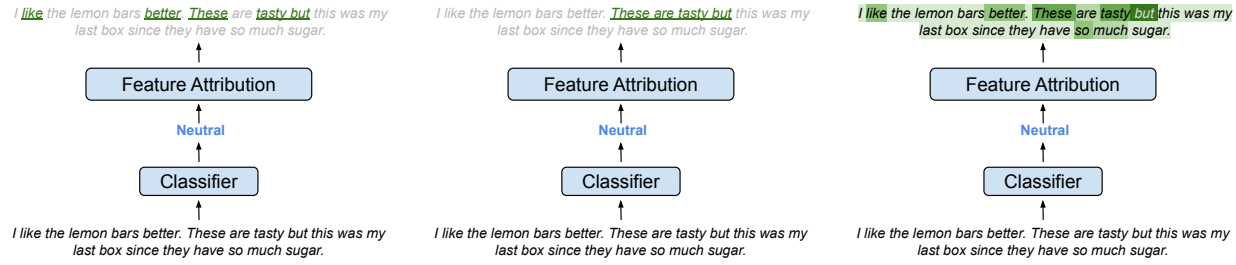
Defining Model Interpretability: Model interpretability as a concept has received increased attention over recent years. This has subsequently led to an evolving definition of what is model interpretability, with multiple possible directions for research forming under

this umbrella term. Several studies have focused on examining the linguistic information learned by these models (Brunner et al., 2020; Pascual et al., 2020). Other studies have focused on generating explanations or rationales for model predictions, by identifying the subset of the input that is more important for a model's prediction (Lei et al., 2016; Jain et al., 2020; Treviso and Martins, 2020). Whilst understanding the learning capabilities of models is important, this project focuses on the latter, which is concerned about identifying which parts of the input contributed the most towards a model's prediction.

Lipton (2016) and Miller (2019) define interpretability as the degree to which an observer can understand the cause of a decision, whilst an explanation is defined as "a mode in which an observer may obtain understanding". Miller (2019) equate *explainability* with *interpretability*, whilst specifying that *justification* explains why a decision is good, but does not aim to explain the actual process that achieved a prediction. Doshi-Velez and Kim (2017) define interpretability as the ability to explain or present in understandable terms to a human, however they argue that there is not a formal definition for *explanation* in machine learning and thus adopted a definition from the field of psychology. Whilst some researchers use interpretations and explanations to increase trust in their models (Miller, 2019; Ribeiro et al., 2016), Lipton (2016) argues that interpretability as a term is ill-defined. This claim is supported by the large disparity between the definitions given in literature, which clearly shows a lack of a formal framework on what constitutes a good explanation or interpretation.

Jacovi and Goldberg (2020) helped accelerate research in this area, by framing two crucial concepts or 'properties' of what an explanation or rationale should be. These 'properties' or categorisations are *plausibility* and *faithfulness*. A plausible explanation is one which is intuitive to a human reader, whilst a faithful explanation is one which is truly reflective of the reasoning behind a model's prediction. An explanation can be both plausible and faithful or either, as these two properties have been previously shown to not correlate (Atanasova et al., 2020). Each property has its own importance under different settings. For example, a plausible rationale might be more desirable where we need to increase the end-user's trust in a model's prediction. On the other hand, a faithful rationale is essential in safety-critical settings, where a practitioner needs to understand the true reasoning behind a model's prediction to avoid catastrophic decision making.

As the motivation of this work is to improve understanding for model predictions, particularly in safety-critical settings, this thesis focuses on improving and evaluating the *faithfulness*



(a) Top-k Rationale **(b)** Contiguous Rationale **(c)** Heat-map

Figure 2.6: Examples of the different types of post-hoc explanations, with (a) showing the k highest scored tokens (Top-k); (b) showing the highest score contiguous subset (Contiguous) and (c) the explanation as a heat-map.

of explanations. The formal definition that is adopted therefore is: *"A faithful explanation or rationale is one that is representative of the true reasoning behind a model's prediction (Jacovi and Goldberg, 2020), regardless of if the explanation is plausible or not (Atanasova et al., 2020)".*

2.5 Rationale Extraction

Given a model $\mathcal{M}$ trained on an end task, we are interested in explaining why $\mathcal{M}$ predicted $\hat{y}$ for a particular instance $\mathbf{x} \in \mathbf{X}$. An extracted faithful rationale $\mathcal{R}$, should therefore represent as accurately as possible the most important subset of the input ($\mathcal{R} \in \mathbf{x}$) which contributed mostly towards the model's prediction $\hat{y}$.

Currently, there are two popular approaches for extracting rationales. The first approach consists of first identifying the most important input tokens, by using feature attribution methods (Ω). Feature attribution methods are approaches which allocate importance scores ($\boldsymbol{\omega}$) to the input tokens with respect to a prediction.[1] We can then form a rationale by selecting the k most important tokens (top-k rationale), or by selecting the highest scored contiguous subset of length k (contiguous rationale). Alternatively, another popular use of feature attribution approaches, is to visualise the allocated importance over the input sequence as a heat-map (Barbieri et al., 2018). As this approach relies on using a model that

[1]Feature attribution methods are also referred to as feature scoring methods.

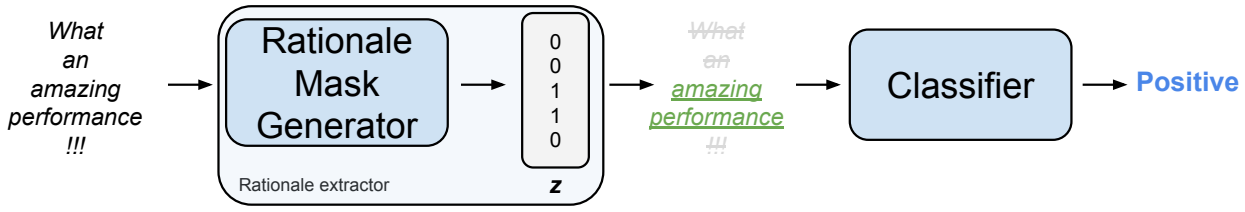


Figure 2.7: A typical pipeline followed by an inherently faithful select-then-predict model.

is already trained on an end-task, these explanations are typically referred to as post-hoc explanations. For clarity, we illustrate types of post-hoc explanations in Figure 2.6.

The second approach for extracting rationales, focuses on forming inherently faithful classifiers by jointly training two modules, a *rationale generator* and a *rationale classifier* (Lei et al., 2016; Bastings et al., 2019). The rationale generator first generates a rationale mask that can be applied over the input. The classifier is then trained on an end-task using only the extracted rationales. As the classifier is trained only on the rationales, it is therefore inherently faithful. Recent studies have used feature attribution methods as part of the rationale generator (Jain et al., 2020; Treviso and Martins, 2020; Guerreiro and Martins, 2021), showing improvements in classifier predictive performance. We visualise select-then-predict models in Figure 2.7.

For the remainder of this section, we first present popular feature attribution methods used for extracting post-hoc explanations and will then follow with popular implementations of inherently faithful select-then-predict models.

2.5.1 Feature Attribution Methods for Post-hoc Explanations

Given a pre-trained model $\mathcal{M}$ and an output distribution $\mathcal{Y}$ (where $\hat{y} \in \mathcal{Y}$), feature attribution approaches Ω extract importance scores $\boldsymbol{\omega}$ for an input sequence $\mathbf{x}$ such that:

$$\boldsymbol{\omega} = \Omega(\mathcal{M}, \mathbf{x}, \mathcal{Y}) \tag{2.16}$$

Feature attribution approaches can be categorised into different groups, based on the

framework they adopt. Through this work we will present feature attribution approaches, that are split into the following groups: (1) gradient-based; (2) perturbation-based; (3) simplification-based and finally (4) attention-based.

Gradient-based methods: Gradient-based methods rely on computing the gradients of a prediction with respect to the input as indicators of input importance. The foundation of gradient-based methods is the *sensitivity analysis* approach (Simonyan et al., 2013), which directly computes the gradients of the prediction with respect to the input such that $\frac{\partial \hat{y}}{\partial \mathbf{x}}$. Since the input to a neural network are token representations (i.e. the embeddings), rather than the tokens themselves, there exist different formulations on how to map the embedding gradients to a single token-wise importance score (DeYoung et al., 2020). Popular formulations include computing the average gradient score for each word representation, summing all gradient scores, computing their l_1 norm or their l_2 norm. It has been shown that preserving large gradient values by performing aggregation (e.g. l_1 or l_2 norm) rather than averaging, results in more faithful allocation of input importance (Atanasova et al., 2020).

Several other variants have been developed with the aim of obtaining a more accurate allocation of input importance. Such an approach is *InputXGradient*, which uses the *sensitivity analysis* method to generate the gradients to then multiply them with the input (Kindermans et al., 2016). *Layer-wise Relevance Propagation* (LRP), uses the target class (predicted class) and decomposes the function that produces the target class prediction. The decomposition is performed by back-propagating through the network using a set of rules and allocates relevance to the neurons of each layer recursively until it reaches the input layer. This approach is based on the principle that relevance is conserved across all layers of the network. Another popular approach, *Integrated-Gradients* (IG) allocates input importance, 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 (Sundararajan et al., 2017). In a similar direction, *DeepLift* computes the difference between the activation of each neuron and a reference activation (zero embedding vector) to allocate input importance (Shrikumar et al., 2017).

Perturbation-based methods: The intuition behind this class of feature scoring methods, is that we can compute the importance of a single token on the prediction by ei-

ther removing or replacing it. A popular example of such a feature attribution method is *Word Omission*, which allocates input importance by computing the difference between the probabilities of the predicted class when including a word i and omitting it: $\text{WO}_i = p(\hat{y}|\mathbf{x}) - p(\hat{y}|\mathbf{x}_{\setminus x_i})$ (Robnik-Šikonja and Kononenko, 2008; Nguyen, 2018a). Typically, the tokens are omitted by replacing them with a zero embedding vector. Kim et al. (2020) suggest an alternative formulation to occlusion style approaches, which relies on input marginalisation. Instead of replacing the input token with a zero embedding vector, Kim et al. (2020) suggest replacing the target input token with the most likely candidate to replace it. In the case of transformer-based models like BERT, this is easily achievable as they are pre-trained using a masked language objective and as such the Masked Language Model (MLM) can be used. They then suggest marginalising over the likelihoods of candidate tokens in place of the target token, to reach an attribution score for the particular token.

Simplification-based methods: These approaches are inspired by using simpler, explainable models to approximate the predictions of the original 'black box' model. A prominent simplification method is Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al., 2016). LIME generates explanations by perturbing parts of the input and then observing how the output is affected. LIME essentially generates these explanations by training a linear model to describe the underlying more complex model, based on the aforementioned perturbations. A key benefit of this method is that it is model-agnostic and thus it can provide explanations for the predictions of any pre-trained model.

Attention-based methods: Derived from attention mechanisms (described in Section 2.2), *attention weights* (α) from a model trained on an end-task have been implicitly or explicitly used as indicators of input importance (Cho et al., 2014; Xu et al., 2015; Barbieri et al., 2018; Ghaeini et al., 2018). In an attempt to improve them, other variants emerged. Such a variant that is not frequently used are the *attention gradients* (Serrano and Smith, 2019), which essentially use the gradients of the prediction with respect to the attention weights as indicators of input importance. Building on that, *scaled attention* multiplies the *attention weights* with their corresponding *attention gradients*. The intuition is that attention weights offer an output-generic representation of importance. As such, scaling them by their gradients to the prediction results in class-specific importance scores.

2.5.2 Select-then-Predict Models

This section describes popular variants of select-then-predict, inherently faithful models. For simplicity this work will describe the implementations of different variants, in terms of the processes they follow which can be depicted in Figure 2.7. Generally, select-then-predict models are comprised of two modules, a rationale extractor and a rationale encoder followed by a classification layer, which typically operate together.[2] The rationale extractor accepts as input an entire text sequence and is responsible for extracting the most informative subset of the input sequence, i.e. the rationale. The input sequence is first fed into the rationale mask generator, which generates a rationale mask $\mathbf{z}$ which is conditioned on the input, such that $p(\mathbf{z}|\mathbf{x})$. The rationale mask indicates which tokens should be extracted and which should be omitted. The preserved input (i.e. the rationale) is then passed through the classifier to reach a prediction. Depending on the implementation, these two modules can be either trained jointly (Lei et al., 2016; Bastings et al., 2019; Treviso and Martins, 2020) or independently (Jain et al., 2020; Treviso and Martins, 2020).

Bernoulli-Latent-Model: The work by Lei et al. (2016), is one of the first popular implementations of inherently faithful select-then-predict models in NLP. To gain a better understanding of the processes underlying select-then-predict models, we describe this model in detail. Their proposed rationale mask generator ($\mathbf{gen}(\mathbf{x}; \phi)$), is tasked with predicting a sequence of t Bernoulli parameters given an input $\mathbf{x}$, where t is the number of tokens in the input sequence and ϕ the parameters of the generator model. The Bernoulli parameters are then used to generate a rationale mask $\mathbf{z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_T\}$, which acts as a binary gating mechanism that decides which tokens in the input sequence to preserve. We can therefore generate a rationale $\mathcal{R}$ by:

$$\begin{aligned} \mathbf{z} &\sim \mathit{Bernoulli}(\mathbf{gen}(\mathbf{x}; \phi)) \\ \mathcal{R} &= \mathbf{z} \odot \mathbf{x} \end{aligned} \tag{2.17}$$

[2]We refer the rationale encoder and the classification layer jointly as the rationale classifier for brevity.

The rationale $\mathcal{R}$ is then passed through a classifier to make a prediction, such that:

$$\hat{y} \sim Classifier(\mathbf{z} \odot \mathbf{x}; \theta) \tag{2.18}$$

where θ are the classifier parameters and during training, the generator and classifier are jointly learned. The learning challenge is two-fold. Firstly, the predicted outcome should be as close as possible to the actual outcome (i.e. the given labels). For example if we were to minimise the mean squared error:

$$\mathcal{L}(\mathbf{z}, \mathbf{x}, \mathbf{y}) = ||Classifier(\mathbf{z} \odot \mathbf{x}; \theta) - \mathbf{y}||_2^2 \tag{2.19}$$

Secondly, the generator is guided to select short and coherent rationales, such that:

$$\Omega(\mathbf{z}) = \lambda_1 ||\mathbf{z}|| + \lambda_2 \sum_{i=0}^{t} |z_i - z_{i-1}| \tag{2.20}$$

where the first term penalizes the number of selections and the second encourages the contiguity of selections. The final loss function is a combination of the two, where the overall expected loss $\mathcal{L}$ is minimised over both the generator and classifier such that:

$$\min_{\phi\ \theta} \sum_{i=1}^{n} \mathbb{E}_{z_i \sim gen(\mathbf{x}_i)} \mathcal{L}(Classifier(\mathbf{x}_i, \mathbf{z}_i), \mathbf{y}_i) \tag{2.21}$$

where n is the number of documents in the training corpus. The expected loss is difficult to optimise as it requires marginalising over all the possible rationale masks $\mathbf{z}$. Lei et al. (2016) perform parameter approximation by drawing samples from the generator and averaging their gradients during training. This works well for extracting rationales, however as a result of the high difficulty in exploring the space of all possible rationales, there is large variation in predictive performance across random seeds.

HardKuma: Bastings et al. (2019) build upon the work of Lei et al. (2016) to train a jointly the rationale mask generator and the classifier. They key difference in their work, is the

introduction of the Hard Kumaraswamy (HardKuma) distribution, which they use to sample the latent rationale mask $\mathbf{z}$. HardKuma allows them to draw samples using the reparametrisation trick, such that gradient estimation is possible without REINFORCE (Williams, 1992) as in the case of Bernoulli latent variables. REINFORCE algorithms allow for parameter adjustments by approximation, without explicitly computing gradient estimates. Bastings et al. (2019) also propose the use of Lagrangian relaxation for constraining the extracted rationale.

A significant drawback of stochastic rationale models is that they require marginalization over all possible rationales, which in practice is intractable (Guerreiro and Martins, 2021). For training, this results to either REINFORCE style gradient estimates, such as the Bernoulli-Latent-Model, or reparametrised gradients such as the case of HardKuma. As such, these models require careful hyperparameter tuning and they typically exhibit high variance over multiple runs (Jain et al., 2020).

FRESH: In a different direction, Jain et al. (2020) introduced FRESH. Unlike the two previously discussed variants, FRESH uses three modules: a *support model*, an *extractor* and a *classifier*. The *support model* and *extractor* can be considered as parts of the rationale extractor depicted in Figure 2.7. A crucial difference in FRESH, is that the support model and classifiers can be trained independently. More formally, the *support model* is trained using the full length sequences on the end-task, the *extractor* module extracts rationales by using a feature attribution approach and the *classifier* is trained on the rationales for an end-task. The fact that these modules are independent, allows for one to use an already trained support model and then specify an extractor which uses a pre-defined feature scoring method, sparsity level (i.e. length of the rationale) and rationale type (i.e. top-k or contiguous) to generate a rationale mask $\mathbf{z}$. Additionally it has been shown that FRESH does not require the extensive hyperparameter search that the Bernoulli Latent variables and HardKuma require.

In a similar direction to FRESH, Treviso and Martins (2020) propose using sparse attention as the feature scoring method to generate rationales. Sparse attention probability distributions are obtained by using the sparsemax (Martins and Astudillo, 2016) instead of the softmax over the attention weights. Sparsemax allows for assigning zero probability scores and as such the attention probability distribution is sparse. This arguably makes sparse attention easier to interpret than when using standard attention with softmax (see Section 2.2). They form a rationale mask $\mathbf{z}$ by selecting all tokens with non-zero attention

probability. A limitation of using sparse attention however and FRESH, is that the rationale sparsity and contiguity cannot be enforced during training.

2.6 Evaluating Explanation Faithfulness

In literature, the large majority of evaluation approaches for post-hoc explanation faithfulness rely on *erasure* approaches (Nguyen, 2018b; Serrano and Smith, 2019; Atanasova et al., 2020; DeYoung et al., 2020; Madsen et al., 2021a). Erasure approaches constitute of masking input tokens (i.e. by replacing certain tokens with an empty placeholder, such as [MASK]) and observing changes in the prediction. There are multiple variants and implementations of erasure approaches, which are described below:

Word Relevance: Arras et al. (2017) evaluate the faithfulness of explanations generated by LRP and sensitivity analysis, by recursively removing (i.e. by masking) words in order of increasing and decreasing importance. The intuition is, that by deleting important words in order of decreasing importance in correctly classified sentences, then we should observe performance degradation. This degradation should continue with increasing numbers of removed "important" words. Inversely, by removing tokens of increasing importance in falsely classified sentences, we should observe increases in performance. They argue that a higher degradation rate, in the case of removing tokens of decreasing importance, indicates a feature attribution method that can more faithfully allocate importance.

Fraction of Tokens: In a different direction to word relevance, which removes words based on their *true* class, Nguyen (2018b) suggest removing words based on their *predicted* class. They argue that this allows for an unsupervised evaluation of explanation faithfulness, when the true labels are not available. Additionally, this work considers this evaluation to be more closely aligned with the concept of faithfulness, as a faithful explanation is concerned with explaining what a model has predicted and not what it should have predicted. Fraction of tokens therefore, removes tokens in order of decreasing importance and records when a prediction switch has occurred (i.e. how many tokens are required relative to the length of the sequence to cause a prediction flip). The intuition is that a feature scoring method that

results in a low fraction of tokens score is more faithful, as it highlights correctly the most important tokens in a sequence (Serrano and Smith, 2019).

Comprehensiveness: DeYoung et al. (2020) suggest that for a rationale to be faithful, we should observe a drop in the probability score of the originally predicted class when removing it from the sequence. The comprehensiveness score therefore aims to quantify how necessary the rationale is for a model's predictions. This is achieved by measuring the change in a model's probability score between the original prediction probability score and the probability score with the rationale masked, such that:

$$\text{Comp}(\mathbf{x}, \hat{y}, \mathcal{R}) = p(\hat{y}|\mathbf{x}) - p(\hat{y}|\mathbf{x}_{\setminus\mathcal{R}}) \tag{2.22}$$

Carton et al. (2020) bind this metric between 0 and 1, but also suggest normalising comprehensiveness to account for the behaviour of the model when presented with a baseline input (i.e. an all zero embedding vector). They argue and empirically demonstrate that normalisation, paints a truer picture of rationale comprehensiveness and sufficiency when comparing across models and datasets. Normalised comprehensiveness (NormComp) is therefore defined as:

$$\text{NormComp}(\mathbf{x}, \hat{y}, \mathcal{R}) = \frac{\text{Comp}(\mathbf{x}, \hat{y}, \mathcal{R})}{\text{Comp}(\mathbf{x}, \hat{y}, 0)} \tag{2.23}$$

where $\text{Comp}(\mathbf{x}, \hat{y}, 0)$ is the comprehensiveness of an all zero embedding vector.

Sufficiency: A sufficient rationale, is one which is adequate for a model to make a prediction (DeYoung et al., 2020). To measure sufficiency, DeYoung et al. (2020) measure the difference between the predicted class probability of a model using the full input against using only the rationale, similar to comprehensiveness. The intuition is that if a rationale is sufficient, there should be a low difference in the predicted class probability:

$$\text{Suff}(\mathbf{x}, \hat{y}, \mathcal{R}) = p(\hat{y}|\mathbf{x}) - p(\hat{y}|\mathcal{R}) \tag{2.24}$$

Carton et al. (2020) bind again this metric between 0 and 1. Similar to comprehensiveness, they again normalise sufficiency (NormSuff):

$$\text{NormSuff}(\mathbf{x}, \hat{y}, \mathcal{R}) = \frac{\text{Suff}(\mathbf{x}, \hat{y}, \mathcal{R}) - \text{Suff}(\mathbf{x}, \hat{y}, 0)}{1 - \text{Suff}(\mathbf{x}, \hat{y}, 0)} \tag{2.25}$$

AOPC: As sufficiency and comprehensiveness evaluate rationales at a specified length, DeYoung et al. (2020) suggest capturing these metrics under different rationale lengths by computing the Area over the Perturbation Curve (AOPC).[3] To achieve that, they construct bins designating the number of tokens to be deleted and aggregate their importance. For comprehensiveness (normalised or not) this is defined as:

$$AOPC = \frac{1}{|\mathcal{B}|} \sum_{b=0}^{\mathcal{B}} p(\hat{y}|\mathbf{x}) - p(\hat{y}|\mathbf{x}_{\setminus\mathcal{R}}) \tag{2.26}$$

where $\mathcal{B}$ is the set of pre-defined bins, e.g. $\mathcal{B} = \{10\%, 20\%, ..\}$. The same formulation can be applied for sufficiency analogously.

F1 macro: Similar to comprehensiveness we can measure the drops in F1 macro performance of model $\mathcal{M}$ when masking the rationale in the original input ($\mathbf{x}_{\setminus\mathcal{R}}$) (Arras et al., 2017). Larger drops in F1 scores indicate that the extracted rationale is more faithful. Arras et al. (2017) measured this using the dataset gold-labels, however for evaluating faithfulness the F1 macro performance can be evaluated using the predicted labels of a model using the full text.

ROAR: Remove-and-Retrain (ROAR) is based on the principle that a rationale is important, then removing it from the input and retraining the model should result in degraded performance Hooker et al. (2019). As ROAR was originally conceptualised for computer vision tasks, Madsen et al. (2021a) adapted it for NLP and proposed an improvement, Recursive ROAR (Rec-ROAR). Rather than assuming a fixed length rationale, similar to AOPC

[3] Carton et al. (2020) also use the term *fidelity* when jointly referring to sufficiency and comprehensiveness.

ROAR recursively removes tokens (or bins of tokens) in order of decreasing importance. At each iteration the model is re-trained and the updated performance is recorded. As with Word-Relevance, we expect that the most faithful feature attribution method is the one which results in the highest performance degradation at each recursion.

The evaluation approaches described previously are applicable for post-hoc explanations. Select-then-predict models are inherently faithful and as such there is no need to evaluate the faithfulness of their rationales. However, a good measure of their explanation quality or informativeness, is the predictive performance of the classifier which is trained only on the rationales. When comparing across the same classifier architecture, increases in performance indicate that the rationales contain more information relevant to the task, as the model is able to learn better. Overall, recently developed select-then-predict models exhibit higher and more stable predictive performance across multiple runs and tasks, with a reduced need for extensive hyperparameter tuning (Jain et al., 2020; Treviso and Martins, 2020; Guerreiro and Martins, 2021) compared to work in this area (Lei et al., 2016; Bastings et al., 2019).

2.6.1 Faithfulness of Post-hoc Explanations

Due to the rapid developments in the area of model interpretability, there has not always been a clear framework of what constitutes a good explanation or rationale (Jacovi and Goldberg, 2020). This has impacted the evaluation of post-hoc explanations in literature, with previous research showing conflicting outcomes. For this purpose, this section describes in a chronological order how the faithfulness of post-hoc explanations was studied in previous work, describing those relevant to the context of this project. This past-to-present analysis, will help form a clearer understanding of what is the current state and limitations when extracting faithful post-hoc explanations.

Nguyen (2018b) have compared the faithfulness of three feature scoring methods, namely LIME, the sensitivity analysis and Word Omission. They used two text classification datasets to perform their analysis and their evaluation was largely based on computing the AOPC comprehensiveness (non-normalised) of the generated post-hoc explanations. Their results have shown that for Logistic Regression, Word Omission outperformed LIME regardless of the number of samples used in it. Similarly, for an MLP model simple gradients performed better followed by Word Omission and finally LIME. It is important to note that with an

increasing number of samples, LIME becomes more computationally expensive. As such, their research shows that LIME rarely justifies the computational overhead compared to the faithfulness of its generated explanations. Nguyen (2018b) have also shown that outcomes from the AOPC comprehensiveness evaluation also hold, when evaluating using the Fraction of Tokens approach. Arras et al. (2017) used Word Relevance to compare the efficacy of LRP against the sensitivity analysis approach. They show that there is a larger degradation in model predictive performance by removing important tokens when indicated by LRP compared to sensitivity analysis. However, Arras et al. (2017) use the true labels of the model rather than the predicted to evaluate post-hoc explanations, an important limitation towards evaluating explanation faithfulness. This is not representative of explanation faithfulness, as a faithful explanation aims to show why a model made a certain prediction, despite of if the model is right or wrong.

Atanasova et al. (2020) have examined post-hoc explanations against a wide range of properties, including faithfulness. Their study, one of the more extensive on evaluating faithfulness, includes four gradient-based feature scoring methods, two perturbation based and one simplification approach (LIME). The gradient-based include the simple gradients approach, two variants of InputXGrad and Gradient Backpropagation, a variant of LRP. Additionally, for the gradient-based methods two aggregation approaches were tested (i.e. from embedding level importance to token level importance). First, averaging across the embedding dimension and secondly computing the l_2 norm. For the perturbation based approaches, Atanasova et al. (2020) opted for Word Omission and Shapley Value Sampling (ShapSampl) (Castro et al., 2009). ShapSampl is based on the Shapley Values approach, that computes the average contribution of each word across all possible word perturbations. The Sampling variant considers only a fixed number of random perturbations as opposed to all possible perturbations and thus is a faster approximation of Shapley Values.

The findings of Atanasova et al. (2020) shed light into some problematic behaviour of feature scoring methods. Firstly, it was shown that the faithfulness of a feature scoring method depends on the task and encoder. Faithfulness was evaluated by measuring comprehensiveness under different rationale lengths. Encoder-wise, they show that post-hoc explanations were more faithful when using simpler encoder architectures (such as CNNs) when compared to transformer-based models (independent of the feature scoring method). This is expected, as the less complex a model, the better the explanations are (Jain and Wallace, 2019; Wiegr-

effe and Pinter, 2019). Overall, it was shown that InputXGradient, a gradient-based variant with l2 aggregation, performed more consistently than all other feature scoring methods and resulted in more faithful explanations. However, the performance of the remainder of feature scoring methods is inconsistent and depends on the dataset and encoder. Finally, it was demonstrated empirically that the performance of feature scoring methods not only varies across datasets, but also within the same dataset across instances. These findings are largely in agreement with the suggestions of Jacovi and Goldberg (2020), who support this argument and also suggest evaluating faithfulness at instance-level.

The case of attention: A prominent feature scoring method that has faced criticism for the quality of its explanations are attention mechanisms (Jain and Wallace, 2019; Serrano and Smith, 2019; Wiegreffe and Pinter, 2019). A common practice was 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; Sun and Lu, 2020) as a mean towards model interpretability (Lipton, 2016). Barbieri et al. (2018) for example, use attention heat-maps to qualitatively show that the attention weights from their proposed, multi-label attention mechanism allocate importance more closely to a particular label.

However, their efficacy in producing explanations has not been assessed until recently. Jain and Wallace (2019) were the first to address the faithfulness of post-hoc explanations from attention weights. Their analysis was two-fold; for attention weights to provide explanations: (1) they should correlate with other feature attribution methods and (2) adversarial attention distributions that diverge from the original (i.e. that show alternative input importance to the original attention distribution), should yield significant changes in the prediction. For their first analysis they correlated the importance rankings of attention weights with the simple gradients method and Word Omission. They found that with recurrent encoders, attention weights exhibited low correlation to either of the other two feature attribution methods and suggested that attention weights cannot produce explanations. Regarding their adversary analysis, they first kept a pre-trained model fixed and scrambled the attention weight distribution. They found that there were many alternative attention distributions that yielded comparable prediction distributions. However, their evaluation has received criticism itself.

Wiegreffe and Pinter (2019) argued against some of the experiments conducted by Jain and Wallace (2019), showing that in certain cases attention mechanisms can produce faithful explanations. Their work is based on the argument that the adversary experiments were not structured and it was not clear what property of explanations was evaluated. As such, they assessed how useful to a model the attention mechanism as a proxy for evaluating their explanation quality. Serrano and Smith (2019) conducted a concurrent but independent study to Jain and Wallace (2019), to assess the faithfulness of attention-based post-hoc explanations. For their experiments, they used Fraction of Tokens and a variant of it, which computes the percentage of decision flips (i.e. changes in a model's prediction) observed after masking the more important token (as indicated by a feature scoring method). What is interesting about this study, is the fact that they also examined other attention-based feature scoring methods, namely attention-gradients and scaled attention. Similar to Jain and Wallace (2019), they also compared attention-based explanations with those from a gradient-based approach. They show across both experiments that explanations from attention-weights are less faithful than those of the sensitivity analysis approach. However, an interesting outcome was that scaled attention and attention-gradients outperformed attention weights. In fact, they performed comparably to the sensitivity analysis approach. Despite this finding, many studies following this work only use attention weights to compute input importance and not scaled attention or attention-gradients, which have been shown to be more faithful.

In a different direction, Madsen et al. (2021a) evaluated the faithfulness of explanations generated from attention weights, integrated gradients and simple gradients using ROAR and Rec-ROAR. Their comprehensive analysis shows that the performance of feature scoring methods is task-dependent, agreeing with the results of Atanasova et al. (2020). However, the performance of these three feature scoring methods overall is comparable to each other. This is an important finding as, attention weights do not impose any additional computational overhead when interpreting predictions, whilst integrated gradients require the computation of gradients multiple times in their experiments. This significant increase in computational overhead is not justified by the faithfulness of explanations generated by integrated gradients. Madsen et al. (2021a) show that attention weights *do* produce faithful explanations. They argue that the faithfulness of an explanation largely depends on how much the performance of a model drops when compared to a random baseline. The argument that attention weights can produce explanations is also supported by Jain et al. (2020). Using FRESH to build inherently faithful classifiers, Jain et al. (2020) show that using attention as part of the

rationale extractor results in classifiers with higher performance, compared to when using the simple gradients feature scoring method.

The conflicting outcomes of how faithful post-hoc explanations are from attention weights, are strong indicators of the difficulties in defining and evaluating faithfulness. Results vary across tasks, model architectures and rarely even across faithfulness evaluation approaches. Whilst proof via counter-example tests, such as finding adversaries (Jain and Wallace, 2019), are useful for evaluating how robust an explanation is, are not recently used for evaluating faithfulness. Latest research points to erasure approaches (i.e. masking the rationale or keeping only the rationale whilst masking the rest of the input) as the gold standard proxy for evaluating post-hoc explanations. Overall, there is a general trend which suggests that post-hoc explanation performance depends not only on the feature scoring method, but also on the task and model architecture used. Attention weights, seem to be regaining ground (Madsen et al., 2021a) after the criticism they have received (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019), whilst attention-based variants show promising that they can be more faithful (Serrano and Smith, 2019). Finally, it has been shown that more computationally expensive approaches such as LIME and IG, rarely justify this computational overhead with increased post-hoc explanation faithfulness (Nguyen, 2018b; Madsen et al., 2021a).

2.7 Improving Explanations

Previous research has shown that it is possible to influence attention-based explanations during training, by using auxiliary objectives (Kennedy et al., 2020; Wiegreffe and Pinter, 2019; Ross et al., 2017a; Liu and Avci, 2019). Auxiliary objectives are additional tasks given to the model during training, that complement the primary goal which is to learn the association between the input and the predicted labels. These objectives have typically been used as a tool for evaluating explanation faithfulness generated by attention (Kennedy et al., 2020; Wiegreffe and Pinter, 2019; Pruthi et al., 2020; Ghorbani et al., 2019). Wiegreffe and Pinter (2019) for example, use an auxiliary objective during training to influence the attention distribution of a model to be as far as possible from that of another model, whilst yielding similar predictions.

A different branch of studies, uses auxiliary objectives to improve the plausibility of ex-

planations generated by non-transformer based models (Ross et al., 2017a; Liu and Avci, 2019; Mohankumar et al., 2020). Liu and Avci (2019) use auxiliary objectives to reduce unintended biases exhibited by classifiers. Their objective function essentially penalises the distribution of token importance as indicated by a feature scoring method (in their case integrated gradients) using human annotated priors. Ross et al. (2017a) use masks to penalise the gradients to a prediction of tokens, where a model should not focus on. These masks are either annotated by human experts or are formed in an unsupervised manner. The unsupervised approach functions by generating an ensemble of masks and thus models, that are "right for different reasons". They then suggest that they can present the masks to a domain expert for finding the "right reasons" and using those to penalise model gradients. Mohankumar et al. (2020) take a different approach to improving the plausibility of attention weight explanations, by reducing the contextualisation in the word representations resulting from an LSTM encoder. They argue, similar to Tutek and Snajder (2020), that the contextualisation harms the explanation plausibility of attention weights and as such modify the LSTM-cell to reduce it.

Moradi et al. (2021) attempt to improve the faithfulness of attention weights in neural machine translation, by also proposing an objective function. Their approach is based on erasure evaluation, where the premise is that by perturbing or erasing important tokens indicated by attention, there should be a large divergence in the output probability distribution. As such, their auxiliary objective penalises the LSTM model when the output distribution does not change, when they (1) randomly perturb the attention weights; (2) erase the most important tokens and (3) introduce a uniform attention distribution. In a similar direction, Tutek and Snajder (2020) design a word-level auxiliary objective to reduce the contextualisation in the word representations of an LSTM encoder. Similar to Mohankumar et al. (2020), they argue that contextualisation harms the faithfulness of explanation. Their auxiliary objective attempts to reduce contextualisation, by penalising when the l2 norm of the difference between the word representations from the LSTM encoder and the word embeddings is large.

Such studies illustrate the effectiveness of auxiliary objectives for improving the faithfulness of model explanations. This suggests that there can be potential applications of auxiliary objectives for explanation faithfulness, particularly when considering improving the explanation faithfulness in large transformer-based models.

2.8 Summary

This chapter presents a background on model interpretability with a focus on explanation faithfulness. We begun by first providing prerequisite knowledge on neural text encoders and attention mechanisms. We then described the definition of what is a faithful explanation and presented different approaches available in literature for extracting explanations (i.e. feature scoring methods and select-then-predict models). We then show how explanation faithfulness is typically evaluated in previous studies and describes how feature scoring methods and select-then-predict models perform in terms of producing faithful explanations. Finally, we discuss a different branch of studies which focuses on improving explanations in non-transformer based model architectures.

Chapter 3

Improving the Faithfulness of Post-hoc Explanations

Previously, we have described the current state of post-hoc explanation faithfulness and how recent studies attempted to improve it. However, proposed methods are either encoder-specific and often do not cover transformer-based models (Moradi et al., 2021; Tutek and Snajder, 2020; Mohankumar et al., 2020). Inspired by these limitations, this chapter proposes two novel approaches for improving the faithfulness of post-hoc explanations extracted using feature attribution methods. Section 3.1 is concerned with improving attention-based post-hoc explanations for text classification, whilst Section 3.2 focuses on improving the faithfulness of post-hoc explanations in transformer-based models.

3.1 Improving Attention-based Explanations

3.1.1 Motivation

As described in Section 2.6.1, a common practice is providing 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 series

of recent studies illustrated 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).

Attention indicates how well inputs around a position i correspond to the output (Bahdanau et al., 2015), due to the contextual dependencies of the words in the sequence. 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. 2.1) by its attention score and a non-contextualised word type scalar score, attention-based explanations can be improved. The intuition is that by having a less contextualised sequence representation $\mathbf{c}$, information mixing can be reduced for attention.

For example, consider the following review: *"The food was amazing, service was excellent and the price decent."*. With current architectures, the attention score for the word "amazing", is affected by the surrounding context, with stronger influence from the tokens close to it, e.g. the words "service" and "was". As such, by applying the non-contextualised score s_{x_i}, we aim to reduce this influence and have a more representative importance score from attention, without losing the all-important contextualisation in the token representation h_i. This section is organised into five sub-sections. Section 3.1.2 describes the methodology for infusing non-contextualised information in the model during training. Section 3.1.3 presents the experimental set-up and finally Section 3.1.4 discusses the results obtained.

3.1.2 Methodology

We propose the Task-Scaling (**TaSc**) mechanism, which learns task specific token level importance s_{x_i} complementary to the encoder representations and attention scores in Eq. 2.1, such that:

$$\mathbf{c} = \sum_i \mathbf{h}_i \alpha_i s_{x_i}, \quad \mathbf{c} \in \mathrm{R}^{\mathrm{N}} \tag{3.1}$$

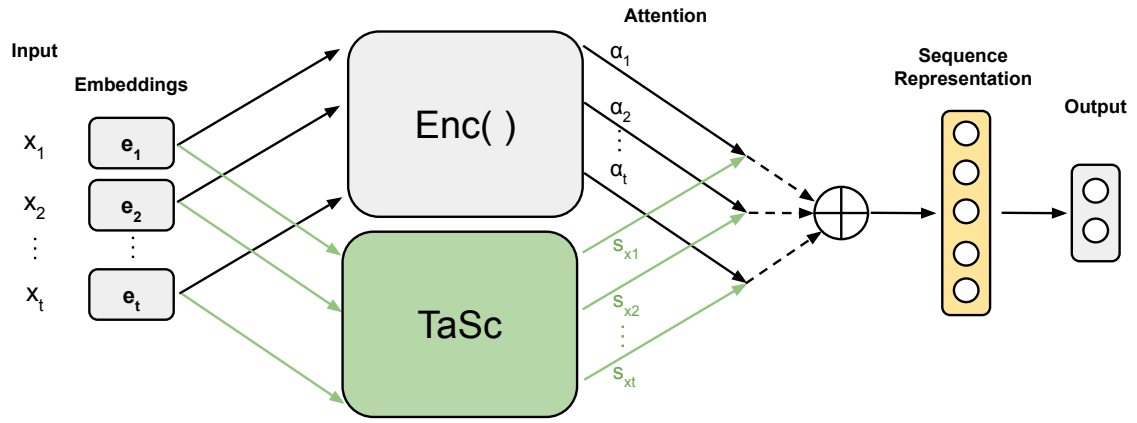


Figure 3.1: A text classification pipeline with the proposed TaSc component (where, $|\mathbf{x}| = |\boldsymbol{\alpha}| = |\boldsymbol{s}_x|$).

The proposed mechanism can be visualised in Figure 3.1, where we compute s_{x_i} by proposing three Task-Scaling (**TaSc**) mechanisms.[1]

Linear TaSc (Lin-TaSc): The first TaSc mechanism to introduce is Linear TaSc (Lin-TaSc), the simplest and lightest 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:

$$\hat{\mathbf{e}}_i = u_i \mathbf{e}_i \tag{3.2}$$

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 $\hat{\mathbf{e}}_i$; $s_{x_i} = \sum^{d} \hat{\mathbf{e}}_i$

[1]Number of parameters for each proposed mechanism in Table 3.2.

in a similar way that token embeddings are averaged in the top-layers of a 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). Additionally, max and mean-pooling or using the u_i directly instead of s_i in early experimentation resulted in lower results. We hypothesize that the summation resulted in the best outcomes, as it retains all of the information of the scaled embedding vector rather than using subsets of it as with the max-pooling, whilst mean-pooling has similar effects to averaging the vector (where averaging has been shown to reduce performance (Atanasova et al., 2020)).

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 in turn results into a richer sequence representation $\mathbf{c}$.

Feature-wise TaSc (Feat-TaSc): Lin-TaSc assigns equal weighting to all the dimensions of the word embedding $\mathbf{e}_i$ (see Eq. 3.2), 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 $\mathbf{e}_i$;

$$s_{x_i} = \mathbf{u}_i \cdot \mathbf{e}_i \tag{3.3}$$

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.[2] We apply a CNN[3] with n channels over the scaled embedding $\hat{\mathbf{e}}_i$ from Lin-TaSc, keeping a single stride and a 1-dimensional

[2]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.

[3]See CNN configurations in Section 3.1.3.

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};

$$s_{x_i} = \sum^{d} \hat{\mathbf{e}}_i^f \tag{3.4}$$

3.1.3 Experimental Setup

Dataset	Av. $\lvert W \rvert$	$\lvert \mathbf{V} \rvert$	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 3.1: Dataset statistics including average number of words (**Av.** $\lvert W \rvert$) per instance, vocabulary size ($\lvert \mathbf{V} \rvert$) and splits.

Datasets: We consider the following datasets for text classification following Wiegreffe and Pinter (2019) and Jain and Wallace (2019) (see Table 3.1 for details):

SST: *Stanford Sentiment Treebank* consists of sentences tagged with sentiment on a 5-point-scale from negative to postive (Socher et al., 2013). Jain and Wallace (2019) removed sentences with neutral sentiment and labelled the remaining sentences to negative and positive if they have a score lower or higher than 3 respectively.

IMDB: The *Large Movie Reviews Corpus* consists of 50,000 movie reviews labelled either as positive or negative (Maas et al., 2011). We filter the dataset as per Jain and Wallace (2019) to include movie reviews with sequence length less than 400 words.

ADR: A dataset of ∼20,000 tweets with labels indicating whether a Twitter post contains an adverse drug reaction or not Sarker et al. (2015).

TaSc Mechanism	Additional Parameters
Lin-TaSc	$\|\mathbf{V}\|$
Feat-TaSc	$\|\mathbf{V}\| \times d$
Conv-TaSc	$\|\mathbf{V}\| + d \times n + n$

Table 3.2: Additional parameters resulting from the proposed TaSc mechanisms where $|V|$ is the vocabulary size, d the embedding dimension and n the number of channels in a CNN.

AG: A subset of the original news articles[4] dataset compiled by Jain and Wallace (2019) for topic categorisation (*Business* and *World* news).

MIMIC: A sample of discharge summaries from the MIMIC III dataset of health records (Johnson et al., 2016). The task is to recognise if a given summary has been labelled as relevant to acute or chronic anemia (Jain and Wallace, 2019).

Models and Hyperparameters: Similar to Jain and Wallace (2019) we use FastText pretrained embeddings (Joulin et al., 2016) for the SST and ADR datasets, Glove pretrained embeddings (Pennington et al., 2014) for the IMDB and AG News datasets, while we use Word2Vec (Mikolov et al., 2013) from Gensim (Řehůřek and Sojka, 2010) to train embeddings for MIMIC. All embeddings are of size $d = 300$. We also replace all numbers in text with a special symbol q and initialise the embeddings of unknown words randomly from a normal distribution, $\mathcal{N}(0, 1)$. The embeddings are not trained alongside the rest of the model.

We train the models using default Adam learning rate (1e-3) with 1e-4 weight decay, which adds an l_2 regulariser across all parameters. We use 64 dimensional hidden representations for one-layered bi-LSTM and bi-GRU encoders and 128 dimensional hidden representation for the MLP encoder following Jain and Wallace (2019). For the CNN we use 4 kernels of sizes [1, 3, 5, 7], each with 32 filters, giving a final contextual representation $\mathbf{h}_i$ of size $N = 128$, with ReLU activation function on the output of the filters, as per Jain and Wallace (2019). For BERT we use the pre-trained version from Wolf et al. (2019) and fine-tune with a learning rate of $1e-5$ all BERT parameters except from the word embeddings, to simulate the scenario with the rest of the encoders, and $1e-4$ for the remainder of the parameters. We train our models three times using different random seeds and a batch size of 8 for BERT

[4] https://di.unipi.it/~gulli/AG_corpus_of_news_articles.html. Accessed on Sep 2019

and 32 for the rest of the models.

For Conv-TaSc we apply a CNN with 15 channels over the scaled embedding $\mathbf{e}_i$ 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 $\mathbf{e}_i^f$, to obtain the scores s_{x_i}. We have also experimented with filter sizes of [2, 10, 20 , 30, 50] individually and simultaneously.

For the MIMIC dataset we also attempted to use LongFormer (Beltagy et al., 2020), which is a BERT version that has the ability to accept and deal with longer sequences. However due to the increasing time to train and evaluate the model, this BERT variant was abandoned. Additionally we attempted to use Hierarchical BERT to deal with the longer sequences, however increases were not substantial and run times where similarly increased. Finally, contrary to the remainder of the datasets to deal with the long sequences of MIMIC we truncated the 256 first tokens and 256 last tokens, following the suggestions of Sun et al. (2019). We experimented with using the first and the last 512 tokens, but the head and tails truncation approach yielded the best performances. In Table 3.2 we also present the additional parameters introduced by each variant, with Lin-TaSc requiring the lowest number of parameters and Feat-TaSc the most.

Attention-based Importance Metrics We use three input importance metrics by Serrano and Smith (2019): (1) attention (α); (2) attention-gradients ($\nabla\alpha$); and finally (3) scaled attention ($\alpha\nabla\alpha$).[5]

Evaluating Attention-based Interpretability: We use the Fraction of Tokens erasure approach to evaluate faithfulness, similar to Serrano and Smith (2019), Atanasova et al. (2020) and Nguyen (2018a). Also, similar to Serrano and Smith (2019) we use a variation of Fraction of Tokens, called Decision Flip which records the percentage of flips observed by removing the single most informative token (for evaluation see details described in Section 2.6), where higher is better.[6] 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 correspond-

[5]See Section 2.5.1 for more details on feature attribution approaches.

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

Dataset	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[7]	.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 3.3: Average F1 macro (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

ing 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, 2018a; Vashishth et al., 2019; Grimsley et al., 2020; Atanasova et al., 2020; Madsen et al., 2021a).

3.1.4 Results

Predictive Performance: A prerequisite of interpretability is to obtain robust explanations without sacrificing predictive performance (Lipton, 2016). Table 3.3 shows the macro

	Attention	No-TaSc	Lin-TaSc	Feat-TaSc	Conv-TaSc
α	Tanh	**7.46**	5.69 (0.8)	5.89 (0.8)	5.00 (0.7)
	Dot	**4.89**	3.93 (0.8)	4.42 (0.9)	3.90 (0.8)
$\nabla\alpha$	Tanh	7.88	9.60 (1.2)	**9.92** (1.3)	9.92 (1.3)
	Dot	7.09	10.79 (1.5)	**11.04** (1.6)	10.40 (1.5)
$\alpha\nabla\alpha$	Tanh	11.32	**12.69** (1.1)	12.19 (1.1)	11.52 (1.0)
	Dot	8.47	**11.36** (1.3)	**11.36** (1.3)	10.56 (1.2)

Table 3.4: 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).

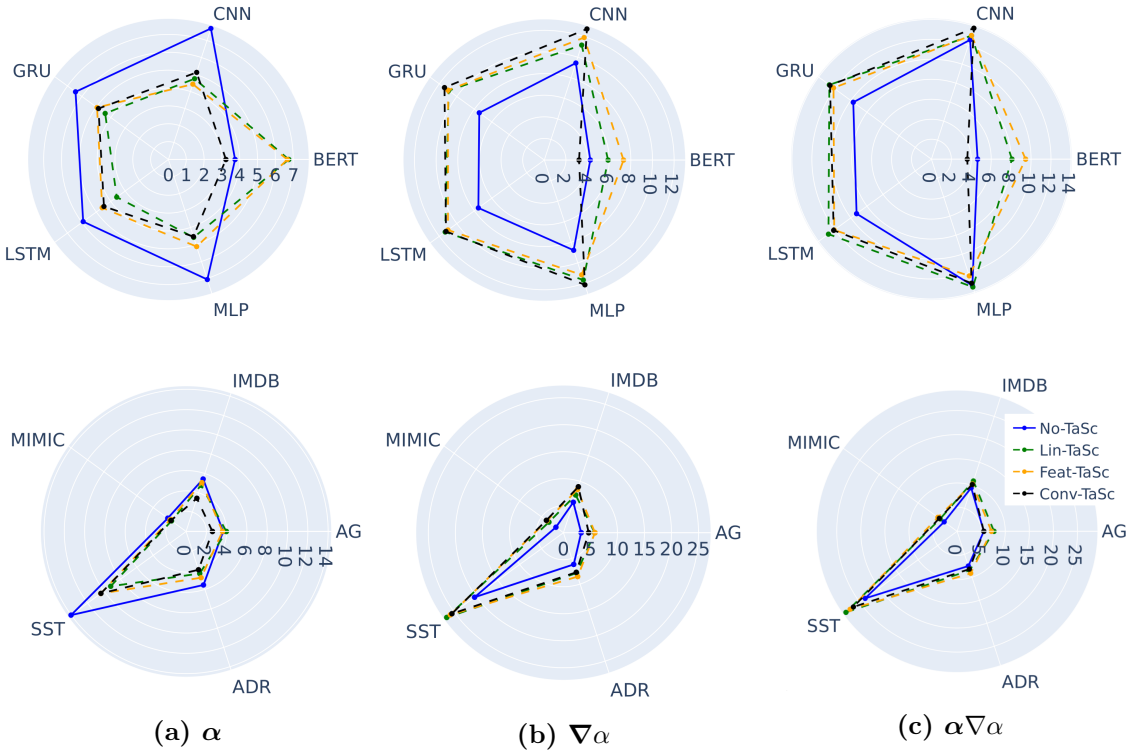
F1-scores of all models across datasets, encoders and attention mechanisms using the three TaSc variants (Lin-TaSc, Feat-TaSc and Conv-TaSc described in Section 3.1.2) and without TaSc (No-TaSc).

In general, all TaSc models obtain comparable performance and in some cases outperform No-TaSc across datasets and attention mechanisms. Our main aim is not to improve predictive performance, however it is positive to observe comparable performances with only three cases resulting in slight performance deteriorations (maximum of 2 F1 points with MIMIC and CNN). We argue that even in those cases it is up to the end-user to select whether or not this sacrifice is worth the improvements in faithfulness of attention-based explanations, which we illustrate below.

Decision Flip: Table 3.4 and Figure 3.2 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.

In Table 3.4, 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

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



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

Figure 3.2: Mean 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.

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 mechanisms, with Dot mechanism benefiting the most, making it comparable to Tanh. For example, Dot moves from 8.47% with No-TaSc to 11.36% with Lin-TaSc, which is closer to 12.69% achieved by Lin-TaSc with Tanh (for $\alpha\nabla\alpha$).

The first row of Figure 3.2 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 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

	Attention	No-TaSc	Lin-TaSc	Feat-TaSc	Conv-TaSc
α	Tanh	.52	**.48** (0.9)	.51 (1.0)	.53 (1.0)
	Dot	.60	**.58** (1.0)	.59 (1.0)	.61 (1.0)
$\nabla\alpha$	Tanh	.36	.23 (0.6)	**.22** (0.6)	.26 (0.7)
	Dot	.40	**.23** (0.6)	**.23** (0.6)	.27 (0.7)
$\alpha\nabla\alpha$	Tanh	.32	**.20** (0.6)	**.20** (0.6)	.25 (0.8)
	Dot	.37	**.22** (0.6)	**.22** (0.6)	.27 (0.7)

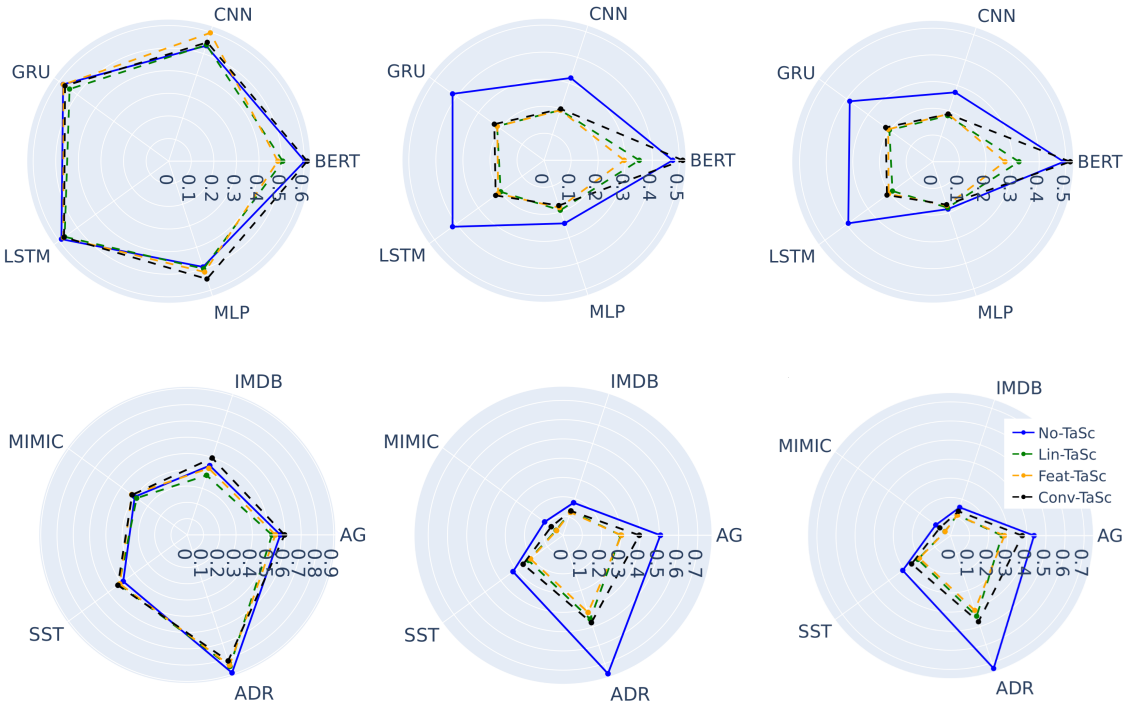
Table 3.5: Mean *fraction of tokens* required to cause a decision flip across attention mechanisms, using the three TaSc variants and No-TaSc (lower is better). **Bold** and underlined values denote best performing method row-wise and overall (for each attention mechanism). Relative improvement over No-TaSc in parenthesis (<1 TaSc is better than No-TaSc).

(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 3.2, 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.

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 3.5 and Figure 3.3 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.

In Table 3.5, 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-TaSc (in 16 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.

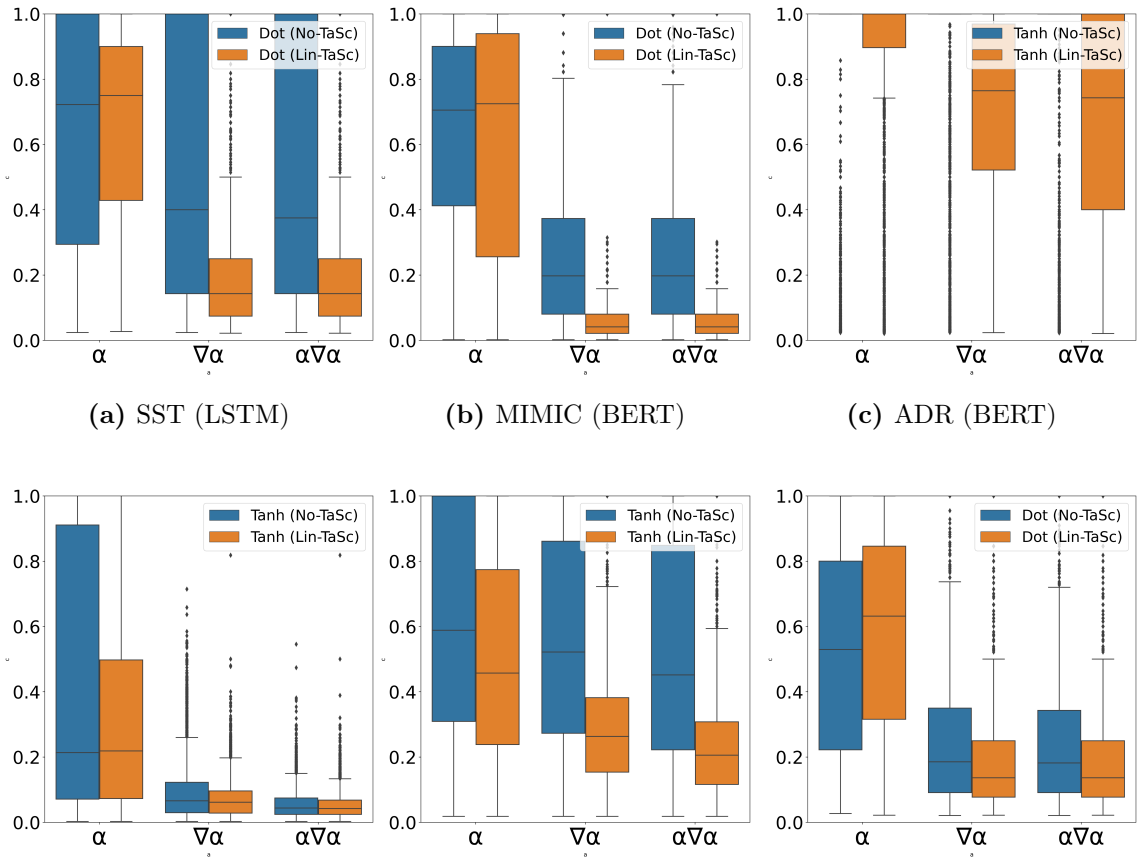


(a) $\boldsymbol{\alpha}$ **(b)** $\boldsymbol{\nabla\alpha}$ **(c)** $\boldsymbol{\alpha\nabla\alpha}$

Figure 3.3: Mean *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.

We present an across encoders comparison in the first row of Figure 3.3. 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. 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 .40 with No-TaSc to .17 with Lin-TaSc, .18 with Feat-TaSc and .20 with Conv-TaSc (for $\alpha\nabla\alpha$).

The second row of Figure 3.3 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 .75 of all tokens to be removed



(a) SST (LSTM) **(b)** MIMIC (BERT) **(c)** ADR (BERT)

(d) IMDB (MLP) **(e)** AG (GRU) **(f)** SST (CNN)

Figure 3.4: 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).

for a decision flip to occur compared to .43 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.

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 3.4 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 six 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 MIMIC 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 more consistent.

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, 2018a; Atanasova et al., 2020). For this purpose we use: (1) Word Omission (**WO**); (2) InputXGrad ($\mathbf{x}\nabla\mathbf{x}$); and (3) Integrated Gradients (**IG**).

Table 3.6 shows the results using Fraction of Tokens for evaluation, 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.[8] 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 46 out of 50 cases, whilst obtaining comparable performance in other 3 cases. This demonstrates the efficacy of TaSc in providing more faithful attention-based explanations than strong baselines without TaSc (Nguyen, 2018a; Atanasova et al., 2020). The improvements are particularly evident using BERT as an encoder. In ADR, WO with Tanh requires on average .81 of the tokens to be removed for a decision flip compared to just .44 for $\alpha\nabla\alpha$ with TaSc.

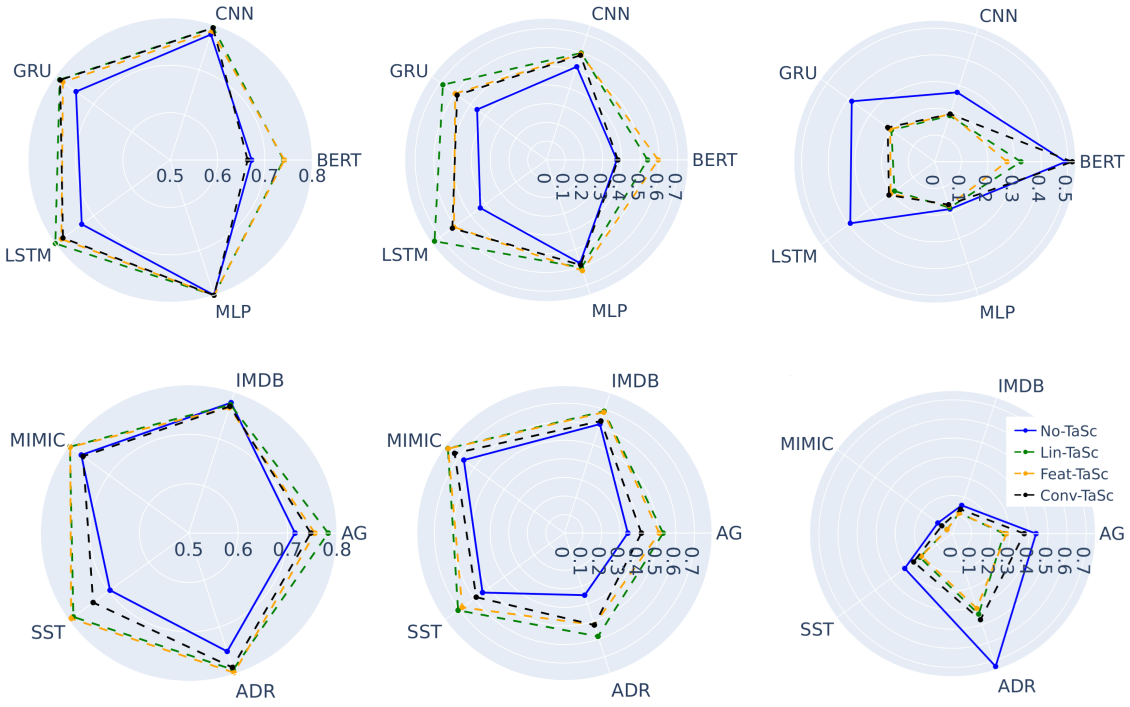
We also observe that the attention-based importance metric ($\alpha\nabla\alpha$) with TaSc is a more robust explanation technique than non-attention based ones, obtaining lower variance in the fraction of tokens required to cause a decision flip across encoders. For example $\alpha\nabla\alpha$ with TaSc and Tanh requires a fraction of tokens in the range of .03-.07 compared to IG which requires .05-.14 in MIMIC, showing the consistency of our proposed approach.

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

		Tanh							Dot						
		Non-TaSc			TaSc				Non-TaSc			TaSc			
Dataset	**Enc()**	**WO**	**x∇x**	**IG**	**WO**	**x∇x**	**IG**	**α∇α**	**WO**	**x∇x**	**IG**	**WO**	**x∇x**	**IG**	**α∇α**
SST	BERT	0.29	0.57	0.49	0.38	0.30	0.35	**0.26**	**0.30**	0.57	0.46	0.31	0.31	0.38	0.31
	LSTM	0.22	0.22	0.38	0.22	**0.20**	0.45	**0.20**	0.26	0.28	0.45	0.22	**0.20**	0.45	**0.20**
	GRU	0.24	0.23	0.36	0.23	0.20	0.43	**0.19**	0.22	0.23	0.42	0.22	**0.20**	0.45	**0.20**
	MLP	0.29	0.23	0.38	0.26	0.21	0.40	**0.19**	0.21	0.19	0.38	0.21	0.19	0.39	**0.18**
	CNN	0.27	0.24	0.36	0.23	0.21	0.40	**0.20**	0.21	0.20	0.37	0.21	0.20	0.40	**0.19**
ADR	BERT	0.77	0.89	0.86	0.79	0.81	0.80	**0.65**	0.81	0.92	0.90	0.50	0.47	0.57	**0.44**
	LSTM	0.77	0.77	0.80	0.57	0.50	0.75	**0.44**	0.81	0.82	0.86	0.48	0.41	0.73	**0.40**
	GRU	0.77	0.76	0.80	0.55	0.47	0.73	**0.42**	0.78	0.81	0.84	0.49	**0.41**	0.74	**0.41**
	MLP	0.68	0.61	0.74	0.58	0.45	0.71	**0.40**	0.52	0.47	0.55	0.53	0.46	0.73	**0.45**
	CNN	0.68	0.68	0.74	0.53	0.47	0.68	**0.40**	0.66	0.66	0.72	0.49	0.45	0.70	**0.41**
AG	BERT	0.45	0.71	0.63	0.56	0.42	0.48	**0.31**	0.50	0.72	0.66	**0.35**	0.46	0.53	0.45
	LSTM	0.45	0.54	0.65	0.43	0.31	0.65	**0.23**	0.35	0.43	0.59	0.36	0.28	0.69	**0.23**
	GRU	0.40	0.48	0.61	0.46	0.32	0.67	**0.22**	0.37	0.44	0.55	0.33	**0.32**	0.61	0.40
	MLP	0.55	0.63	0.77	0.44	0.23	0.61	**0.21**	0.44	0.71	0.73	0.28	**0.22**	0.60	**0.22**
	CNN	0.51	0.40	0.63	0.47	0.33	0.62	**0.22**	0.41	0.39	0.61	0.35	0.29	0.62	**0.22**
IMDB	BERT	0.26	0.66	0.58	0.28	0.26	0.22	**0.09**	0.32	0.68	0.65	**0.27**	0.68	0.62	0.59
	LSTM	0.12	0.09	0.26	0.08	**0.06**	0.19	**0.06**	0.09	0.09	0.32	**0.06**	0.06	0.19	0.06
	GRU	0.10	0.10	0.28	0.08	0.06	0.18	**0.05**	0.10	0.15	0.36	**0.06**	**0.06**	0.20	**0.06**
	MLP	0.06	0.06	0.25	0.07	**0.05**	0.14	**0.05**	0.07	**0.06**	0.30	**0.06**	**0.06**	0.14	**0.06**
	CNN	0.11	0.08	0.21	0.09	0.07	0.17	**0.05**	0.09	0.08	0.21	0.08	0.08	0.17	**0.06**
MIMIC	BERT	0.12	0.68	0.54	0.12	0.13	0.14	**0.07**	0.15	0.70	0.60	0.09	0.07	0.10	**0.06**
	LSTM	0.26	0.24	0.28	0.07	0.03	0.10	**0.02**	0.07	0.40	0.42	**0.03**	**0.03**	0.09	**0.03**
	GRU	0.17	0.12	0.20	0.04	**0.03**	0.08	**0.03**	0.05	0.15	0.23	**0.03**	**0.03**	0.08	**0.03**
	MLP	0.09	0.04	0.12	0.04	**0.02**	0.05	**0.02**	0.03	0.04	0.19	**0.03**	**0.03**	0.06	**0.03**
	CNN	0.12	0.07	0.11	**0.03**	**0.03**	0.06	**0.03**	0.06	0.05	0.11	0.04	0.04	0.06	**0.03**

Table 3.6: Average *fraction of tokens* required to cause a decision flip using the best performing attention-based ranking ($\alpha\nabla\alpha$) with TaSc and *Word omission*, (**WO**), *InputXGrad*, ($\nabla\mathbf{x}$) and *Integrated Gradients* without TaSc (Non-TaSc) and with TaSc (**IG**). Underlined values denote that Lin-TaSc is better and **bold** values denote the best performing method row-wise. (lower is better).

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.



(a) AOPC NormSuff **(b)** AOPC NormComp **(c)** Fraction of Tokens

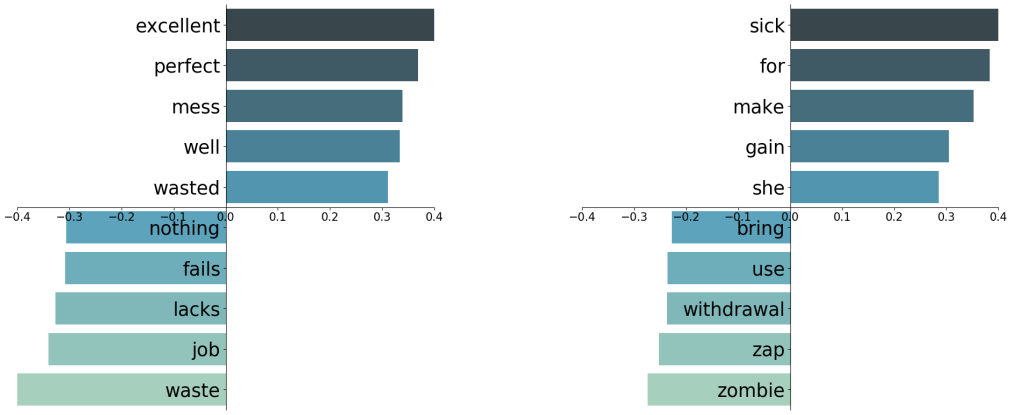
Figure 3.5: Mean AOPC Normalised Sufficiency (higher is better), AOPC Normalised Comprehensiveness (higher is better) and Fraction of Tokens (lower is better) occurred by removing the most informative token, using the three TaSc variants and No-TaSc across encoders (first row) and datasets (second row), using $\alpha\nabla\alpha$.

3.1.5 Evaluation across Faithfulness Metrics

We now compare the performance of our proposed TaSc mechanisms, across a larger range of post-hoc explanation faithfulness evaluation metrics. This analysis aims to examine whether observations hold across evaluation approaches. For this purpose, alongside Fraction of Tokens (lower is better) we also employ AOPC Normalised Sufficiency (AOPC NormSuff; higher is better) and Normalised Comprehensiveness (AOPC NormComp; higher is better). For more details on these approaches, see Section 2.6. For clarity, in Figure 3.5 we compare results across the three evaluation metrics when using $\alpha\nabla\alpha$.[9]

Overall, results show that observations hold across all the three evaluation metrics used. For example encoder-wise (first row), our three proposed TaSc mechanisms improve across

[9]For results with α and $\nabla\alpha$ see Appendix A.2.



(a) IMDB - LSTM - Dot

(b) ADR - GRU - Tanh

Figure 3.6: Highest and lowest scored 5 words from learnable parameter **u** with LSTM encoder and Dot mechanism for the IMDB dataset.

all encoders, with increased improvements observed over LSTM, GRU and BERT compared to MLP and CNN. Observing across the three evaluation approaches, AOPC NormComp and Fraction of Tokens appear to compute scores with a larger variation compared to AOPC NormSuff. For example using AOPC NormComp and BERT, scores range from 0.40 to 0.70. In comparison using AOPC NormSuff and BERT, scores range from 0.66 to 0.74.

The findings from these results are two fold. Firstly, our proposed TaSc mechanisms result in more faithful attention-based explanations, irrespective of which metric we use to measure faithfulness. Secondly, we observe that the three tested evaluation metrics produce similar conclusions across tasks and mechanisms. This suggests that using either as a proxy for faithfulness should be sufficient to draw conclusions.

3.1.6 Qualitative Analysis

We finally examine qualitatively what type of information the parameter **u** from Lin-TaSc learns. Similar to a bag-of-words model, our initial hypothesis is that **u** will assign high scores to the words that are most relevant to the task. Figure 3.6 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 3.6a, that indeed words expressing sentiment are assigned high scores (e.g. *excellent, waste, perfect*), either positive or negative. However, a positive or negative sign does 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.

3.2 Improving Transformer-based Explanations

3.2.1 Motivation

In Section 3.1 we proposed a family of mechanisms that improve the faithfulness of attention-based post-hoc explanations. However, as we have previously shown through empirical results and discussed in Section 2.6.1, large pre-trained language models (LMs) such as BERT (Devlin et al., 2019) still produce less faithful explanations compared to simpler model architectures (e.g. LSTM, MLP). This also holds true, despite of the improvements offered by our proposed mechanisms.

Previous work has explored whether LMs encode syntactic knowledge, by studying their multi-head attention distributions (Clark et al., 2019; Htut et al., 2019; Voita et al., 2019). Recent studies have evaluated the faithfulness of explanations for predictions made by these models (Vashishth et al., 2019; Atanasova et al., 2020; Jain et al., 2020). In general, LMs can provide faithful explanations, particularly using attention (Jain et al., 2020), but still fall behind other simpler architectures (Atanasova et al., 2020) possibly due to increased information mixing and higher contextualisation in the model (Brunner et al., 2020; Pascual et al., 2020; Tutek and Snajder, 2020). Previous studies have attempted to improve the explainability of non transformer-based models, by guiding them through an auxiliary objective towards informative input importance distributions (e.g. human or adversarial priors) (Ross et al., 2017a; Liu and Avci, 2019; Moradi et al., 2021).

Targeting post-hoc explanations from transformer-based models, we propose **Sa**lient **Loss** (**SaLoss**); an auxiliary objective that allows the multi-head attention of the model to learn from salient information (i.e. token importance) during training, to reduce the effects of information mixing (Pascual et al., 2020) and improve post-hoc explanation faithfulness. We compute a priori token importance scores (Xu et al., 2020) using TextRank (Mihalcea and Tarau, 2004) (i.e. an unsupervised graph-based method) and penalise the model when the attention distribution deviates from the salience distribution.

This section is organised into five sub-sections. Section 3.2.2 describes the methodology for infusing non-contextualised information in the model during training. Section 3.2.3 presents the experimental set-up and Section 3.2.4 discusses the results obtained.

3.2.2 Methodology

Even though attention scores can be more faithful than other feature attribution approaches (Jain et al., 2020), they usually pertain to their corresponding input tokens in *context* and not individually due to information mixing (Tutek and Snajder, 2020; Pascual et al., 2020). As such, we hypothesise that we can improve the ability of a pretrained LM in providing *faithful* explanations, by showing to the model alternative distributions of input importance (i.e. word salience). We assume that by introducing the salience distribution via an auxiliary objective (Ross et al., 2017b), we can reduce information mixing by "shifting" the model's attention to other informative tokens. In a similar direction to ours, Xu et al. (2020) showed that by computing attention together with salience information from keyword extractors improves text summarisation.

Computing Word Salience: We compute word salience $\boldsymbol{\sigma}$ using TextRank (Mihalcea and Tarau, 2004), an unsupervised graph-based model for keyword extraction, where $G = (V, E, W)$.[10] Inspired by PageRank (Page et al., 1999), TextRank calculates the indegree centrality of graph nodes iteratively based on a Markov chain, where each node (V) is a wordpiece and each edge (E) links wordpiece pairs within a context window (Xu et al., 2020). W represents the edge (E) thicknesses. For each input document X we first construct an undirected graph. We represent the graph as a square matrix of vocabulary length, where each row represents the inbound links from other nodes V_i and each column the outbound links V_j. We then apply TextRank iteratively to compute the local salience scores (σ_i) of its words by:

$$\sigma_i = (1 - d) + d \sum_{V_j \in In(V_i)} \frac{W_{ji}}{\sum_{V_k \in Out(V_j)} W_{jk}} \sigma_j \tag{3.5}$$

where d is the damping coefficient, $In(V_i)$ and $Out(V_j)$ are the incoming and outgoing nodes. As shown from the above equation, the score of node i (V_i) depends on the edge weight W_{ji} (i.e. from V_j to V_i) and the sum of edge weights from node V_j to all other nodes, where the maximum k is equal to the sequence length t. The salience scores are updated until either

[10] We also considered the use of Tfidf and χ^2 scores observing comparable but lower performance in early experimentation. We hypothesise that TextRank performs well due to its effectiveness in improving performance in text summarisation (Xu et al., 2020). See also Section 3.2.6 for Tfidf and χ^2 results on input erasure experiments.

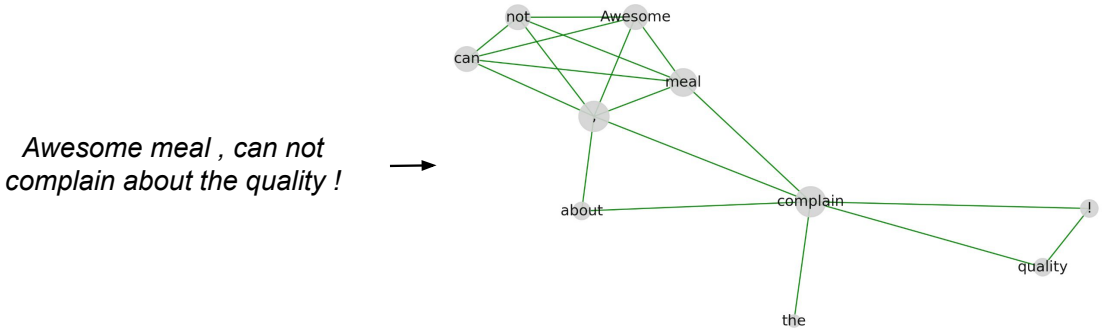


Figure 3.7: Demonstrative example of using TextRank to compute token sequence importance σ_i (larger node size indicates higher importance, window size of 5).

convergence or the pre-defined number of iterations has been reached.[11]

Our intuition is that by using the task-agnostic TextRank, we can extract words that are important in the context of the sequence and as such offer an alternative view of token importance. We include a demonstrative example of TextRank for clarity in Figure 3.7.

Salience Loss: We propose Salient Loss (SaLoss), an auxiliary objective which allows the model to learn attending to more informative input tokens jointly with the task. SaLoss penalises the model when the attention distribution ($\boldsymbol{\alpha}$) deviates from the word salience distribution ($\boldsymbol{\sigma}$).[12] For $\boldsymbol{\alpha}$ we compute the average attention scores of the CLS token from the last layer (Jain et al., 2020). We only use the attention scores from the last layer, as it we can influence the remainder downstream model parameters through them. The joint objective for adapting a LM to a downstream classification task with SaLoss is:

$$\mathcal{L} = \mathcal{L}_c + \lambda \mathcal{L}_{sal} \tag{3.6}$$

where $\mathcal{L}_c$ is the Cross-Entropy Loss for a downstream text classification task and λ a regularisation coefficient for the proposed SaLoss ($\mathcal{L}_{sal}$) which can be tuned in a development set. $\mathcal{L}_{sal}$ is defined as the KL divergence between $\boldsymbol{\alpha}$ and $\boldsymbol{\sigma}$:

$$\mathcal{L}_{sal} = KL(\boldsymbol{\alpha}, \boldsymbol{\sigma}) = \sum_{i=0}^{t} \alpha_i (\log \alpha_i - \log \sigma_i) \tag{3.7}$$

[11] We describe how σ_i parameters are initialised in 3.2.3.

[12] $\boldsymbol{\alpha} \in \mathrm{R}^{\mathrm{t}}$; $\boldsymbol{\sigma} \in \mathrm{R}^{\mathrm{t}}$, where t is the sequence length.

Data	Av. $\lvert W \rvert$	C	Splits Train/Dev/Test
SST	18	2	6,920 / 872 / 1,821
AG	36	4	102,000 / 18,000 / 7,600
Ev.Inf.	363	3	5,789 / 684 / 720
MultiRC	305	2	24,029 / 3,214 / 4,848
SEMEVAL	20	3	6,000 / 2,000 / 20,630

Table 3.7: Dataset statistics including average words per input, number of classes and splits.

We assume a standard text classification setting where a set of labeled documents is used for fine-tuning a pretrained LM by adding an extra output classification layer. We normalise the salience scores for compatibility with the KL divergence.[13]

3.2.3 Experimental Setup

Datasets: For our experiments we use **SST** (Socher et al., 2013) as in Section 3.1.3 and also the following tasks (see dataset details in Table 3.7.):

AgNews (AG): News articles categorised by the following topics; Science, Sports, Business, and World (Corso et al., 2005).

Evidence Inference (Ev.Inf.): Abstract-only biomedical articles describing randomised controlled trials (Lehman et al., 2019). The task is to infer the reported relationship between a given intervention and comparator with respect to an outcome.

Multi Reading Comprehension (MultiRC): A reading comprehension dataset composed of questions with multiple correct answers, which depend on information from multiple sentences (Khashabi et al., 2018). Similar to DeYoung et al. (2020) and Jain et al. (2020) we convert this to a binary classification task where each rationale/question/answer triplet forms an instance and each candidate answer has a label of True or False.

[13]We use KL divergence, as it was successfully used in literature to influence model learning through priors (Yang et al., 2017).

Dataset	Model	lr^m	lr^c	F1
SST	bert-base	1e-5	1e-4	.91 ± .00
AG	bert-base	1e-5	1e-4	.93 ± .00
Ev.Inf.	scibert	5e-6	2e-4	.84 ± .01
MultiRC	roberta-base	2e-6	2e-4	.75 ± .01
SEMEVAL	bert-base	1e-5	1e-4	.59 ± .02

Table 3.8: Model and their hyper-parameters for each dataset, including learning rate for the model (lr^m) and the classifier layer (lr^c) and F1 macro scores on the development set across three runs.

SEMEVAL: The Semeval 2017 dataset for Task 4 Subtask A which consists of tweets and the task is to classify whether the message is of positive, negative, or neutral sentiment Rosenthal et al. (2017).

Models and Hyperparameters: Similar to Jain et al. (2020) we use: BERT (Devlin et al., 2019) for (SST, AG, SEMEVAL); SciBERT (Beltagy et al., 2019) for Ev.Inf.; Roberta (Liu et al., 2019b) for MultiRC. Table 3.8 presents the hyper-parameters used to train the models across different datasets, along with F1 macro perfor mance on the development set. Models where finetuned across 3 runs for 10 epochs, with the exception of the SEMEVAL dataset which was finetuned for 20. We implement our models using the Huggingface library (Wolf et al., 2019) and use default parameters of the AdamW optimiser apart from the learning rates and a linear scheduler. Experiments run on a single Nvidia Tesla V100 GPU.

We found that the learning rate of our proposed objective, does not impact significantly F1 macro performance. As such, since our objective is improving faithfulness, our λ selection includes training then evaluating on the development set the average fraction of tokens required to cause a decision flip. We use the model with the lowest fraction of tokens scores and report on the test set.

Computing Salience Scores with TextRank: We run TextRank for 20 steps, or until convergence, with a window of 4 words, a damping coefficient of 0.85 and normalise the salience scores to make them more compatible to attention distributions.

Dataset	Baseline	λ	SaLoss
SST	.91 (.00)	1e-3	.91 (.00)
AG	.93 (.00)	1e-4	.93 (.00)
Ev.Inf.	.82 (.01)	1e-4	.80 (.02)
MultiRC	.76 (.01)	1e-3	.76 (.00)
SEMEVAL	.58 (.01)	1e-3	.57 (.03)

Table 3.9: F1 macro averaged across 3 seeds for vanilla LMs (BASELINE) and SALOSS models. λ represents the regularisation coefficient of our proposed objective.

Feature Attribution Approaches We use the following popular metrics to allocate importance to input tokens: (1) Normalised attention scores ($\boldsymbol{\alpha}$); (2) Scaled attention ($\boldsymbol{\alpha}\nabla\boldsymbol{\alpha}$) (Serrano and Smith, 2019); (3) InputXGrad ($\mathbf{x}\nabla\mathbf{x}$) (Kindermans et al., 2016; Atanasova et al., 2020); and (4) Integrated Gradients (**I.G.**) (Sundararajan et al., 2017).

Evaluating Explanation Faithfulness We evaluate the faithfulness of model explanations using two approaches: (1) Fraction of Tokens; and (2) FRESH (see Section 2.6).

3.2.4 Results

Predictive Performance: Table 3.9 shows F1 macro scores averaged over three runs with standard deviation across tasks, for vanilla pretrained LMs (Baseline) and models with our proposed objective SaLoss. Results demonstrate that models trained with our proposed salience objective[14] achieve similar performance to the Baseline models across datasets.

Fraction of Tokens: Table 3.10 shows results for the average fraction of input tokens required to be removed to cause a decision flip for Baseline and SaLoss models in the test set. Results suggest that models trained with our proposed objective require a significantly lower fraction of tokens removed to cause a decision flip in 19 out of 20 cases (Wilcoxon Rank Sum, $p < .05$), with the exception of AG and α. This demonstrates that SaLoss obtains more faithful explanations in the majority of cases (Jacovi and Goldberg, 2020). For example in Ev.Inf., the BASELINE approach with α requires .25 fractions of tokens on average to observe

[14]We treat λ as a hyper-parameter tuned on the development set, where $\lambda \in$ {1e-2, 1e-3, 1e-4}.

	Metric	SST	AG	Ev.Inf.	MultiRc	SEMEVAL
	Random	.66	.67	.51	.44	.54
Baseline	α	.55	.43	.25	.40	.43
	$\mathbf{x}\nabla\mathbf{x}$	.65	.64	.42	.40	.55
	$\alpha\nabla\alpha$	.57	.52	.25	.38	.48
	I.G.	.63	.63	.42	.42	.50
SaLoss	α	**.42**†	.53	.14†	**.19**†	**.39**†
	$\mathbf{x}\nabla\mathbf{x}$	.61†	.59†	.38†	.30†	.51†
	$\alpha\nabla\alpha$	.48†	.50†	**.12**†	.24†	.41†
	I.G.	.61†	.57†	.33†	.33†	.45†

Table 3.10: Average fraction of tokens required to cause a decision flip across datasets and feature attribution metrics (lower is better). **Bold** denotes the best method in each dataset. † denotes a significant difference compared to Baseline using the same attribution metric (Wilcoxon Rank Sum, $p < .05$).

a decision flip compared to .14 with SaLoss (approximately 40 tokens less). We also observe that in MultiRc where $\boldsymbol{\alpha}$ is not the most effective feature attribution method with Baseline, with SaLoss it becomes the most effective. In fact, $\boldsymbol{\alpha}$ is the best performing feature attribution approach across most tasks and metrics using SaLoss, indicating the effectiveness of infusing salient information.

We also performed an analysis on the differences in Part-of-Speech (PoS) tags of the rationales selected by SaLoss and the Baseline, to obtain insights towards why rationales with SaLoss are shown to be more faithful to those from models trained without our proposed objective. In SST, we observe that SaLoss allocates more importance on adverbs and adjectives, which are considered important in sentiment analysis (Dragut and Fellbaum, 2014; Sharma et al., 2015). In Ev.Inf., we observed that SaLoss allocates importance to subordinating conjunction words such as *than*, which are indeed important for the task, which consists of inferring relationships (i.e. *higher than*). We thus hypothesise that SaLoss guides the model to other informative tokens, complementing the task specific information learned by the model.[15]

[15]We include a more extensive analysis in Appendix B.1.

Dataset	TopK			Contiguous		
	Baseline	SaLoss		Baseline	SaLoss	
		TextRank	Uniform		TextRank	Uniform
SST (20%)	.83 (.00)	**.87** (.00) †	.82 (.00)	.82 (.00)	**.83** (.00) †	.80 (.00)
AG (20%)	.92 (.00)	.92 (.00)	.92 (.00)	**.90** (.00)	.89 (.00)	.89 (.00)
Ev.Inf. (10%)	**.82** (.00)	.81 (.00)	.78 (.00)	**.79** (.00)	.78 (.00)	.78 (.00)
MultiRc (20%)	.75 (.00)	.75 (.00)	.75 (.00)	.70 (.00)	.67 (.00)	**.71** (.00)
SEMEVAL (20%)	.48 (.03)	**.53** (.01)†	.43 (.00)	.46 (.03)	**.47** (.01)†	.42 (.00)

Table 3.11: F1 macro on models trained with extracted rationales (using α) using FRESH for Baseline and SaLoss models. **Bold** denotes best performance in each dataset. † indicates that SaLoss rationales perform significantly better (t-test, $p < .05$).

FRESH Performance: We finally compare our SaLoss models with vanilla LMs (Baseline) on rationale extraction using FRESH (Jain et al., 2020), by measuring the predictive performance of the classifier trained on the extracted rationales. For completeness we also include an uninformative baseline for SaLoss, which comprise of a normalised uniform distribution over the input (i.e. all inputs are assigned the same salience score). For brevity, Table 3.11 presents results using α with TopK and Contiguous rationales.

Using TopK rationales, our approach significantly outperforms Baseline in 2 out of 5 datasets (t-test, $p < 0.05$), whilst achieving comparable predictive performance on the rest. For example in SST we observe a 3% increase in F1 using the same ratio of rationales. It is notable that in MultiRc, AG and Ev.Inf., performance of classifiers trained on rationales from both Baseline and SaLoss is comparable to that with full text (1-2% lower). We assume that this is due to the nature of the tasks, which likely do not require a large part of the input to reach high performance. This highlights the effectiveness of our approach, as a simple yet effective solution for improving explanation faithfulness.

Contiguous rationales extracted from models trained with SaLoss, obtain comparable performance to models without (Baseline). Additionally, results show that classifier performance does not reach those with TopK rationales. We can therefore assume that TopK rationales lead higher predictive performance in inherently faithful classifiers. It is encouraging to notice that in the datasets where performance is comparable with our approach (AG, Ev.Inf., MultiRc), it is likely due to reaching close to Full-Text performance. For example, classifier performance trained on Contiguous rationales from Baseline in SST is at .82 compared to

Example 1 **Data.:AG Id: test_239**

[Baseline]: NEW YORK (Reuters) - Shares of **Google Inc. will make their Nasdaq stock market** debut on Thursday after the year 's most anticipated initial public offering priced far below initial estimates , raising $1.67 billion .

[SaLoss (Ours)]: NEW YORK (Reuters) - Shares of Google Inc. will make their Nasdaq stock market debut on Thursday after the year 's most anticipated initial public offering **priced far below initial estimates , raising $1.67** billion .

[Topic]: Business

Example 2 **Data.:SST Id: test_78**

[Baseline]: If nothing else this **movie introduces a** promising unusual kind of psychological horror.

[SaLoss (Ours)]: If nothing else this movie introduces **a promising unusual** kind of psychological horror.

[Sentiment]: Positive

Example 3 **Data.:Ev.Inf. Id: 4118506_0**

[Baseline]: ... analgesics . **ABSTRACT.AIM : : The aim of this study is to evaluate the efficacy of fentanyl along with LA field infiltration in controlling pain and discomfort associated with CVC insertion . ABSTRACT**.SETTINGS AND DESIGN : :...

[SaLoss (Ours)]: ... ABSTRACT.RESULTS : : The median interquartile range pain score is worst for placebo group after LAI (5 [3 - 6]) and in the immediate postprocedure period (5 [4 - 5]) **which was significantly attenuated by addition of fentanyl (3.5 [2 - 5] and 3 [2 - 4]) (P = 0.009 and 0.001** respectively) ...

[Intervention || Comparator || Outcome]: Fentanyl || Normal saline || Pain score

[Relationship]: Significantly decreased

Table 3.12: True examples of extracted rationales from models using our proposed approach (SaLoss) and from models that do not (Baseline)

.83 with SaLoss rationales.

Results also suggest that our uninformative baseline (Uniform), reduces the faithfulness of rationales in most cases resulting in lower classifier performance. We hypothesise that in cases where performance is comparable with Baseline and SaLoss, it is due to the task being relatively easy and as such the loss function not impacting the faithfulness of rationales. We consider this direction as an interesting area for future work.

3.2.5 Qualitative Analysis

In Table 3.12 we present examples of extracted rationales from a model trained with our proposed objective (SaLoss) and without (Baseline) using $\alpha\nabla\alpha$, to gain further insights to complement the PoS analysis. For clarity we present rationales of Contiguous type.

In AG we observed similar performance between models trained with SaLoss and without. Example 1 illustrates such a case, where both models predicted correctly but attended to different parts of the input. Despite in different locations, both segments are closely associated with the label of "Business". Example 2 is an instance from the SST dataset, were the SaLoss rationale points to a phrase that is more associated with the task ("*a promising unusual*") compared to the Baseline. This also aligns with previous observations from the PoS analysis, that models trained with our proposed objective attend to more adjectives compared to Baseline. Example 3 considers an instance from the Ev.Inf. dataset, which shows that the model trained with SaLoss and Baseline attended to two different sections. In fact what we observed in agreement with the PoS analysis, is that models with SaLoss attend mostly to segments including words related to relationships, such as "*significantly attenuated*" in this particular example.

3.2.6 Comparing Salience Distributions

Table 3.13 presents the average fraction of tokens required to cause a prediction switch (decision flip), when training models with SaLoss and (1) TextRank; (2) Chisquared; (3) Tfidf. We observe that when models are regularised with TextRank scores, the feature attribution approaches result in a lower average fraction of tokens to cause a prediction switch compared to the other two salience functions. We also observe that Tfidf is comparable with TextRank in most cases, outperforming Chisquared. We hypothesise that Tfidf performs poorer than TextRank is due to the way these two approaches compute their "importance" scores. The first computes them globally, whilst the latter locally (at instance-level) which we assume is more beneficial for explanation faithfulness.

	Metric	SST	AG	Ev.Inf.	MultiRc	SEMEVAL
	Random	.66	.67	.51	.44	.54
TextRank	α	.42	.53	.14	.19	.39
	$\mathbf{x}\nabla\mathbf{x}$	.61	.59	.38	.30	.51
	$\alpha\nabla\alpha$	.48	.50	.12	.24	.41
	I.G.	.61	.57	.33	.33	.45
Chisquared	α	.49	.67	.29	.38	.44
	$\mathbf{x}\nabla\mathbf{x}$	.60	.59	.47	.34	.54
	$\alpha\nabla\alpha$	.61	.71	.28	.33	.49
	I.G.	.58	.56	.48	.38	.47
Tfidf	α	.47	.43	.20	.33	.48
	$\mathbf{x}\nabla\mathbf{x}$	.62	.57	.41	.36	.57
	$\alpha\nabla\alpha$	.50	.47	.20	.37	.58
	I.G.	.58	.56	.40	.38	.53

Table 3.13: Average fraction of tokens required to cause a decision flip across datasets and feature attribution metrics (lower is better).

3.2.7 Combining SaLoss with TaSc

We now examine if by combining SaLoss with TaSc (presented in Section 3.1), we can improve further post-hoc explanation faithfulness. We use Lin-TaSc (Section 3.1.2) to compute the non-contextualised scores s_{x_i} and subsequently scale the hidden representations $\mathbf{h}_i$ from the final layer of a transformer-based network. This can result in a less contextualised sequence representation $\mathbf{c}_i$, which in turn can improve post-hoc explanation faithfulness as empirically demonstrated in Section 3.1. Table 3.14 shows the average fraction of tokens required to cause a decision flip (lower is better) across datasets and feature attributions from models with: (1) No TaSc and no SaLoss (Baseline); (2) with only SaLoss (SaLoss); (3) with only TaSc (TaSc) and finally (4) when using both (SaLoss + TaSc).

We first observe that training models with SaLoss and TaSc together, results in the majority of cases to improved results compared to the Baseline (i.e. lower fraction of tokens to cause a decision flip). However, improvements in faithfulness are less compared to when using either SaLoss or TaSc independently (i.e. higher average fraction of tokens). Results also suggest, that using either TaSc or SaLoss we can obtain a lower fraction of tokens compared to the Baseline across any dataset or feature attribution method, with both performing comparably. For example, using α with SaLoss in SST results to .42 fraction of tokens on

	Metric	SST	AG	Ev.Inf.	MultiRc	SEMEVAL
	Random	.66	.67	.51	.44	.54
Baseline	α	.55	.43	.25	.40	.43
	$\mathbf{x}\nabla\mathbf{x}$	.65	.64	.42	.40	.55
	$\alpha\nabla\alpha$	.57	.52	.25	.38	.48
	I.G.	.63	.63	.42	.42	.50
SaLoss	α	.42	.53	.14	.19	.39
	$\mathbf{x}\nabla\mathbf{x}$	.61	.59	.38	.30	.51
	$\alpha\nabla\alpha$	.48	.50	.12	.24	.41
	I.G.	.61	.57	.33	.33	.45
TaSc	α	.45	.42	.15	.23	.43
	$\mathbf{x}\nabla\mathbf{x}$	.61	.55	.40	.25	.52
	$\alpha\nabla\alpha$	.48	.47	.19	.25	.41
	I.G.	.58	.50	.37	.34	.36
SaLoss + TaSc	α	.49	.65	.22	.47	.42
	$\mathbf{x}\nabla\mathbf{x}$	.62	.64	.37	.52	.51
	$\alpha\nabla\alpha$	.51	.60	.40	.51	.45
	I.G.	.44	.58	.37	.48	.36

Table 3.14: Average fraction of tokens required to cause a decision flip across datasets and feature attribution metrics (lower is better), when comparing Baseline against using: (1) only SaLoss; (2) only TaSc and (3) both SaLoss and TaSc.

average compared to .45 with TaSc, whilst using $\mathbf{x}\nabla\mathbf{x}$ with SaLoss in MultiRC leads to .30 compared to .25 with TaSc. We also observe a variation in the attention explanation performance of Baseline between Table 3.14 and results from Section 3.1. This is attributed to the fact that in Section 3.1, we did not use the internal attention mechanism of BERT to produce attention scores, but rather a standalone attention mechanism over the model's representations. This allowed us to perform a fairer comparison across the encoders we were comparing.

Using only TaSc appears to improve consistenly the faithfulness of explanations from both attention-based metrics (α, $\alpha\nabla\alpha$) when compared to SaLoss. For example in AG, SaLoss with α results to a higher fraction of tokens compared to the BaseLine, whilst TaSc to a lower fraction of tokens (.53 with SaLoss, .42 with TaSc and .43 with Baseline). This is expected, as TaSc targets specifically the final layer of the transformer-based models with non-contextualised information from the model. On the other hand, SaLoss targets the model

through the attention mechanism via the training phase of the model. Whilst both combined result to a deterioration in explanation faithfulness, compared to when both are deployed individually, a case can be made for when to use each. SaLoss can be used in cases where we cannot afford introducing additional parameters and we are content with the prior used to influence model learning and the hyperparameter tuning. On the contrary, TaSc is a more stable and robust component that does not necessarily require an informative prior or any additional tuning for its parameters.

3.3 Summary

This chapter presented two novel approaches for improving the faithfulness of post-hoc explanations extracted using feature attribution methods. Section 3.1 introduced TaSc, a family of three encoder-independent mechanisms that induce context-independent task-specific information to attention. Through an extensive series of experiments, we show that attention-based explanation faithfulness with TaSc is improved compared to without TaSc. Through a robustness analysis we showed that explanation quality remains more consistent with TaSc over instances in a datasets, whilst results hold across faithfulness evaluation metrics. We also show that attention-based explanations with TaSc outperform other popular feature attribution methods. Section 3.2 introduced SaLoss, an auxiliary objective using TextRank importance scores to shift the models attention to other a priori extracted informative distributions. Evaluating post-hoc explanation faithfulness using Fraction of Tokens, we have shown that LMs trained with SaLoss require on average a lower number of tokens to cause a prediction switch. This indicates that they are more faithful compared to models trained without SaLoss. Finally, models with SaLoss return rationales that lead to higher predictive performance with FRESH compared to models trained without SaLoss.

Chapter 4

Instance-Specific Rationalisation for NLP Models

4.1 Motivation

We have previously shown how feature scoring (i.e. attribution) methods, such as gradient and attention-based scores (Arras et al., 2016; Sundararajan et al., 2017; Jain and Wallace, 2019), are used to identify important (i.e. salient) segments of the input to subsequently extract them as rationales (Jain et al., 2020). We have also demonstrated in Section 3.1 how post-hoc explanation faithfulness can be improved. However, a single feature scoring method is typically applied across the whole dataset (i.e. globally). Previous research has shown that this might not be optimal for individual instances, resulting into less faithful explanations (Jacovi and Goldberg, 2020; Atanasova et al., 2020). Additionally, rationales are usually extracted using a pre-defined fixed length (i.e. the ratio of a rationale compared to the full input sequence) and type (i.e. top k terms or contiguous) globally. We hypothesise that using a fixed length or type for different instances could result into shorter (i.e. not sufficient for explaining a model's prediction) or longer than needed rationales reducing rationale faithfulness, whilst finding the explanation length is an open problem (Zhang et al., 2021). For example a pre-defined rationale length might not suffice for explaining a model's prediction (i.e. shorter than needed) or provide a larger number of tokens than needed resulting into reduced rationale faithfulness. Moreover to extract rationales, practitioners are

currently required to make assumptions for the rationale parameters (i.e. feature scoring method, length and type), whilst different choice of parameters might substantially affect the faithfulness of the rationales.

For this purpose, we propose a simple yet effective method that operates at instance-level and mitigates the a priori selection of a specific: (1) feature scoring method; (2) length and (3) type when extracting faithful rationales. Our proposed method is flexible and allows the automatic selection of some of these instance-specific parameters or all. Inspired by erasure methods, it functions by computing the difference between a model's output distributions obtained using the full input sequence and the input without the rationale respectively. We base this on the assumption that by removing important tokens from the sequence, we should observe large divergences in the model's predicted distribution (Nguyen, 2018b; Serrano and Smith, 2019; DeYoung et al., 2020) resulting into more faithful rationales (Atanasova et al., 2020; Chen and Ji, 2020).

This chapter is organised into five sections. Section 4.2 describes the methodology for computing instance-specific feature attribution, rationale length and types. Section 4.3 presents the experimental set-up and Section 4.4 discusses the results obtained. Finally, Section 4.7 summarises and concludes the contributions and findings of this Chapter.

4.2 Methodology

The aim is to address the *"one-size-fits-all"* ad-hoc approach of previous work on rationale extraction with feature scoring methods that typically extracts rationales using the same feature scoring method, length and type across all instances in a dataset. Inspired by word erasure approaches (Nguyen, 2018b; Serrano and Smith, 2019; DeYoung et al., 2020), we mask the tokens that constitute a rationale and record the difference δ in a model's output distribution by using the full text and the reduced input. Our main assumption is that a sufficiently faithful rationale is the one that will result into the largest δ (Atanasova et al., 2020; Chen and Ji, 2020; DeYoung et al., 2020). Following this assumption, we can extract rationales by selecting for each instance a specific (1) *feature scoring method*; (2) *length*; and (3) *type*. Similar to Jain et al. (2020), we consider two rationale types: (a) top k tokens ranked by a feature scoring method, treating each word in the input sequence independently

(TopK); and (b) Contiguous span of input tokens of length K with the highest overall score computed by a feature scoring method.

4.2.1 Instance-level Feature Scoring Selection

For computing at instance-level the feature scoring method, given a set of M feature scoring methods $\{\Omega_1, \ldots, \Omega_M\}$, we extract a rationale $\mathcal{R}$ as follows:

1. For each Ω_i in the set we compute input importance scores $\boldsymbol{\omega}_i = \Omega_i(\mathcal{M}, \mathbf{x}, \mathcal{Y})$;
2. We subsequently select the K highest scored tokens (TopK) or the highest K-gram (Contiguous) to form a rationale $\mathcal{R}_i$, where K is the rationale length;
3. For each rationale we compute the difference δ_i, between the reference model output (using full text input) and the model output having masked the rationale, such that:

$$\delta_i = \Delta(\mathcal{Y}, \mathcal{Y}_i^m) = \Delta(\mathcal{M}(\mathbf{x}), \mathcal{M}(\mathbf{x}_{\setminus \mathcal{R}_i}))$$

   where Δ is the function used to compute the difference between the two outputs;
4. We select the rationale $\mathcal{R}$ with the highest difference $\delta_{max} = max(\{\delta_1, \ldots, \delta_i, \ldots, \delta_M\})$.

For computing δ, we experiment with the following divergence metrics (Δ): (a) Kullback-Leibler (KL); (b) Jensen-Shannon divergence (JSD); (c) Perplexity (Perp.) and (d) Predicted Class Probability (ClassDiff). We describe the metrics in detail in Section 4.2.3.

4.2.2 Instance-level Rationale Length Selection

For computing at instance-level the rationale length k and extracting the rationale R using a single feature scoring method Ω, we propose the following steps:

1. Given Ω, we first compute input importance scores $\boldsymbol{\omega} = \Omega(\mathcal{M}, \mathbf{x}, \mathcal{Y})$;

2. We then iterate over the sequence such that $k = range(1, N)$, where N is the fixed, predefined rationale length and k the possible rationale length at the current iteration. We set N as the upper bound rationale length for our approach to make results comparable with fixed length rationales.

3. At each iteration we begin by masking the top k tokens (as indicated by ω) to form a candidate rationale $\mathcal{R}_k$. When using TopK we mask the k highest scored tokens, whilst with Contiguous we mask the highest scored k-gram;

4. We compute the difference δ_k between the reference model output $\mathcal{Y}$ and the model output having masked the candidate rationale $\mathcal{Y}_k^m = \mathcal{M}(\mathbf{x}_{\setminus \mathcal{R}_k})$;

5. We record every δ until $k = N$ and extract the rationale $\mathcal{R}$ with the highest difference $\delta_{max} = max(\{\delta_1, \ldots, \delta_k, \ldots, \delta_N\})$, where k at δ_{max} is the computed rationale length.[1]

In a similar way to selecting a feature scoring method, our approach can also be used to select between different rationale types (i.e. Contiguous or TopK) for each instance in the dataset.

Finally, our approach is flexible and can be easily modified to support selecting any of these parameters while keeping the rest fixed (i.e. feature scoring method, rationale length and rationale type) or by selecting any combination of them. An important benefit of our approach is that we extract rationales with different settings for each instance rather than using uniform settings globally (i.e. across the whole dataset), which we empirically demonstrate to be beneficial for faithfulness below.

4.2.3 Divergence metrics

To compute the value δ for how much $\mathcal{Y}^m$ differs from $\mathcal{Y}$, we consider four divergence metrics (Δ), also previously used in literature (Robnik-Šikonja and Kononenko, 2008; Jain and Wallace, 2019; Wiegreffe and Pinter, 2019):

[1] We also experimented with early stopping, whereby the difference between δ_k and the δ_{max} until k are under a specified threshold, however this resulted in reduced performance.

Kullback Leibler (KL): A non-symmetric divergence measure of how a particular distribution divergences from a reference distribution:

$$KL(\mathcal{Y}||\mathcal{Y}^*) = \mathcal{Y}(\log(\mathcal{Y} - \log(\mathcal{Y}^m)) \tag{4.1}$$

Jensen-Shannon (JSD): A symmetric divergence metric based on the KL divergence of two distributions from their mean:

$$JSD(\mathcal{Y}||\mathcal{Y}^m) = \frac{1}{2}(KL(\mathcal{Y}||\mu) + \frac{1}{2}(KL(\mathcal{Y}^m||\mu)) \tag{4.2}$$

where μ is the average distribution of $\mathcal{Y}$ and $\mathcal{Y}^m$.

Perplexity (Perp.): A measure of how well a model can predict a sample, where:

$$PERP(\mathcal{Y}||\mathcal{Y}^m) = \exp^{\mathcal{H}(\mathcal{Y},\mathcal{Y}^m)} \tag{4.3}$$

where we consider $\mathcal{Y}$ as the ground truth and $H(\mathcal{Y}||\mathcal{Y}^m)$ is the Cross Entropy Loss.

Class Difference (ClassDiff): The direct difference between the predicted class probability from the model with full text ($\mathbf{x}$) and the same class probability with reduced text ($\mathbf{x}_{\setminus\mathcal{R}}$) :

$$CLASSDIFF(\mathcal{Y}||\mathcal{Y}^m) = p(\hat{y}|\mathbf{x}) - p(\hat{y}|\mathbf{x}_{\setminus\mathcal{R}}) \tag{4.4}$$

where $\hat{y} = \arg\max(\mathcal{Y})$.

Data	$\|W\|$	C	Splits Train/Dev/Test	F1	N
SST	18	2	6,920 / 872 / 1,821	90.1 ± 0.2	20%
AG	36	4	102,000 / 18,000 / 7,600	93.5 ± 0.2	20%
Ev.Inf.	363	3	5,789 / 684 / 720	83.0 ± 1.6	10%
MultiRC	305	2	24,029 / 3,214 / 4,848	73.2 ± 1.7	20%

Table 4.1: Dataset statistics including average words at instance ($|W|$), number of classes (**C**), data splits, F1 macro performance and the fixed, pre-defined rationale ratio across all instances (N).

4.3 Experimental Setup

4.3.1 Datasets

For our experiments we use the following datasets, also previously used in Section 3.2 (details about the tasks in Section 3.2.3 and data characteristics in Table 4.1): (1) **SST**; (2) **AG**; (3) **Ev.Inf.** and (4) **MultiRC**.

4.3.2 Models

Similar to Jain et al. (2020), we use BERT (Devlin et al., 2019) for SST and AG; SciBERT (Beltagy et al., 2019) for Ev.Inf. and Roberta (Liu et al., 2019b) for MultiRc. Table 4.2 presents the hyper-parameters used to train the models across different datasets, along with F1 macro performance on the development set. Models where finetuned across 3 runs for 5 epochs. We implement our models using the Huggingface library (Wolf et al., 2019) and use default parameters of the AdamW optimiser apart from the learning rates. We use a linear scheduler with 10% of the steps in the first epoch as warmup steps. Experiments are run on a single Nvidia Tesla V100 GPU.

Dataset	Model	lr^m	lr^c	F1
SST	bert-base	1e-5	1e-4	90.7 ± 0.2
AG	bert-base	1e-5	1e-4	93.3 ± 0.0
Ev.Inf.	scibert	1e-5	1e-4	82.5 ± 0.9
MultiRC	roberta-base	1e-5	1e-4	76.3 ± 0.2

Table 4.2: Model and their hyper-parameters for each dataset, including learning rate for the model (lr^m) and the classifier layer (lr^c) and F1 macro scores on the development set across three runs.

4.3.3 Feature Scoring Methods

We use a random baseline and six other feature scoring methods to compute input importance scores, as described in Section 2.5.1 and similar to Jain et al. (2020) and Serrano and Smith (2019). Namely, we use: (1) attention ($\boldsymbol{\alpha}$); (2) scaled attention ($\boldsymbol{\alpha}\nabla\boldsymbol{\alpha}$); (3) InputXGrad ($\mathbf{x}\nabla\mathbf{x}$); (4) Integrated Gradients (**IG**); (5) **DeepLift** and finally (6) **LIME**.

4.3.4 Evaluating Explanation Faithfulness

We evaluate explanation faithfulness using the previously described approaches (see Section 2.6 for details): (1) F1-macro using the model predicted labels; (2) Normalised Sufficiency (NormSuff) and finally (3) Normalised Comprehensiveness (NormComp).

We do not conduct human experiments to evaluate explanation faithfulness since that is only relevant to explanation plausibility (i.e. how understandable by humans a rationale is (Jacovi and Goldberg, 2020)) and in practice faithfulness and plausibility do not correlate (Atanasova et al., 2020). Finally we do not compare with select-then-predict methods (Lei et al., 2016; Jain et al., 2020), as we are interested in faithfully explaining the model $\mathcal{M}$ and not forming inherently faithful classifiers.

4.3.5 Performance-Time Trade-off

Input erasure approaches typically require N forward passes to compute a rationale length when removing one token at a time. Albeit significantly faster to computing LIME scores,

selecting a rationale length at each instance in a dataset can be computationally expensive when we compute δ for every token, being similar to evaluating using fraction of tokens (Nguyen, 2018b; Serrano and Smith, 2019; Atanasova et al., 2020). This takes into consideration that we have to perform a forward pass for every token until we reach N tokens, for each feature attribution approach Ω.

Similar to Nguyen (2018b); Atanasova et al. (2020), we expedite this process when selecting a rationale length by "skipping" every $X\%$ of tokens. We use a 2% skip rate which led to a seven-fold reduction in the time required to compute rationales for datasets comprising of long sequences, such as MultiRc and EvInf, with comparable performance in faithfulness to the slower process of removing one token at a time. We examine the performance/skip-rate trade-off when (1) we do not use a skip-rate; (2) at 2% and (3) at 5% in Section 4.5.1.

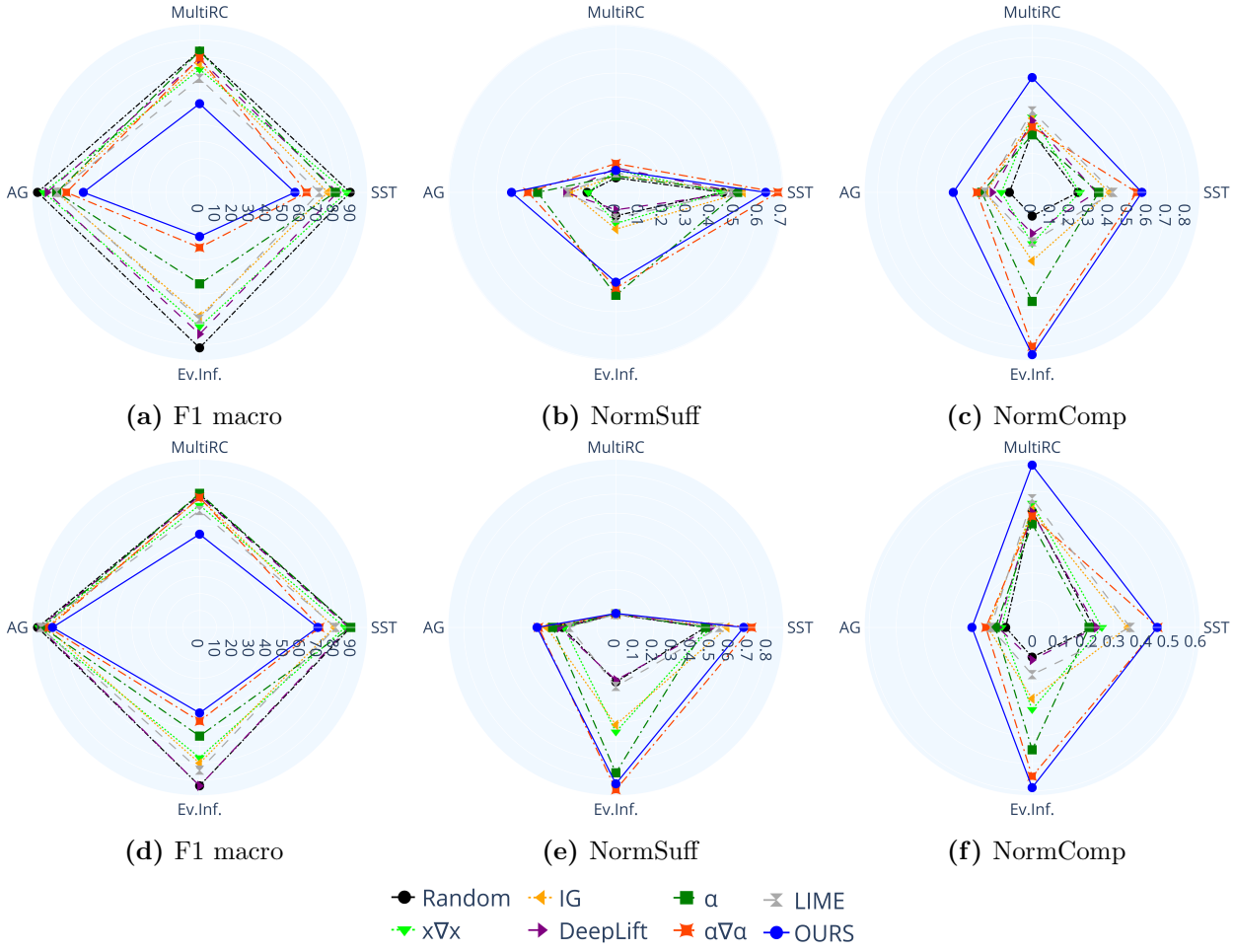
4.4 Results

4.4.1 Selecting Instance-specific Feature Scoring

Figure 4.1 compares the faithfulness of extracted rationales when using our proposed method for selecting an instance-specific feature scoring method (OURS) and our baselines, that use a single fixed pre-defined feature scoring method globally (i.e. across all instances in a dataset). We measure faithfulness using F1 macro (lower is better), mean NormSuff and mean NormComp (higher is better respectively). Results using both the TopK rationale type (sub-figures (a); (b) and (c)) and Contiguous (sub-figures (d); (e) and (f)).[2]

We first observe that results are similar across rationale types. As such, for clarity the remainder of this section will focus on describing results with TopK rationales. Overall, results demonstrate that rationales extracted with our proposed approach are highly sufficient and comprehensive. In fact, our approach results in more sufficient rationales against all single feature scoring methods in AG and is comparable with the best NormSuff scores in the remainder of the datasets. This suggests that even when rationales with our proposed method are not the most sufficient, they are consistently highly sufficient (i.e. rationales extracted

[2]Also for clarity, all results presented are using JSD for Δ. The other divergence functions performed comparably and we include a comparison between them in Section 4.5.2.



(a) F1 macro **(b)** NormSuff **(c)** NormComp

(d) F1 macro **(e)** NormSuff **(f)** NormComp

Figure 4.1: F1 macro (lower is better), mean NormSuff (higher is better) and mean NormComp (higher is better), when using any single feature scoring method across all instances in a dataset and our proposed method of selecting a feature scoring method for each instance (OURS) for TopK rationale types (sub-figures (a), (b), (c)) and Contiguous (sub-figures (d), (e), (f)).

with our approach are significantly more sufficient than fixed, pre-defined feature scoring methods in 18 out of 24 test cases). Compared to our six baselines, the rationales extracted with our approach are significantly more comprehensive across all four datasets (Wilcoxon Rank Sum, $p < .05$). Additionally, the larger drops in F1 macro performance demonstrate that rationales extracted with our proposed approach are more necessary for a model to make a prediction compared to a globally used, pre-defined feature scoring approach.

Our results strengthen the hypothesis that whilst some feature scoring methods are better

%		x∇x	IG	DeepLift	LIME	α	$\alpha\nabla\alpha$	Avg.
TopK	SST	15.8	16.3	15.7	16.9	15.5	16.6	16.1
	AG	14.5	16.5	15.2	16.3	16.3	16.3	15.8
	Ev.Inf.	7.7	7.7	8.4	7.4	6.6	7.0	7.5
	MultiRC	14.1	14.6	16.1	13.4	15.9	16.0	15.0
Contiguous	SST	15.5	15.6	15.4	15.9	14.7	15.7	15.5
	AG	14.0	15.7	14.9	15.0	14.5	15.1	14.9
	Ev.Inf.	7.2	6.9	7.6	7.3	6.6	6.9	7.1
	MultiRC	14.0	15.1	15.9	13.6	16.1	16.1	15.1

Table 4.3: Average instance-specific rationale lengths (as a percentage %) computed using JSD (as Δ), across instances for TopK and Contiguous rationale types.

than others globally, they might not be optimal for all instances in a dataset (Jacovi and Goldberg, 2020) and our approach helps mitigate that. Similar to Atanasova et al. (2020), we observe that the faithfulness performance of single feature scoring methods varies across datasets. For example, LIME returns more comprehensive rationales than $\alpha\nabla\alpha$ in MultiRC, however is outperformed by the latter in SST. By returning consistently highly comprehensive and sufficient rationales, our propose method helps reducing the variability in faithfulness performance observed when using any single feature scoring method across datasets.

4.4.2 Selecting Instance-specific Rationale Length

Table 4.4 shows the Relative Improvement (R.I.) ratio in mean NormSuff and NormComp (>1.0 is better) between rationales extracted using a fixed pre-defined length (see N in Table 4.1) and rationales extracted using our method with instance-specific length across feature scoring methods and datasets. Overall, rationales extracted using our approach are on average shorter than fixed length rationales (see Table 4.3). Specifically, rationale length drops from 20% to 16% on average in SST, AG; from 20% to 15% in MultiRc and from 10% to 7% in Ev.Inf..

F1 macro scores indicate that rationales extracted with our proposed approach are comparably faithful to the longer pre-defined rationales, despite being shorter (R.I. in the majority of cases between 0.9 and 1.1). NormSuff scores indicate that our shorter on average ra-

	Feature Attribution	F1 macro				NormSuff				NormComp			
		SST	MultiRc	AG	Ev.Inf.	SST	MultiRc	AG	Ev.Inf.	SST	MultiRc	AG	Ev.Inf.
TopK	DeepLift	1.0	1.0	1.0	1.0	0.9	0.8	0.8	1.1	0.8	1.1	1.0	1.0
	LIME	1.1	0.9	1.0	1.0	1.0	0.7	0.9	0.9	0.9	1.1	1.0	1.0
	α	1.1	1.0	1.0	0.8	0.9	0.9	0.7	0.8	0.8	1.1	0.9	1.2
	$\alpha\nabla\alpha$	1.0	1.0	1.0	0.8	0.9	0.9	0.8	0.9	1.0	1.1	0.9	1.0
	IG	1.0	0.9	1.0	1.0	0.9	0.9	0.8	0.9	0.9	1.1	1.0	1.1
	$\mathbf{x}\nabla\mathbf{x}$	1.0	0.9	1.0	0.9	1.0	0.8	0.7	0.8	0.9	1.1	0.9	1.2
Contiguous	DeepLift	1.0	0.9	1.0	0.9	0.9	0.9	0.8	1.2	0.9	1.1	1.3	1.5
	LIME	1.0	0.8	1.0	0.9	0.9	0.7	0.8	0.9	1.0	1.1	1.2	1.3
	α	1.0	1.0	1.0	0.8	0.9	0.9	0.7	0.9	0.7	1.1	1.0	1.2
	$\alpha\nabla\alpha$	1.0	0.9	1.0	0.8	0.9	0.8	0.8	0.9	1.0	1.1	1.1	1.1
	IG	1.0	0.9	1.0	0.9	0.9	0.8	0.8	1.0	1.0	1.2	1.2	1.4
	$\mathbf{x}\nabla\mathbf{x}$	1.0	0.9	1.0	0.9	0.9	0.8	0.7	1.0	1.0	1.1	1.0	1.3

Table 4.4: Relative Improvement (R.I.) ratios for F1 macro, mean NormSuff and mean NormComp between fixed length rationales (see N in Table 4.1) extracted using our method and rationales with instance-specific length (>1.0 is better).

tionales are overall less but comparably sufficient with longer, fixed-length rationales. For example with SST rationales with instance-specific length are 0.9-1.0 times less sufficient that rationale with pre-defined length. We find this particularly evident in datasets such as MultiRc and Ev.Inf., where our rationales are on average 4-5% shorter (approximately 15 tokens shorter on average for α in MultiRc) but still retain comparable sufficiency, while in some cases improving it (e.g. 1.2 R.I. in Ev.Inf. with DeepLift).

We also note that rationales extracted with instance-specific length are more comprehensive in most cases, despite being shorter on average compared to fixed-length rationales. For example in Ev.Inf., Contiguous rationales with I.G. are 1.4 times more comprehensive when we select their length at instance-level. Results also indicate that using our proposed method benefits more Contiguous rationales compared to TopK for comprehensiveness, leading to increased R.I. in the majority of cases. Overall, findings support our initial hypothesis that in certain cases a rationale with longer than needed length might contain unnecessary information and adversely impact its comprehensiveness.

Type	Len	Feat	F1 macro				NormSuff				NormComp			
			SST	**MultiRc**	**AG**	**Ev.Inf.**	**SST**	**MultiRc**	**AG**	**Ev.Inf.**	**SST**	**MultiRc**	**AG**	**Ev.Inf.**
TopK	Fix	Fix	63.26	67.33	78.80	32.59	.68	**.12**	.37	.43	.54	.42	.28	.80
	I-L	Fix	64.41	62.74	79.52	25.26	.61	.11	.30	.37	.52	.46	.27	.82
	Fix	I-L	**56.40**	52.40	**68.80**	26.10	.63	.09	**.44**	.38	**.57**	.59	**.41**	.84
	I-L	I-L	57.20	48.80	69.90	21.70	.59	.07	.38	.36	.55	.62	.39	.86
Contiguous	Fix	Fix	70.89	68.96	89.18	55.32	**.71**	.07	.41	**.85**	.46	.47	.17	.55
	I-L	Fix	68.80	57.07	87.70	45.69	.63	.06	.33	.78	.47	.54	.19	.62
	Fix	I-L	69.80	55.00	86.90	50.50	.67	.07	.42	.82	.46	.60	.22	.59
	I-L	I-L	66.40	45.10	85.30	40.70	.61	.05	.33	.76	.48	.65	.24	.67
I-L	I-L	I-L	56.70	**38.00**	69.60	**19.80**	.60	.06	.39	.49	**.57**	**.69**	**.41**	**.88**

Table 4.5: F1 macro, Mean NormSuff and NormComp scores when we select at instance-level (I-L) a combination of the: (1) rationale length (Len); (2) feature scoring method (Feat.); and (3) rationale type (Type). {Type}-Fix-Fix and {Type}-I-L-Fix values are from the highest scoring feature scoring method (see Figure 4.1). **Bold** values denote the highest performing combination in column-wise (higher is better).

4.4.3 Selecting Instance-specific Feature Scoring, Length and Type

Table 4.5 shows F1 macro, mean NormSuff and NormComp scores when using our proposed method to select at instance-level (I-L) a combination of: (1) the feature scoring method (Feat); (2) the rationale length (Len); and (3) the rationale type (Type). For comparison, we also show scores of the best performing fixed (Fix) feature scoring function, rationale type and length (see Figure 4.1).

Selecting at instance-level all rationale settings results in lower F1 macro performance compared to any combination across two datasets (MultiRc and Ev.Inf.), whilst being comparable with the rest. For example with MultiRc, F1 performance drops to just 38.00 when we select all parameters compared to 45.1 with the second best (7 F1 point difference). Where comparable, we observe that differences are marginal (e.g. 0.30 F1 macro difference in SST with the best). This highlights the efficacy of our approach in extracting rationales that are necessary for a model to make a prediction, without requiring any a priori assumptions about any of the rationale parameters.

We then observe that the highest NormSuff scores across three datasets (SST, MRc,

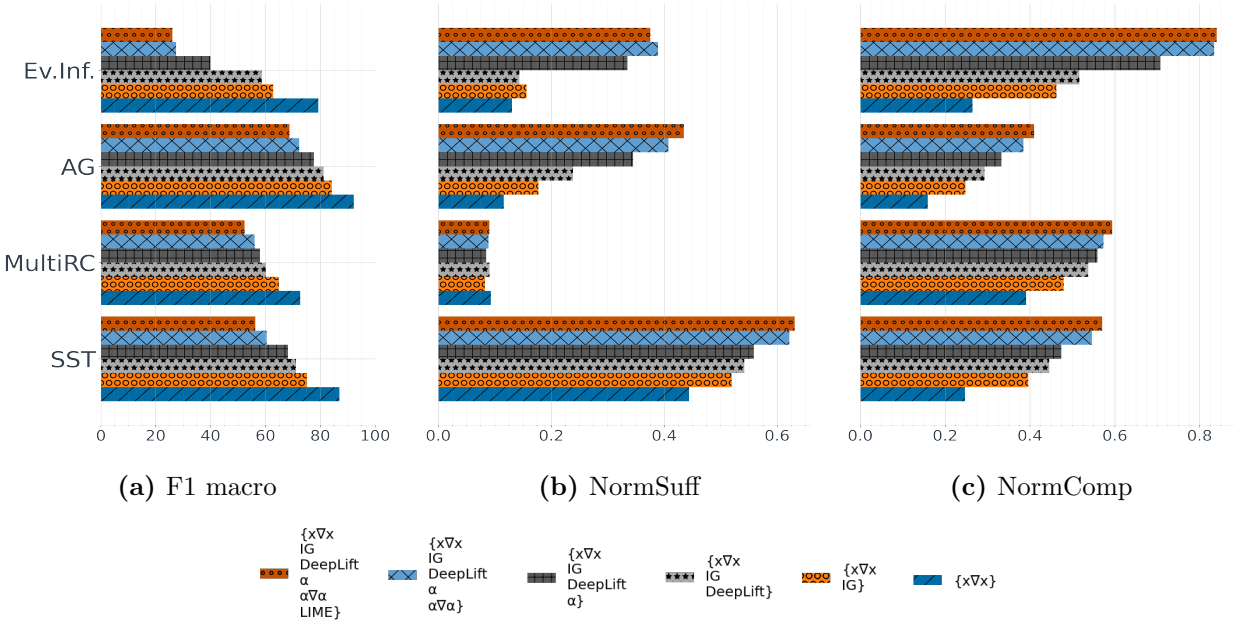
EvInf), are from the best performing fixed scoring method with fixed length and rationale type. Additionally, the best performing combination of our proposed approach for sufficiency is when we only select the feature scoring method keeping the length and type fixed. This combination results in the highest NormSuff scores in AG (.44 with TopK type compared to .42, which is the second best with Contiguous) and competitive NormSuff scores with the highest scoring combination (e.g. .82 in Ev.Inf. and Contiguous compared to .85). We assume that using combinations which include instance-specific lengths do not perform as well for sufficiency due to the shorter rationale length, which we have previously shown to partially degrade rationale sufficiency.

Finally, our results demonstrate that we obtain highly comprehensive rationales when selecting at instance level all parameters (Feat. + Len + Type) using our approach. In fact, this results in higher NormComp scores compared to any other setting combination across all datasets. For example in MultiRc., selecting all parameters results in a NormComp score of .69 which is .22 units higher than the rationales extracted with fixed feature scoring method and length and type. This highlights the efficacy of our approach in extracting highly comprehensive rationales, without requiring strong a priori assumptions about rationale parameters.

4.4.4 Ablation Study

We finally perform an ablation study to examine the behavior and effectiveness of our approach by sequentially removing one feature scoring method at a time to measure changes in F1 macro, NormSuff and NormComp. The intuition is that we should observe drops in faithfulness scores when removing feature attribution methods for our approach to be effective (i.e. we should extract more faithful rationales when having more feature scoring options to choose from). Figure 4.2 shows the results.

We first observe that removing one feature scoring method at a time results in increases in F1 macro (lower is better) and drops in NormComp scores (higher is better). This demonstrate that the faithfulness of the rationales extracted with our approach deteriorates as the number of feature scoring methods becomes smaller highlighting the efficacy of our proposed approach. For example, in Ev.Inf. by removing $\alpha\nabla\alpha$ results in a drop of .14 in mean NormComp (.84 when including $\alpha\nabla\alpha$ compared to .70 without it). On the other hand, we also



(a) F1 macro **(b)** NormSuff **(c)** NormComp

Figure 4.2: F1 macro (lower is better), mean NormSuff and mean NormComp (higher is better), when extracting rationales with our approach given decreasing numbers of feature scoring methods.

observe that our method can still benefit from feature scoring methods that achieve low NormComp scores when used standalone, resulting in improvements in comprehensiveness and drops in F1 macro (e.g. α in SST). This indicates that our approach steadily improves rationale faithfulness for model's predictions given a larger pool of available feature scoring methods.

Results show a deterioration in NormSuff scores as the number of feature scoring methods becomes smaller, showing that our method results in more sufficient rationales when presented with a larger list of available feature scoring methods in the majority of the datasets. We hypothesise that this is not true for MultiRC due to the already low NormSuff scores of the rationales (e.g. no more than 0.12). By using all six feature scoring methods, our approach produces highly sufficient rationales and is comparable to the set achieved the highest sufficiency. For example in Ev.Inf. using all feature scoring methods results to a NormSuff score of approximately .38 compared to the highest scoring feature scoring set (all except LIME) and the lowest scoring ($\mathbf{x\nabla x}$) which achieved .39 and .15 respectively.

		@Token	@2%		@5%	
		(s)	(s)	R.I.	(s)	R.I.
TopK	SST	0.05	0.05	1.0	0.05	1.0
	AG	0.29	0.29	1.0	0.15	1.9
	Ev.Inf.	1.99	0.26	7.7	0.11	18.1
	MultiRC	3.07	0.51	6.0	0.21	14.6
Contiguous	SST	0.06	0.06	1.0	0.06	1.0
	AG	0.37	0.38	1.0	0.19	1.9
	Ev.Inf.	2.59	0.33	7.8	0.13	19.9
	MultiRC	3.72	0.65	5.7	0.26	14.3

Table 4.6: Average time taken (s) to extract a rationales of instance-specific length per instance, when computing δ at: (1) each token (@Token); (2) at every 2% (@ 2%) and at every 5% (@ 5%), where lower time is better. We also denote relative improvements (R.I.) where higher is better.

We also tested different combinations of feature scoring methods with similar observations. Finally, we experimented with doubling the upper bound of the rationale length (from N to $2 \times N$) for both fixed length rationales and our proposed approach. Our approach still yielded more comprehensive rationales compared to the fixed-length ones that were also highly sufficient (results included in Section 4.5.3).

4.5 Quantitative Analysis

4.5.1 Reducing Time for Computing Instance-Specific Length

In Table 4.6, we present the average time taken (in seconds (s)) to extract a rationale of instance-specific length for each instance, when computing δ at: (1) each token (@Token); (2) at every 2% (@ 2%) and at every 5% (@ 5%) using JSD. We observe that in datasets with a short average length of instance when moving from @Token to @2% does not reduce time, but results in significant reductions in computations with MultiRc (∼6x R.I.) and Evinf (∼8x R.I.). As expected, these are further reduced when reducing the granularity to searching @5%, with AG recording a ∼2x R.I., MultiRC ∼14.5 R.I. and Ev.Inf. ∼19% R.I. However alongside time improvements we also need to evaluate how faithfulness is impacted.

		@Token				@2%				@5%			
		F1	**Suff.**	**Comp.**	**I.S.L.** (%)	**F1**	**Suff.**	**Comp.**	**I.S.L.** (%)	**F1**	**Suff.**	**Comp.**	**I.S.L.** (%)
TopK	SST	57.22	0.59	0.55	17.15	57.22	0.59	0.55	17.15	57.34	0.59	0.55	17.23
	AG	69.90	0.37	0.39	17.05	69.93	0.37	0.39	17.06	70.12	0.38	0.39	17.52
	Ev.Inf.	21.09	0.35	0.87	6.52	21.74	0.36	0.86	7.25	23.22	0.37	0.84	8.28
	MultiRC	47.63	0.07	0.63	13.59	48.76	0.07	0.62	14.31	49.92	0.08	0.61	15.37
Contiguous	SST	66.43	0.61	0.48	16.01	66.43	0.61	0.48	16.01	66.48	0.61	0.48	16.21
	AG	85.27	0.33	0.24	15.04	85.27	0.33	0.24	15.06	85.42	0.34	0.24	16.00
	Ev.Inf.	37.03	0.68	0.70	6.12	40.66	0.76	0.67	7.08	46.20	0.84	0.62	8.37
	MultiRC	43.38	0.05	0.66	12.29	45.12	0.05	0.65	13.22	47.87	0.06	0.63	14.42

Table 4.7: F1 macro (lower is better), NormSuff (higher is better) and NormComp (higher is better) for our rationales with instance-specific length (I.S.L.) and feature scoring method at each instance, when computing δ at: (1) each token (@Token); (2) at every 2% (@ 2%) and at every 5% (@ 5%). We also include average rationale lengths for helping with the analysis.

In Table 4.7 we present the faithfulness performance as we reduce the granularity of our search. Results suggest that by reducing granularity, NormComp scores reduce whilst F1 macro performance increase, suggesting a reduction in faithfulness. However, we observe that moving from @Token to @2% this reduction is negligible considering the significantly improved computational times. However, as expected moving from @Token to @5% performance degrades rapidly and as such 2% seems like a more appropriate step to consider. Unsurprisingly, with increasing step-size we observe increases in the computed rationale lengths, which leads to an increase in sufficiency.

Combining feature scoring rankings: We considered further reducing our computation time by merging importance scores from all feature scoring methods. The intuition is that we obtain a combined ranking and avoid selecting the best feature scoring method at each instance and computing a rationale length for all feature scoring methods. We attempted this by averaging the normalised importance scores for each sequence from all the feature scoring methods, however as expected results where not comparable (63.2 average F1 macro compared to 54.4) with our proposed approach or even our best performing baseline.

	Δ	NormSuff	NormComp	F1
TopK	JSD	0.38	0.64	46.03
	KL	0.38	0.63	46.40
	ClassDiff	0.38	0.63	45.02
	Perp.	0.35	0.55	52.22
Contiguous	JSD	0.38	0.64	46.03
	KL	0.38	0.63	46.40
	ClassDiff	0.38	0.64	45.02
	Perp.	0.35	0.55	52.22

Table 4.8: NormSuff (higher is better), NormComp (higher is better) and F1 macro (lower is better) when using different divergence metrics to select the rationale length and feature scoring method(Δ)

4.5.2 Performance across Divergence Metrics

We now compare the effectiveness of divergence metrics in computing a rationale with instance-specific length and selecting the best feature scoring method at instance level. Table 4.8 presents F1 macro (lower is better), NormSuff and NormComp (higher is better) macro scores for our proposed instance-specific length rationales from the best feature scoring method at instance level.

Results demonstrate that all divergence metrics perform comparably with the exception of perplexity. The remainder of the metrics result in similar scores for NormSuff, NormComp and F1 macro, with JSD and ClassDiff having a slight edge over KL.

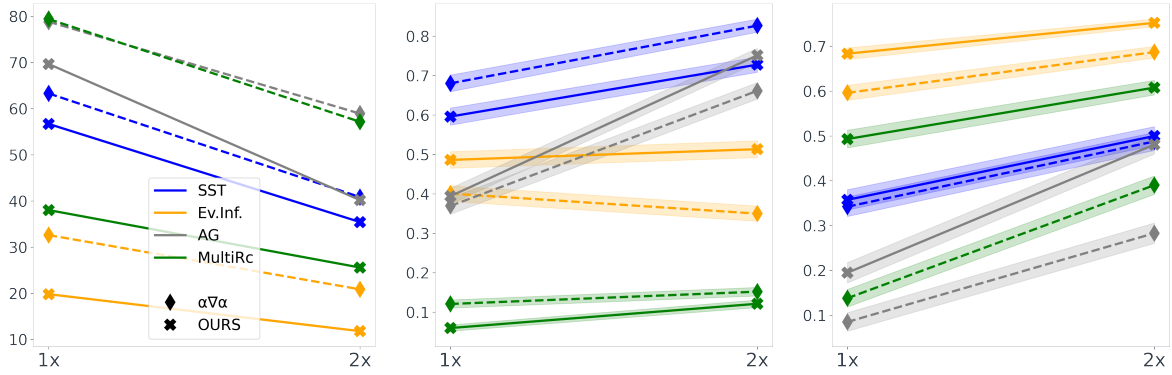
4.5.3 Effects of Increasing the Rationale Length Upper Bound N

We hypothesise that the information a rationale holds, increases with increasing rationale lengths similar to Jain et al. (2020). We therefore evaluate the effectiveness of our approach, when doubling the upper-bound of the maximum allowed rationale length N (see §4.2). We assume that this should result into better rationale comprehensiveness and sufficiency.

In Table 4.9 we present the computed rationale lengths when we double N. As we observe

		x∇x	IG	DeepLift	LIME	α	$\alpha\nabla\alpha$	Avg.
TopK	SST	29.5	30.5	30.3	31.0	28.4	30.3	30.0
	AG	32.4	34.8	32.4	32.8	33.6	32.9	33.2
	Ev.Inf.	14.8	15.3	16.9	15.3	11.9	14.0	14.7
	MultiRC	26.9	28.5	31.7	25.4	31.1	32.1	29.3
Contiguous	SST	28.3	28.7	28.8	28.4	27.0	29.3	28.4
	AG	31.7	32.2	31.8	30.6	31.4	31.4	31.5
	Ev.Inf.	13.4	13.6	15.3	14.4	12.5	13.0	13.7
	MultiRC	26.4	28.6	30.0	25.1	30.1	30.0	28.4

Table 4.9: Average instance-specific rationale lengths computed using JSD, across instances for TopK and Contiguous rationale types when we double N to $2 \times N$.



(a) F1 **(b)** NormSuff **(c)** NormComp

Figure 4.3: F1 macro (lower is better), mean NormSuff and NormComp scores (higher is better) extracted rationales when using $2 \times N$.

our rationales are still shorter compared to $N \times 2$, with certain cases resulting in significant reductions. For example in Ev.Inf. with α, contiguous rationales extracted with our approach are on average 7.5% shorter than fixed length rationales (approximately 27 tokens shorter). In MultiRc the differences are even more apparent, with an average reduction in rationale length of approximately 12%. This translates to 25 tokens less when forming a rationale with our proposed approach.

Figure 4.3 shows F1 macro, NormSuff and NormComp scores of rationales, when we increase N (1x) to $2 \times N$ (2x). For brevity we show results from rationales extracted with our best performing method (OURS, i.e. using instance-level feature scoring method, rationale

length and type) and when using $\alpha\nabla\alpha$ across the entire dataset with fixed length. Results highlight that our approach successfully scales with increasing rationale lengths, resulting in highly sufficient and more comprehensive rationales with an average length shorter than the $2 \times N$ upper-bound. For example, with Ev.Inf. rationales extracted with our proposed approach have slightly improved sufficiency from N to $2 \times N$ (0.49 to 0.51 respectively). In contrast, extracting rationales with a priori defined $2 \times N$ length and $\alpha\nabla\alpha$ results to a decrease in NormSuff scores (0.37 compared to 0.40). It is notable that improvements are observed, regardless of our proposed approach resulting in rationales which are on average 6% shorter (translating to approximately 22 tokens less for Ev.Inf. to form a rationale).

4.6 Qualitative Analysis

Table 4.10 shows examples of the qualitative comparison between our approach (Ours) for selecting at instance-level (I-L) a combination of the: (1) rationale length (LEN); (2) feature scoring method (FEAT against our baseline of fixed-length rationales from a fixed feature scoring method.

Concise rationales: Example 1 presents an instance from AG. Our approach extracts a rationale that is six tokens shorter than the one with fixed length while also achieving a higher NormComp score. However, the fixed length rationale scores higher in NormSuff. We can assume from this that sufficiency positively correlates with rationale length.

Error analysis: Our assumption is that if a model makes a wrong prediction, we should be able to extract the rationale that better demonstrates what led to a wrong prediction. Example 2 shows an instance from Ev.Inf., where the model has wrongly predicted that "Lenke scores at 12 months" have 'increased significantly' instead of the correct 'no significant difference'. Surprisingly, both rationales recorded maximum scores (1.0) in NormSuff and NormComp. We observe that the correct answer is included in the fixed length rationale, however the model made a wrong prediction. On the contrary, our rationale highlights something directly related to its prediction.

Example 3 presents an instance from SST, where the fixed-length rationale and the

Example 1 **Data.:AG Id: test_4614**

[Fixed-Len + α]: ... game last **Friday night will stand , the CFL announced yesterday. While** a review ...

[I-L-Len + α (Ours)]: ... game last Friday night will stand , **the CFL announced yesterday**. While a review ...

[Predicted Topic || True Topic]: Decreased significantly || Decreased significantly

---

Example 2 **Data.:Ev.Inf. Id: 3162205_2**

[Fixed-Len + $\alpha\nabla\alpha$]: ... computed tomography (3D - CT **) scans . ABSTRACT.RESULTS : The control sides treated with an autograft showed significantly better Lenke scores than the study sides treated with β - CPP at 3 and 6 months postoperatively , but there was no difference between the two sides at 12 months** . The fusion ..

[I-L-Len + $\alpha\nabla\alpha$ (Ours)]: ... computed tomography (3D - CT) scans **. ABSTRACT.RESULTS : The control sides treated with an autograft showed significantly better Lenke scores** than the study sides treated with β - CPP at 3 ...

[Predicted Relationship || True Relationship]: Increased significantly || No significant difference

---

Example 3 **Data.:SST Id: test_694**

[Fixed-Len + α]: ... Frontal **is the antidote for Soderbergh** fans who think he s gone too commercial ...

[I-L-Len + I-L-Feat (Ours)]: ... Frontal is the antidote for Soderbergh fans who think he **s gone too commercial** ...

[Predicted Sentiment || True Sentiment]: Negative || Positive

---

Example 4 **Data.:SST Id: test_1039**

[Fixed-Len + α]: It 's just **incredibly** dull.

[I-L-Len + I-L-Feat (Ours)]: It 's just **incredibly dull**.

[Predicted Sentiment || True Sentiment]: Negative || Negative

Table 4.10: Examples when using our approach (Ours) to select at instance-level (I-L) a combination of the: (1) rationale length (LEN); (2) feature scoring method (FEAT) against our baseline of fixed-length rationales from a fixed feature scoring method.

instance-specific rationale attend at different sections of the text. Our rationale scored lower for NormSuff, however we observe that it aligns more closely with the predicted sentiment.

When using a fixed pre-defined length is not sufficient: Example 4 presents a different scenario, where the fixed-length rationale for SST is at 20% whilst the upper bound N for our rationale is at 40%. The intuition is that in certain cases a fixed rationale length might not be sufficient for all instances to explain a prediction. We argue that our approach highlighted something more informative for the task ("incredibly dull" compared to "incredibly"), due to removing the restriction of a pre-defined fixed length.

4.7 Summary

We have proposed a simple yet effective approach for selecting at instance-level: (1) a feature scoring method; (2) a rationale length; and (3) a rationale type. We empirically demonstrated that rationales extracted with our approach are significantly more comprehensive and highly sufficient, while being shorter compared to rationales extracted with a fixed feature scoring method, length and type. We also show that our approach can scale effectively, given a larger number of feature attribution approaches and an increased rationale length upper bound. Finally, through a qualitative analysis we found that rationales extracted using our proposed method can be useful for error analysis.

Chapter 5

An Empirical Analysis of Faithfulness in Out-of-Domain Settings

5.1 Motivation

As described in Section 2.5, two popular methods for extracting explanations are feature attribution (i.e. *post-hoc* explanation methods) or inherently faithful classifiers (i.e. *select-then-predict* models). In the previous chapters, we first attempted to improve post-hoc explanation faithfulness and proposed an approach for extracting instance-specific rationales. Similar to previous studies, we have previously evaluated explanation faithfulness on in-domain settings (i.e. the train and test data come from the same distribution). However, when deploying models in real-world applications, inference might be performed on data from a different distribution, i.e. out-of-domain (Desai and Durrett, 2020; Ovadia et al., 2019). This can create implications when the extracted explanations are used for assisting human decision making. Whilst we are aware of the limitations of current state-of-the-art models in out-of-domain predictive performance (Hendrycks et al., 2020), to the best of our knowledge, how faithful out-of-domain post-hoc explanations are has yet to be explored. Similarly, we are not aware on the generalization of inherently faithful select-then-predict models in out-of-domain settings.

Inspired by this, we conduct an extensive empirical study to examine the faithfulness of

five feature attribution approaches and the generalizability of two select-then-predict models in out-of-domain settings across six dataset pairs. We hypothesize that similar to model predictive performance, post-hoc explanation faithfulness reduces in out-of-domain settings and that select-then-predict performance degrades.

This chapter is organised into five sections. Section 5.2 describes the approaches we evaluate for extracting rationales. Section 5.3 presents the experimental setup and how we evaluate post-hoc explanation faithfulness and select-then-predict performance in out-of-domain settings. Section 5.4 discusses the results obtained. Finally, Section 5.5 summarises and concludes the contributions and findings of this chapter.

5.2 Extracting Rationales

The aim of this chapter, is to evaluate the faithfulness of rationales in out-of-domain settings. To extract rationales we employ two methods: (1) feature attribution methods for post-hoc explanations (Section 5.2.1) and (2) select-then-predict models (Section 5.2.2), which we expand on below. Our hypothesis is that post-hoc explanations extracted from the same domain that a model was trained on should be more faithful than rationales extracted from different domain using the same model. Similarly, we expect select-then-predict model predictive performance to drop when evaluating in out-of-domain settings.

5.2.1 Feature Attribution Methods

We employ a pre-trained BERT-base and fine-tune it on in-domain training data. We then extract post-hoc rationales for both the in-domain test-set and two out-of-domain test-sets. We use a random baseline and five other feature scoring methods to compute input importance scores, as described in Section 2.5.1. Namely, we use: (1) attention ($\boldsymbol{\alpha}$); (2) scaled attention ($\boldsymbol{\alpha}\nabla\boldsymbol{\alpha}$); (3) InputXGrad ($\mathbf{x}\nabla\mathbf{x}$); (4) Integrated Gradients (**IG**) and finally (5) **DeepLift**.

Dataset	C	Splits
SST	2	6,920 / 872 / 1,821
IMDB	2	20,000 / 2,500 / 2,500
Yelp	2	476,000 / 84,000 / 38,000
AmazDigiMu	3	122,552 / 21,627 / 25,444
AmazPantry	3	99,423/ 17,546 / 20,642
AmazInstr	3	167,145 / 29,497 / 34,702

Table 5.1: Dataset statistics with number of classes (**C**) and train/development/test **splits**.

5.2.2 Select-then-Predict Models

We use two select-then-predict models (see Section 2.5.2 for details): (1) **HardKuma** and (2) **FRESH**. We use BERT-base for the extraction and classification components of FRESH similar to Jain et al. (2020). However, for HardKuma we opt using a bi-LSTM (Hochreiter and Schmidhuber, 1997) as it provides comparable or improved performance over BERT variants (Guerreiro and Martins, 2021), even after hyperparameter tuning. For model details and hyper-parameters see Section 5.3.2.

5.3 Experimental Setup

5.3.1 Datasets

For evaluating out-of-domain model explanations, we consider the **SST** (Socher et al., 2013) and **IMDB** (Maas et al., 2011) datasets as in Section 3.1.3 and the following (see Table 5.1 for details):

Yelp: Yelp polarity review texts. Similar to Zhang et al. (2015) we construct a binary classification task to predict a polarity label by considering one and two stars as negative, and three and four stars as positive.

Amazon Reviews: We form 3-way classification tasks by predicting the sentiment (negative, neutral, positive) of Amazon product reviews across 3 item categories: (1) Digital Music (**AmazDigiMu**); (2) Pantry (**AmazPantry**); and (3) Musical Instruments (**AmazInstr**) (Ni et al., 2019).

5.3.2 Models and Hyperparameters

For feature attributions: We use BERT-base with pre-trained weights from the Huggingface library (Wolf et al., 2019). We use the AdamW optimizer (Loshchilov and Hutter, 2017) with an initial learning rate of $1e-5$ for fine-tuning BERT and $1e-4$ for the fully-connected classification layer. We train our models for 3 epochs using a linear scheduler, with 10% of the data in the first epoch as warm-up. We also use a grad-norm of 1 and select the model with the lowest loss on the development set. All models are trained across 5 random seeds and we report the average and standard deviation. We present their test-set performance in Table 5.2.

For FRESH: For the rationale extractor, we use the same model for extracting rationales with feature attributions. For the classifier (trained only on the extracted rationales), we also use BERT-base with the same optimizer configuration and scheduler warm-up steps. We also use a grad-norm of 1 and select the model with the lowest loss on the development set. We train across 5 random seeds for 5 epochs.

For HardKuma: We use the 300-dimensional pre-trained GloVe embeddings from the 840B release (Pennington et al., 2014) as word representations and keep them frozen during training. The rationale extractor (which generates the rationale mask z) is a 200-d bi-directional LSTM layer (bi-LSTM) (Hochreiter and Schmidhuber, 1997) similar to Bastings et al. (2019) and Guerreiro and Martins (2021). We use the Adam optimizer (Kingma and Ba, 2014) for all models with a learning rate between $1e-5$ and $1e-4$ and a weight decay of $1e-5$. We also enforce a grad-norm of 5 and train for 20 epochs across 5 random seeds. Similar to Guerreiro and Martins (2021) we select the model with the highest F1-macro score on the development set and find that tuning the Lagrangian relaxation algorithm parameters beneficial to model predictive performance. We also attempted training HardKuma models

Trained On	Tested On	BERT-base		bi-LSTM	
		F1	ECE	F1	ECE
SST	SST	90.1 (0.3)	4.4 (0.7)	81.7 (0.9)	3.2 (0.7)
	IMDB	84.3 (0.6)	7.1 (0.6)	71.9 (0.9)	4.9 (2.8)
	Yelp	87.9 (2.3)	4.2 (2.3)	68.7 (3.2)	5.8 (5.1)
IMDB	IMDB	91.1 (0.4)	4.7 (0.6)	87.4 (0.9)	4.7 (1.8)
	SST	85.8 (2.0)	5.8 (0.8)	77.5 (2.0)	6.2 (1.4)
	Yelp	91.0 (1.2)	0.9 (0.2)	41.0 (5.3)	39.4 (7.3)
Yelp	Yelp	96.9 (0.1)	2.2 (0.1)	96.0 (0.0)	0.5 (0.2)
	SST	86.8 (1.7)	8.5 (0.9)	80.4 (0.8)	1.9 (0.7)
	IMDB	88.6 (0.3)	7.9 (0.6)	84.5 (1.0)	5.0 (1.3)
AmazDigiMu	AmazDigiMu	70.6 (0.9)	2.3 (0.1)	67.6 (0.3)	0.5 (0.1)
	AmazInstr	61.2 (1.8)	5.4 (0.2)	54.2 (1.1)	2.6 (0.6)
	AmazPantry	64.6 (1.0)	4.3 (0.4)	55.3 (0.4)	1.9 (0.5)
AmazPantry	AmazPantry	70.2 (1.1)	3.8 (0.4)	67.9 (0.4)	0.7 (0.4)
	AmazDigiMu	59.5 (0.7)	3.2 (0.5)	50.9 (1.9)	1.9 (0.6)
	AmazInstr	64.5 (2.6)	4.9 (0.9)	55.9 (2.2)	2.8 (0.9)
AmazInstr	AmazInstr	71.5 (0.4)	3.9 (0.5)	67.2 (0.7)	1.2 (0.4)
	AmazDigiMu	61.3 (0.3)	3.2 (0.2)	54.3 (1.4)	1.1 (0.1)
	AmazPantry	68.2 (0.7)	4.1 (0.5)	61.1 (1.5)	1.5 (0.6)

Table 5.2: F1 macro performance and Expected Calibration Error (ECE) (five runs) with standard deviation, of full-text BERT-base and bi-LSTM models.

with BERT-base, similar to Jain et al. (2020), however we found performance to be at best comparable with our LSTM variant, as in Guerreiro and Martins (2021), even after hyperparameter tuning.

Full-text LSTM: In Table 5.2 we also present for reference the performance of a 200-dimensional bi-LSTM classifier trained on full-text. We train the full-text LSTM for 20 epochs across 5 random seeds and select the model with the highest F1-macro performance on the development set. We use the Adam optimizer with a learning rate of $1e-3$ and $1e-5$ weight decay. We report predictive performance and ECE scores on the test-set.

All experiments are run on a single NVIDIA Tesla V100 GPU.

5.3.3 Evaluating Out-of-Domain Explanations

Post-hoc Explanations: We evaluate explanation faithfulness using the previously described approaches (see Section 2.6 for details): (1) Normalised Sufficiency (NormSuff) and (2) Normalised Comprehensiveness (NormComp). To measure sufficiency and comprehensiveness across different explanation lengths we compute the "Area Over the Perturbation Curve" (AOPC) following DeYoung et al. (2020). We therefore compute and report the average normalized sufficiency and comprehensiveness scores when keeping (for sufficiency) or masking (for comprehensiveness) the top 2%, 10%, 20% and 50% of tokens extracted by an importance attribution function.

We omit from our evaluation the Remove-and-Retrain method (Madsen et al., 2021a) as it requires model retraining. Whilst this could be applicable for in-domain experiments where retraining is important, in this work we evaluate explanation faithfulness in zero-shot out-of-domain settings.

Select-then-Predict Models: We first train select-then-predict models in-domain and then measure their predictive performance on the in-domain test set and on two out-of-domain test-sets (Jain et al., 2020; Guerreiro and Martins, 2021). Our out-of-domain evaluation is performed without re-training (zero-shot). Similar to full-text trained models, we expect that predictive performance deteriorates out-of-domain. However, we assume that explanations from a select-then-predict model should generalize better in out-of-domain settings when the predictive performance approaches that of the full-text trained model.

We do not conduct human experiments to evaluate explanation faithfulness, since that is only relevant to explanation plausibility (i.e. how intuitive to humans a rationale is (Jacovi and Goldberg, 2020)) and in practice faithfulness and plausibility do not correlate (Atanasova et al., 2020).

Train	Test	Full-text F1	AOPC NormSuff						AOPC NormComp					
			Rand	$\alpha\nabla\alpha$	α	DeepLift	$x\nabla x$	IG	Rand	$\alpha\nabla\alpha$	α	DeepLift	$x\nabla x$	IG
SST	SST	90.1	.38	.51	.42	.42	.40	.41	.19	.39	.22	.25	.26	.26
	IMDB	84.3	.31	.53	.39	.32	.31	.32	.23	.54	.34	.27	.27	.28
	Yelp	87.9	.32	.56	.40	.35	.33	.34	.21	.48	.28	.24	.24	.25
IMDB	IMDB	91.1	.32	.55	.46	.36	.36	.36	.16	.48	.31	.25	.23	.24
	SST	85.8	.24	.35	.28	.28	.27	.27	.27	.46	.32	.33	.33	.33
	Yelp	91.0	.35	.48	.41	.36	.36	.36	.21	.45	.32	.26	.26	.26
Yelp	Yelp	96.9	.23	.32	.31	.29	.24	.25	.12	.20	.14	.16	.15	.16
	SST	86.8	.41	.45	.43	.44	.41	.41	.17	.24	.18	.21	.22	.22
	IMDB	88.6	.18	.34	.32	.25	.22	.22	.19	.34	.29	.23	.23	.24
AmazDigiMu	AmazDigiMu	70.6	.34	.56	.34	.31	.41	.39	.13	.32	.14	.10	.16	.17
	AmazInstr	61.2	.29	.54	.32	.31	.33	.32	.19	.47	.23	.19	.22	.23
	AmazPantry	64.6	.33	.55	.33	.31	.37	.36	.21	.46	.22	.17	.23	.25
AmazPantry	AmazPantry	70.2	.25	.46	.36	.19	.28	.27	.20	.42	.31	.15	.25	.25
	AmazDigiMu	59.5	.24	.47	.37	.19	.27	.26	.19	.41	.32	.15	.23	.24
	AmazInstr	64.5	.17	.42	.30	.15	.20	.20	.24	.52	.40	.23	.30	.30
AmazInstr	AmazInstr	71.5	.16	.34	.18	.21	.18	.17	.26	.52	.26	.29	.28	.29
	AmazDigiMu	61.3	.21	.38	.21	.22	.24	.22	.23	.46	.20	.22	.24	.25
	AmazPantry	68.2	.22	.39	.21	.23	.24	.23	.27	.51	.22	.25	.27	.29

Table 5.3: AOPC Normalized Sufficiency and Comprehensiveness (higher is better) in-domain and out-of-domain for five feature attribution approaches and a random attribution baseline.

5.4 Results

5.4.1 Post-hoc Explanation Faithfulness

Table 5.3 presents the normalized comprehensiveness and sufficiency scores for post-hoc explanations on in-domain and out-of-domain test-sets, using five feature attribution methods and a random baseline. For reference, we include the averaged F1 performance across 5 random seeds, of a BERT-base model finetuned on the full text and evaluated in- and out-of-domain (Full-text F1).

In-domain results show that feature attribution performance varies largely across datasets.

This is in line with the findings of Atanasova et al. (2020) and Madsen et al. (2021a) when masking rationales (i.e. comprehensiveness). We find the only exception to be $\alpha\nabla\alpha$, which consistently achieves the highest comprehensiveness and sufficiency scores across all in-domain datasets. For example $\alpha\nabla\alpha$ evaluated on in-domain AmazDigiMu, results in sufficiency of 0.56 compared to the second best of 0.39 with IG.

Contrary to our expectations, results show that post-hoc explanation sufficiency and comprehensiveness are in many cases higher in out-of-domain test-sets compared to in-domain. For example using DeepLift, comprehensiveness for the in-domain test-set in Yelp (0.16) is lower compared to the out-of-domain test-sets (0.21 for SST and 0.23 for IMDB). This is also observed when measuring sufficiency with $\alpha\nabla\alpha$, scoring 0.32 when tested in-domain on Yelp and 0.45 for the out-of-domain SST test-set. We hypothesize that this is due to a low lexical and semantic overlap with the training set, when in out-of-domain settings. We assume that this therefore leads to the model assigning higher weights to the shared "lexicon", making it more likely to be important for a prediction.

Apart from increased sufficiency and comprehensiveness scores in out-of-domain post-hoc explanations, we also observe increased scores obtained by our random baseline. In fact, the random baseline outperforms several feature attribution approaches in certain cases in out-of-domain settings. As an example, consider the case where the model has been trained on AmazInstr and tested on AmazPantry. Our random baseline achieves a comprehensiveness score of 0.27 while α, DeepLift, $x\nabla x$ perform similarly or lower (0.22, 0.25 and 0.27 respectively). Similarly, using a model trained on Yelp and tested on SST, the random baseline produces equally sufficient rationales to $x\nabla x$ and IG, with all of them achieving 0.41 normalized sufficiency. A glaring exception to this pattern is $\alpha\nabla\alpha$, which consistently outperforms both the random baseline and all other feature attribution approaches in in- and out-of-domain settings, suggesting that it produces the more faithful explanations. For example with out-of-domain AmazPantry test data, using a model trained on AmazInstr results in sufficiency scores of 0.39 with $\alpha\nabla\alpha$. This is a 0.15 point increase compared to the second best ($x\nabla x$ with 0.24).

We recommend considering *a feature attribution for producing faithful explanations out-of-domain, if it only scores above a baseline random attribution. We suggest that the higher the deviation from the random baseline, the more faithful an explanation is.*

Train	Test	Full-text	HardKuma	
		F1	F1	L (%)
SST	SST	81.7 (0.9)	77.6 (1.4)	56.8 (26.2)
	IMDB	71.9 (0.9)	**65.7**(15.1)	39.5 (33.5)
	Yelp	68.7 (3.2)	**67.7**(11.6)	32.7 (30.7)
IMDB	IMDB	87.4 (0.9)	82.0 (0.6)	1.9 (0.2)
	SST	77.5 (2.0)	73.6 (2.2)	16.8 (2.7)
	Yelp	41.0 (5.3)	**47.2**(5.8)	3.1 (2.0)
Yelp	Yelp	96.0 (0.0)	92.4 (0.3)	7.4 (0.7)
	SST	80.4 (0.8)	72.4 (0.8)	14.1 (1.2)
	IMDB	84.5 (1.0)	73.3 (3.5)	4.7 (0.7)
AmazDigiMu	AmazDigiMu	67.6 (0.3)	66.8 (0.5)	18.4 (0.5)
	AmazInstr	54.2 (1.1)	**53.3**(1.2)	25.8 (6.1)
	AmazPantry	55.3 (0.4)	**54.7**(1.4)	27.8 (3.6)
AmazPantry	AmazPantry	67.9 (0.4)	66.6 (0.5)	18.9 (1.1)
	AmazDigiMu	50.9 (1.9)	**51.0**(0.6)	11.2 (3.3)
	AmazInstr	55.9 (2.2)	**57.4**(1.2)	18.2 (1.3)
AmazInstr	AmazInstr	67.2 (0.7)	**66.7**(0.8)	19.2 (1.5)
	AmazDigiMu	54.3 (1.4)	**53.7**(1.2)	13.9 (2.9)
	AmazPantry	61.1 (1.5)	**59.5**(1.4)	24.4 (2.8)

Table 5.4: F1 macro performance (five runs) with standard deviation for HardKuma models and the selected rationale length (L). **Bold** denotes no significant difference between HardKuma and Full-text (t-test; $p > 0.05$).

5.4.2 Select-then-predict Model Performance

HardKuma: Table 5.4 presents the F1-macro performance of HardKuma models (Bastings et al., 2019) and the average rationale lengths (the ratio of the selected tokens compared to the length of the entire sequence) selected by the model. For reference, we also include the predictive performance of a full-text trained bi-LSTM. Results are averaged across 5 runs including standard deviations in brackets.

As expected, predictive performance of HardKuma models degrades when evaluated on out-of-domain data. Surprisingly, though, we find that their performance is not significantly different (t-test; p-value > 0.05) to that of the full-text LSTM in 9 out of the 12 out-of-domain

dataset pairs. For example, by evaluating the out-of-domain performance of a HardKuma model trained on AmazDigiMu on the AmazPantry test-set, we record on average a score of 54.3 F1 compared to 55.3 with an LSTM classifier trained on full text. We also observe that HardKuma models trained on SST and IMDB generalize comparably to models trained on full-text when evaluated on Yelp, however the opposite does not apply. Our assumption is that HardKuma models trained on Yelp, learn more domain-specific information due to the large training corpus (when compared to training on IMDB and SST) so they fail to generalize well out-of-domain.

Results also show, that *the length of rationales selected by HardKuma models depend on the source domain, i.e. training HardKuma on a dataset which favors shorter rationales, leads to also selecting shorter rationales out-of-domain.*[1] For example, in-domain test-set explanation lengths are on average 56.8% of the full-text input length for SST. In comparison, training a model on Yelp and evaluating on SST results in rationale lengths of 14.1%. We observe that *in certain cases, HardKuma models maintain the number of words, not the ratio to the sequence* in out-of-domain settings. For example, in-domain Yelp test-set rationales are about 11 tokens long that is the similar to the length selected when evaluating on IMDB using a model trained on Yelp. This is also observed where in-domain AmazInstr test-set rationales are on average 5 tokens long, which is the same rationale length when evaluating on AmazDigiMu using a model trained on AmazInstr.

In general, our findings show that in the majority of cases, using HardKuma in out-of-domain data leads to comparable performance with their full-text model counterparts. This suggests that *HardKuma models can be used in out-of-domain settings, without significant sacrifices in predictive performance whilst also offering faithful rationales.*

FRESH: Table 5.5 shows the averaged F1-macro performance across 5 random seeds for FRESH classifiers on in- and out-of-domain using TopK rationales.[2] We also include the a priori defined rationale length in parentheses and the predictive performance of the Full-Text model for reference. When evaluating out-of-domain, we use the rationale length of the dataset we evaluate on. This makes FRESH experiments comparable with those of

[1]We also hypothesize that the short rationale lengths (e.g. 2% , ≈6 tokens in IMDB), indicates a more coherent dataset and a simlper task (e.g. when comparing to SST or Yelp).

[2]For clarity we include standard deviations and Contiguous results in Appendix C.3

Train	Test	Full-Text	F1				
			$\alpha\nabla\alpha$	α	DeepLift	$x\nabla x$	IG
SST (20%)	SST	90.1 (0.3)	87.7 (0.4)	81.1 (1.0)	84.4 (0.7)	76.3 (0.5)	76.8 (0.3)
	IMDB	84.3 (0.6)	81.8 (0.2)	52.6 (2.1)	64.0 (2.1)	55.0 (1.7)	56.3 (0.4)
	Yelp	87.9 (2.3)	**88.1**(0.0)	72.6 (4.0)	75.4 (2.3)	59.6 (3.8)	63.9 (1.1)
IMDB (2%)	IMDB	91.1 (0.4)	87.9 (0.2)	80.4 (0.9)	87.2 (0.4)	59.8 (0.2)	59.7 (0.6)
	SST	85.8 (2.0)	80.9 (0.5)	71.8 (1.0)	70.1 (0.5)	69.6 (0.5)	70.7 (1.7)
	Yelp	91.0 (1.2)	87.8 (0.1)	82.0 (0.2)	79.4 (1.4)	69.0 (0.6)	69.1 (0.4)
Yelp (10%)	Yelp	96.9 (0.1)	94.0 (0.0)	90.4 (0.2)	93.6 (0.3)	70.5 (0.2)	71.9 (0.1)
	SST	86.8 (1.7)	59.3 (0.6)	69.8 (1.1)	67.2 (1.5)	67.7 (0.5)	69.3 (0.8)
	IMDB	88.6 (0.3)	78.0 (0.4)	64.5 (0.3)	66.6 (0.5)	53.0 (0.4)	55.8 (0.1)
AmazDigiMu (20%)	AmazDigiMu	70.6 (0.9)	66.1 (1.8)	63.4 (1.0)	**65.8**(2.6)	51.9 (2.0)	65.8 (2.6)
	AmazInstr	61.2 (1.8)	**58.0**(0.8)	**57.2**(1.2)	**57.4**(1.2)	46.0 (0.6)	57.2 (1.2)
	AmazPantry	64.6 (1.0)	59.1 (0.3)	56.5 (1.2)	56.5 (1.7)	44.8 (0.8)	44.8 (0.8)
AmazPantry (20%)	AmazPantry	70.2 (1.1)	67.3 (0.5)	62.6 (1.0)	67.2 (0.0)	48.6 (1.7)	48.7 (2.7)
	AmazDigiMu	59.5 (0.7)	**57.7**(0.6)	54.6 (0.9)	56.2 (0.0)	41.2 (0.4)	57.7 (0.6)
	AmazInstr	64.5 (2.6)	**63.8**(0.4)	58.0 (1.9)	**63.6**(0.2)	40.1 (1.1)	40.3 (2.5)
AmazInstr (20%)	AmazInstr	71.5 (0.4)	69.8 (0.3)	62.1 (2.3)	69.7 (0.3)	45.6 (4.7)	48.6 (2.7)
	AmazDigiMu	61.3 (0.3)	**60.0**(0.7)	53.2 (1.7)	57.8 (0.4)	43.8 (3.3)	60.0 (0.7)
	AmazPantry	68.2 (0.7)	64.5 (0.7)	56.3 (1.9)	63.1 (0.3)	44.6 (3.9)	47.6 (2.6)

Table 5.5: Average F1 macro performance of FRESH models (with standard deviation across five runs) with the a priori defined rationale length in the brackets and TopK rationales. **Bold** denotes no significant difference between FRESH and Full-text (t-test; $p > 0.05$).

HardKuma.

We first observe that in-domain predictive performance varies across feature attribution approaches with attention-based metrics ($\alpha\nabla\alpha$, α) outperforming the gradient-based ones ($x\nabla x$, IG), largely agreeing with Jain et al. (2020). We also find that $\alpha\nabla\alpha$ and DeepLift are the feature attribution approaches that lead to the highest predictive performance across all datasets.

As we initially hypothesized, performance of FRESH generally degrades when testing on out-of-domain data similarly to the behavior of models trained using the full text. The only exceptions are when using $x\nabla x$ and IG in IMDB. We argue that this is due to these feature

attribution methods not being able to identify the appropriate tokens relevant to the task using a rationale length 2% of the original input. Increasing the rationale length to 20% (SST) and 10% (Yelp) also increases the performance. Results also suggest that $\alpha\nabla\alpha$ and DeepLift outperform the rest of the feature attributions, with $\alpha\nabla\alpha$ being the best performing one in the majority of cases. In fact when using $\alpha\nabla\alpha$ or DeepLift, the out-of-domain performance of FRESH is not significantly different to that of models trained on full text (t-test; p-value > 0.05) in 5 cases. For example, a FRESH model trained on AmazPantry and evaluated on AmazInstr records 63.6 F1 macro (using DeepLift) compared to 64.5 obtained by a full-text model. However, this does not apply to the other feature attribution methods (α; $x\nabla x$; IG).

To better understand this behavior, we conduct a correlation analysis between the importance rankings using any single feature attribution from (1) a model trained on the same domain with the evaluation data; and (2) a model trained on a different domain (out-of-domain trained model). High correlations suggest that if a feature attribution from an out-of-domain trained model produces similar importance distributions with that of an in-domain model, it will also lead to high predictive performance out-of-domain. Contrary to our initial assumption we found that the lower the correlation, the higher the predictive performance with FRESH. Results show low correlations when using $\alpha\nabla\alpha$ and DeepLift (highest FRESH performance). Surprisingly, IG and $x\nabla x$ (lowest FRESH performance) showed consistently strong correlations across all dataset pairs. Thus, we conclude that lower correlation scores indicate lower attachment to spurious correlations learned during training. We expand our discussion and show results for the correlation analysis in Appendix C.1.

Our findings therefore suggest that *using FRESH in out-of-domain settings, can result to comparable performance with a model trained on full-text. However this highly depends on the choice of the feature attribution method.*

HardKuma vs. FRESH: We observe that HardKuma models are not significantly different compared to models trained on the full text in out-of-domain settings in more cases, when compared to FRESH (9 out of 12 and 5 out of 12 respectively). However, *FRESH with $\alpha\nabla\alpha$ or DeepLift records higher predictive performance compared to HardKuma models (both in- and out-of-domain) in all cases.* We attribute this to the underlying model architectures, as FRESH uses BERT and HardKuma a bi-LSTM. As we discussed in §5.2.2, we attempted using BERT for HardKuma models in the extractor and classifier similar to Jain et al. (2020).

Train	Test FRESH	ρ Sufficiency	ρ Comprehen.
SST	SST	**0.97**	0.15
	IMDB	0.36	0.21
	Yelp	**0.90**	0.56
IMDB	IMDB	0.69	0.87
	SST	0.65	0.23
	Yelp	**0.92**	**0.92**
Yelp	Yelp	0.82	0.55
	SST	-0.67	-0.67
	IMDB	0.87	0.56
AmazDigiMu	AmazDigiMu	-0.11	0.22
	AmazInstr	0.23	0.69
	AmazPantry	0.11	0.11
AmazPantry	AmazPantry	0.16	0.16
	AmazDigiMu	0.05	0.41
	AmazInstr	0.16	0.16
AmazInstr	AmazInstr	0.79	0.55
	AmazDigiMu	0.24	0.67
	AmazPantry	0.21	0.20

Table 5.6: Spearman's ranking correlation (ρ) between FRESH performance and comprehensiveness, sufficiency across all feature attribution approaches. **Bold** denotes statistically significant (p-value $\leqslant 0.05$) correlations.

However, the performance of HardKuma with BERT is at most comparable to when using a bi-LSTM similar to findings of Guerreiro and Martins (2021).

5.4.3 Correlation between Post-hoc Explanation Faithfulness and FRESH Performance

We hypothesize that a feature attribution with high scores for sufficiency and comprehensiveness, should extract rationales that result in high FRESH predictive performance. We expect that if our hypothesis is valid, faithfulness scores can serve as early indicators of FRESH performance, both on in-domain and out-of-domain settings.

Table 5.6 shows the Spearman's ranking correlation (ρ) between FRESH F1 performance

	$\mathcal{M}$ Trained On	Example
(1)	AmazInstr (ID)	Work great and sound good
	AmazDigiMu	Work great and sound good
	AmazPantry	Work great and sound good
(2)	AmazPantry (ID)	Delicious and at a good price . would recommend .
	AmazDigiMu	Delicious and at a good price . would recommend .
	AmazInstr	Delicious and at a good price . would recommend .
(3)	SST (ID)	A painfully funny ode to bad behavior
	IMDB	A painfully funny ode to bad behavior
	Yelp	A painfully funny ode to bad behavior
(4)	Yelp (ID)	The kouign - amann is so amazing ... must taste to appreciate .
	SST	The kouign - amann is so amazing ... must taste to appreciate .
	IMDB	The kouign - amann is so amazing ... must taste to appreciate .

Table 5.7: True examples of highlights with $\alpha\nabla\alpha$ using a model trained on data from the same distribution as the example(ID; with blue highlights) and two models trained on a different dataset (with red highlights).

(see Table 5.5) and comprehensiveness and sufficiency (see Table 5.3). Correlation is computed using all feature scoring methods for each dataset pair. Results show that only 4 cases achieve statistically significant correlations (p-value < 0.05) with only 3 out-of-domain and mostly between sufficiency and FRESH performance. We do not observe high correlations with comprehensiveness which is expected, as comprehensiveness evaluated the rationale's influence towards a model's prediction. Our findings refute our initial hypothesis and suggest that *there is no clear correlation across all cases, between post-hoc explanation faithfulness and FRESH predictive performance. Therefore, sufficiency and comprehensiveness scores cannot be used as early indicators of FRESH predictive performance.*

5.4.4 Qualitative Analysis

In Table 5.7 we present examples from a qualitative analysis we performed, to understand better the performance out-of-domain of post-hoc explanations. Rows with highlights in blue are from a model trained in the same domain as the presented example (ID), whilst those with red are from models trained in a different domain. Importance scores computed using scaled attention ($\nabla\alpha\nabla$).

From Example (1) we can observe that models trained on two closely related tasks (AmazInstr and AmazDigiMu) are able to attend to the phrase "sound good". On the contrary, the model trained on AmazPantry which has not encountered such phrases focused only on "Work great" as expected. Similarly, observing Example (2) from the AmazPantry dataset, the in-domain model focused on a domain-specific word "delicious". On the other hand, the other two models trained on music-related tasks focus on more generic terms such as "good" and "would recommend". On Example (3) the model trained on Yelp focuses mostly on the word "behavior", a term we consider more relevant to restaurant reviews rather than movie reviews. In comparison, the other models which are both trained on movie reviews focus both on the term "funny". Whilst as expected, in Example (4) again the two movie-review models focus on more generic terms (i.e. "amazing") compared to "must taste" as in the model trained on Yelp.

Overall results show that models applied to a different domain (than that they were trained for), extract as rationales features that are mostly present within the domain they were trained for. This partly explains the performance of out-of-domain FRESH classifiers, where our assumption is that a model's inability to generalise to other domains is based on it latching on to spurious features from the training dataset (Adebayo et al., 2020).

5.5 Summary

We conducted an extensive empirical study to assess the faithfulness of post-hoc explanations (i.e. using feature attribution approaches) and performance of select-then-predict (i.e. inherently faithful) models in out-of-domain settings. Our findings highlight, that using sufficiency and comprehensiveness to evaluate post-hoc explanation faithfulness out-of-domain

can be misleading. To address this issue, we suggest using a random attribution as reference for a more reliable evaluation. We also show that select-then-predict models, which are inherently faithful, perform surprisingly well in out-of-domain settings. Despite performance degradation, in many cases their performance is comparable to those of full-text trained models.

Chapter 6

Conclusion and Final Remarks

This thesis addressed challenges related to extracting and evaluating faithful explanations in NLP. This chapter concludes this thesis, by summarising the findings and contributions presented throughout and finally indicates possible directions for future work.

6.1 Summary of Thesis

Chapter 2 first presented popular NLP task setups used, different neural text encoders and attention mechanisms. Following this prerequisite knowledge, we introduced model interpretability by showing how it is defined by previous studies and the definition adopted by this work, which is focused on faithful rationale extraction. We consider and describe two popular approaches for rationale extraction: post-hoc faithful explanations and inherently faithful select-then-predict models. We then reviewed approaches in literature for extracting explanations and evaluating their faithfulness. Finally, we described how past studies tackled improving explanations and their limitations.

Chapter 3 presented two novel approaches for improving the faithfulness of post-hoc explanations. In Section 3.1, our first proposed approach attempts to improve the faithfulness of attention-based explanations. As such, we introduced a family of encoder-independent mechanisms (TaSc) that scale the attention weights to reduce contextualisation. This is achieved by computing a non-contextualised score using the word embeddings. We then use

that score alongside attention weights, to compute the sequence representation. Evaluation on a battery of experiments for explanation faithfulness, showed that our proposed mechanisms are consistently more faithful compared to post-hoc explanations extracted without TaSc. The second part of this chapter (Section 3.2), proposes an auxiliary loss function for guiding the multi-head attention mechanism during training to be close to salient information extracted using TextRank. Experiments on explanation faithfulness across five datasets, show that models trained with our proposed method consistently provide more faithful explanations across four different feature attribution methods compared to vanilla BERT. We also show that extracting rationales from models trained with SaLoss, results to higher predictive performance compared to rationales extracted from models without SaLoss when using FRESH.

In Chapter 4, we propose a novel method for selecting each instance in a dataset a specific feature attribution, rationale length and type. Inspired by word erasure approaches, we achieve this by iteratively masking tokens in decreasing importance. We then measure the divergence between the output distributions of a model using a full-text as input and with the rationale masked. We assume that the higher the divergence, the more important a rationale is to a model and base our selection of these parameters off that. We empirically demonstrated that rationales extracted using our proposed method are on average shorter compared to rationales with fixed pre-defined parameters, whilst being more comprehensive and highly sufficient. We also show that our approach is flexible and scales with increasing feature attribution approaches and rationale lengths.

Chapter 5 presents an empirical study of explanations in out-of-domain setting. To study explanation faithfulness, we evaluated both post-hoc explanation faithfulness from feature attributions and the performance of select-then-predict models. We used two dataset triplets for these experiments, showing first that metrics such as sufficiency and comprehensiveness are misleading in out-of-domain settings. For this purpose, we suggest interpreting any measure of faithfulness alongside a random feature attribution for reference. On the contrary, both select-then-predict model variants employed exhibit comparable predictive performance with full-text trained models.

6.2 Future Directions

The findings of this thesis, open the following future research directions:

- The family of mechanisms proposed in Section 3.1 can be extended to other NLP applications to reduce contextualisations in attention mechanisms and subsequently improve faithfulness.
- The auxiliary objective presented in Section 3.2, uses TextRank scores to influence transformer-based language models towards other informative importance distributions and in turn results to more faithful rationales. Finding other informative distributions that can be extracted in an unsupervised manner a priori, or improving the loss function of this objective is an interesting direction for future research.
- The novel method proposed in Chapter 4, produced instance-specific explanations that were highly sufficient and on average more comprehensive compared to explanations that used pre-defined feature attribution, length and type. This method can be potentially modified to further improve on the sufficiency of rationales. Furthermore, whilst our approach has been demonstrated to be significantly more computationally efficient than computing importance scores using certain feature attribution approaches, there is still room for improvement in terms of the time taken to compute the rationale length.
- The empirical study of post-hoc explanation faithfulness and select-then-predict performance in Chapter 5, showed that metrics such as sufficiency and comprehensiveness are not reliable in out-of-domain settings. We consider an interesting direction of future study in this area, exploring methods for improving the evaluation of faithfulness for out-of-domain post-hoc explanations.

Bibliography

Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. 2020. Debugging tests for model explanations. In *Advances in Neural Information Processing Systems*, volume 33, pages 700–712. Curran Associates, Inc.

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.

Jasmijn Bastings and Katja Filippova. 2020. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 149–155, Online. Association for Computational Linguistics.

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.

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

Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2020. On identifiability in transformers. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Samuel Carton, Anirudh Rathore, and Chenhao Tan. 2020. Evaluating and characterizing human rationales. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9294–9307, Online. Association for Computational Linguistics.

Javier Castro, Daniel Gómez, and Juan Tejada. 2009. Polynomial calculation of the shapley value based on sampling. *Computers & Operations Research*, 36(5):1726–1730.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020a. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020b. LEGAL-BERT: The muppets straight out of law school. In *Findings*

of the Association for Computational Linguistics: EMNLP 2020, pages 2898–2904, 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.

George Chrysostomou and Nikolaos Aletras. 2021a. Enjoy the salience: Towards better transformer-based faithful explanations with word salience. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8189–8200, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

George Chrysostomou and Nikolaos Aletras. 2021b. Improving the faithfulness of attention-based explanations with task-specific information for text classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 477–488, Online. Association for Computational Linguistics.

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 *International Conference on Learning Representations*.

Gianna M. Del Corso, Antonio Gulli, and Francesco Romani. 2005. Ranking a stream of news. In *Proceedings of the 14th international conference on World Wide Web, WWW 2005, Chiba, Japan, May 10-14, 2005*, pages 97–106. ACM.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302, Online. 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. *arXiv*.

Eduard Dragut and Christiane Fellbaum. 2014. The role of adverbs in sentiment analysis. In *Proceedings of Frame Semantics in NLP: A Workshop in Honor of Chuck Fillmore (1929-2014)*, pages 38–41, Baltimore, MD, USA. Association for Computational Linguistics.

Michael Egmont-Petersen, Jan L Talmon, Jytte Brender, and Peter McNair. 1994. On the quality of neural net classifiers. *Artificial Intelligence in Medicine*, 6(5):359–381.

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.

Amirata Ghorbani, Abubakar Abid, and James Zou. 2019. Interpretation of neural networks is fragile. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3681–3688.

Max Glockner, Ivan Habernal, and Iryna Gurevych. 2020. Why do you think that? exploring faithful sentence-level rationales without supervision. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1080–1095, Online. Association for Computational Linguistics.

Yoav Goldberg. 2016. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420.

Bryce Goodman and Seth Flaxman. 2017. European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38(3):50–57.

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.

Nuno Miguel Guerreiro and André FT Martins. 2021. Spectra: Sparse structured text rationalization. *arXiv preprint arXiv:2109.04552*.

Kevin Gurney. 2014. *An introduction to neural networks*. CRC press.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751, Online. Association for Computational Linguistics.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. 2019. A benchmark for interpretability methods in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R Bowman. 2019. Do attention heads in bert track syntactic dependencies? *arXiv preprint arXiv:1911.12246*.

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.

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.

Brendan Kennedy, Xisen Jin, Aida Mostafazadeh Davani, Morteza Dehghani, and Xiang Ren. 2020. Contextualizing hate speech classifiers with post-hoc explanation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5435–5442, Online. Association for Computational Linguistics.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. 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 252–262, New Orleans, Louisiana. Association for Computational Linguistics.

Siwon Kim, Jihun Yi, Eunji Kim, and Sungroh Yoon. 2020. Interpretation of NLP models through input marginalization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3154–3167, Online. Association for Computational Linguistics.

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*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. 2007. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160:3–24.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

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.

Eric Lehman, Jay DeYoung, Regina Barzilay, and Byron C. Wallace. 2019. Inferring which medical treatments work from reports of clinical trials. 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 3705–3717, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Claude Lemaréchal. 2012. Cauchy and the gradient method. *Doc Math Extra*, 251(254):10.

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*.

Frederick Liu and Besim Avci. 2019. Incorporating priors with feature attribution on text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6274–6283, Florence, Italy. 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. 2019a. Roberta: A robustly optimized bert pretraining approach.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

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.

Andreas Madsen, Nicholas Meade, Vaibhav Adlakha, and Siva Reddy. 2021a. Evaluating the faithfulness of importance measures in nlp by recursively masking allegedly important tokens and retraining. *arXiv preprint arXiv:2110.08412*.

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2021b. Post-hoc interpretability for neural nlp: A survey. *arXiv preprint arXiv:2108.04840*.

Andre Martins and Ramon Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International Conference on Machine Learning*, pages 1614–1623. PMLR.

Seán Mc Loone and George Irwin. 2001. Improving neural network training solutions using regularisation. *Neurocomputing*, 37(1-4):71–90.

Warren S McCulloch and Walter Pitts. 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

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

Marvin L Minsky. 1968. Matter, minds, models.

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.

Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. 2012. *Introduction to linear regression analysis*, volume 821. John Wiley & Sons.

Pooya Moradi, Nishant Kambhatla, and Anoop Sarkar. 2021. Measuring and improving faithfulness of attention in neural machine translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2791–2802, Online. Association for Computational Linguistics.

Dong Nguyen. 2018a. 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.

Dong Nguyen. 2018b. 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.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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 188–197, Hong Kong, China. Association for Computational Linguistics.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can you trust your models uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Damian Pascual, Gino Brunner, and Roger Wattenhofer. 2020. Telling bert's full story: from local attention to global aggregation. *arXiv preprint arXiv:2004.05916*.

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, Doha, Qatar. Association for Computational Linguistics.

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.

Frank Rosenblatt. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver, Canada. Association for Computational Linguistics.

Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. 2017a. Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 2662–2670. ijcai.org.

Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. 2017b. Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 2662–2670. ijcai.org.

Sebastian Ruder. 2016. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

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.

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.

Raksha Sharma, Mohit Gupta, Astha Agarwal, and Pushpak Bhattacharyya. 2015. Adjective intensity and sentiment analysis. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2520–2526, Lisbon, Portugal. Association for Computational Linguistics.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3145–3153. PMLR.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, pages 3266–3280.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

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.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2018. On the practical computational power of finite precision RNNs for language recognition. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 740–745, Melbourne, Australia. Association for Computational Linguistics.

Paul Werbos. 1974. Beyond regression:" new tools for prediction and analysis in the behavioral sciences. *Ph. D. dissertation, Harvard University*.

Paul J Werbos. 1990. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

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.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256.

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.

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.

Song Xu, Haoran Li, Peng Yuan, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. Self-attention guided copy mechanism for abstractive summarization. In *Proceedings of the*

58th Annual Meeting of the Association for Computational Linguistics, pages 1355–1362, Online. Association for Computational Linguistics.

Atsuki Yamaguchi, George Chrysostomou, Katerina Margatina, and Nikolaos Aletras. 2021. Frustratingly simple pretraining alternatives to masked language modeling. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3116–3125, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jufeng Yang, Dongyu She, and Ming Sun. 2017. Joint image emotion classification and distribution learning via deep convolutional neural network. In *IJCAI*, pages 3266–3272.

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.

Wei Zhang, Ziming Huang, Yada Zhu, Guangnan Ye, Xiaodong Cui, and Fan Zhang. 2021. On sample based explanation methods for NLP: Faithfulness, efficiency and semantic evaluation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5399–5411, Online. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Ye Zhang and Byron Wallace. 2017. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 253–263, Taipei, Taiwan. Asian Federation of Natural Language Processing.

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).

Appendix A

Improving Attention-based Explanations

A.1 Predictive Performance

In Table A.1 we present predictive performances on the validation checks for reproducibility on models with TaSc and models without (No-TaSc).

A.2 Across Evaluation Metrics

We compare the performance of our proposed TaSc mechanisms, across a larger range of post-hoc explanation faithfulness evaluation metrics. This analysis aims to examine whether observations hold across evaluation approaches. For this purpose, alongside Fraction of Tokens (lower is better) we also employ AOPC Normalised Sufficiency (AOPC NormSuff; higher is better) and Normalised Comprehensiveness (AOPC NormComp; higher is better). For more details on these approaches, see Section 2.6. In Figures A.1 and A.2 we compare results across the three evaluation metrics when using α and $\nabla\alpha$ respectively.

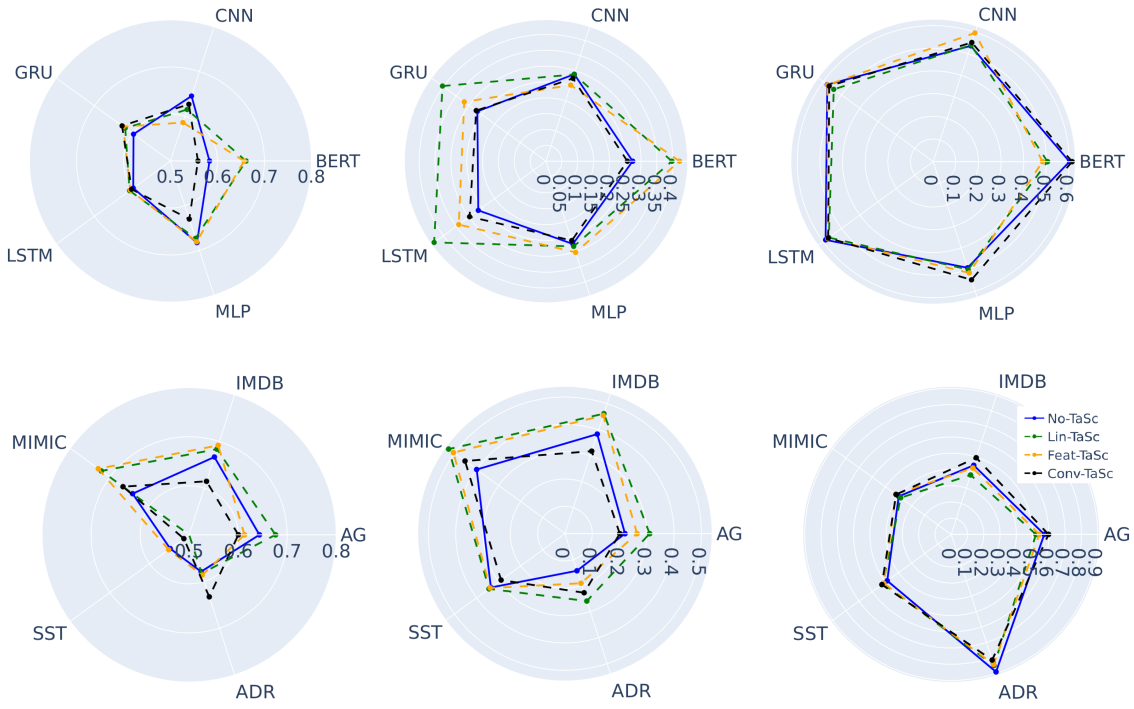
Comparing first results for α in Figure A.1, we can observe that they are largely in agreement across all three evaluation metrics. For example, Feat-TaSc and Lin-TaSc with BERT

Dataset	Enc()	No-TaSc		Lin-TaSc		Feat-TaSc		Conv-TaSc	
		Dot	Tanh	Dot	Tanh	Dot	Tanh	Dot	Tanh
SST	BERT	.89	.90	.90	.87	.87	.87	.90	.90
	LSTM	.77	.78	.77	.78	.77	.80	.79	.80
	GRU	.78	.78	.78	.79	.78	.79	.78	.79
	MLP	.75	.77	.78	.78	.80	.80	.79	.81
	CNN	.77	.77	.79	.80	.80	.79	.79	.78
ADR	BERT	.81	.81	.81	.79	.80	.80	.80	.81
	LSTM	.74	.75	.77	.76	.77	.77	.78	.76
	GRU	.76	.75	.77	.77	.76	.79	.77	.77
	MLP	.73	.78	.76	.76	.78	.77	.76	.76
	CNN	.74	.73	.77	.76	.77	.77	.78	.78
IMDB	BERT	.92	.92	.93	.92	.92	.92	.92	.92
	LSTM	.90	.89	.89	.89	.89	.89	.89	.89
	GRU	.90	.90	.89	.90	.89	.90	.89	.89
	MLP	.88	.88	.88	.88	.89	.88	.89	.88
	CNN	.89	.89	.90	.89	.89	.89	.89	.89
AG	BERT	.95	.95	.94	.94	.95	.95	.94	.95
	LSTM	.93	.93	.92	.93	.93	.93	.93	.93
	GRU	.93	.93	.93	.93	.93	.93	.93	.93
	MLP	.93	.93	.93	.92	.93	.92	.93	.93
	CNN	.93	.93	.93	.93	.93	.93	.93	.93
MIMIC	BERT	.84	.83	.85	.84	.86	.84	.85	.83
	LSTM	.88	.89	.89	.89	.89	.90	.90	.90
	GRU	.89	.90	.89	.89	.90	.90	.90	.90
	MLP	.90	.89	.88	.88	.89	.88	.89	.89
	CNN	.90	.89	.90	.90	.90	.90	.89	.90

Table A.1: Validation set F1-macro average scores (3 runs) across datasets, encoders and attention mechanisms for models with and without TaSc (No-TaSc). Standard deviations do not exceed 0.01.

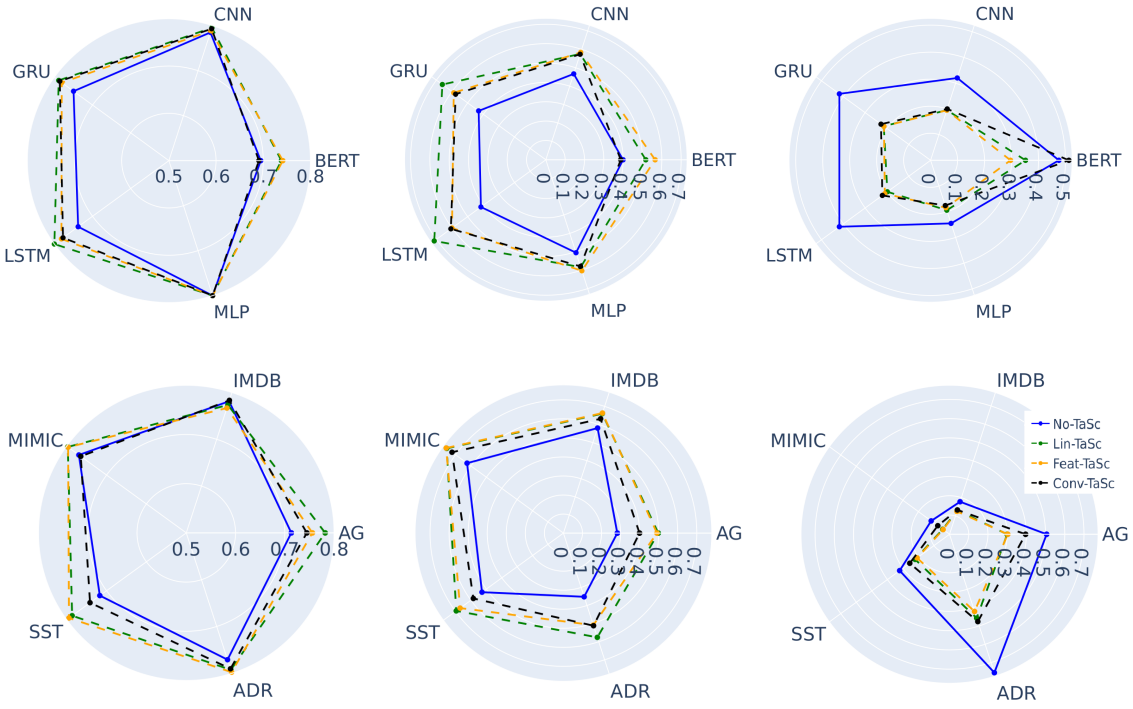
result in more comprehensive and sufficient rationales compared to No-TaSc and require a lower fraction of tokens to cause a prediction switch. This is also observed when comparing across datasets, with Feat-TaSc and Lin-TaSc resulting to more faithful explanations in IMDB across all three metrics.

In Figure A.2, we also observe agreement between metrics in regards to post-hoc explanation faithfulness using $\nabla\alpha$. For example, our proposed TaSc mechanisms require a lower fraction of tokens to cause a decision flip with LSTM and also result to higher AOPC NormComp and AOPC NormSuff scores. Comparing between the two feature attribution approaches (α, $\nabla\alpha$), all three evaluation metrics agree that $\nabla\alpha$ results to more faithful post-hoc explanations compared to α. Overall, our findings show that using either evaluation metric TaSc results to more faithful attention-based explanations.



(a) AOPC NormSuff **(b)** AOPC NormComp **(c)** Fraction of Tokens

Figure A.1: Mean AOPC Normalised Sufficiency (higher is better), AOPC Normalised Comprehensiveness (higher is better) and Fraction of Tokens (lower is better) occurred by removing the most informative token, using the three TaSc variants and No-TaSc across encoders (first row) and datasets (second row), using α.



(a) AOPC NormSuff **(b)** AOPC NormComp **(c)** Fraction of Tokens

Figure A.2: Mean AOPC Normalised Sufficiency (higher is better), AOPC Normalised Comprehensiveness (higher is better) and Fraction of Tokens (lower is better) occurred by removing the most informative token, using the three TaSc variants and No-TaSc across encoders (first row) and datasets (second row), using $\nabla\alpha$.

Appendix B

Improving Transformer-based Explanations

B.1 PoS Importance Allocation

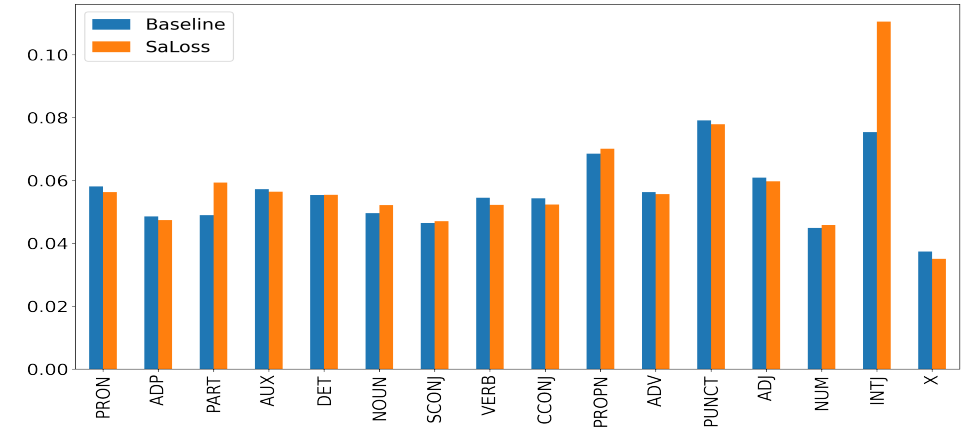
We also conduct an analysis whereby we record the average importance scores under each Part of Speach (PoS) tag. We run a pretrained PoS tagger from `spaCy` (Honnibal et al., 2020) across the text and compute average importance calculated from a feature attribution approach for each PoS tag. We therefore aim to observe differences in allocation of importance in linguistic features between models trained with out our proposed approach (Baseline) and with (SaLoss). In Figure B.1 we present distribution of importance (calculated with $\alpha\nabla\alpha$) across PoS tags, on three datasets (SST, AG and Ev.Inf.).

Observing Figure B.1a, we can see that $\alpha\nabla\alpha$ with SaLoss places greater importance on proper nouns (PROPN), auxiliary words (AUX), pronouns (PRON) and interjections (INTJ). In comparison the most prominent tags with Baseline are INTJ, PROPN, coordinating conjunctions (CCONJ) and nouns (NOUN). In a sentiment analysis task, it is notable that both Baseline and SaLoss base high importance on average on interjections, which typically demonstrate feelings or emotions. Both appear to highlight particularly well adjectives, which we consider more important for sentiment analysis as they name attributes of other words. On the other end we also observe that SaLoss places lower importance on average to CCONJ

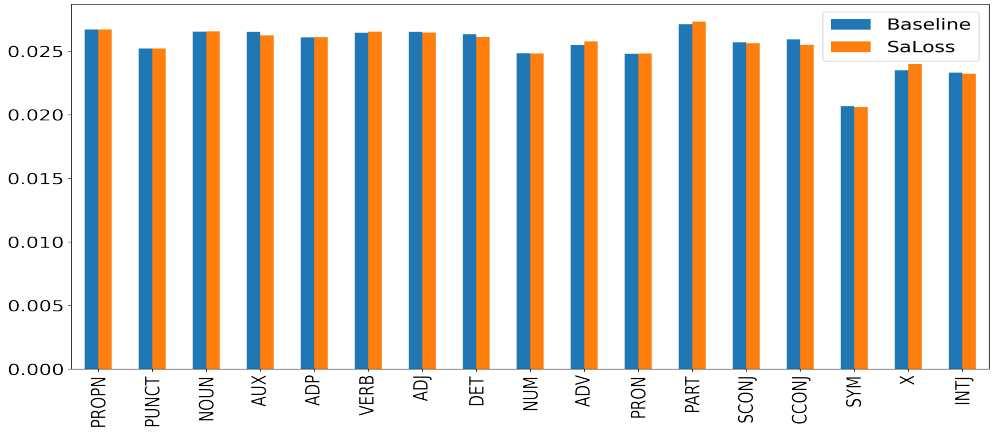
and punctuation (PUNCT) compared to Baseline This suggests that for SST, SaLoss models possibly shift their importance to more informative for the task word groups.

Moving on to Figure B.1b, we observe a very high peak on proper nouns (PROPN) and unidentified tokens (X) with SaLoss compared to Baseline. In a news classification task proper nouns such as "*NATO*" and other organization or city names can indicate the topic of a sequence. We assume that for SaLoss to place such great importance on proper nouns, we manage with our approach to shift the model's attention to more informative for the task tokens. However we also observe unidentified symbols having large average importance scores with SaLoss. Whilst we do not study plausibility (human understandability of explanations), we consider this a limitation and we consider exploring and addressing this an interesting direction for future work.

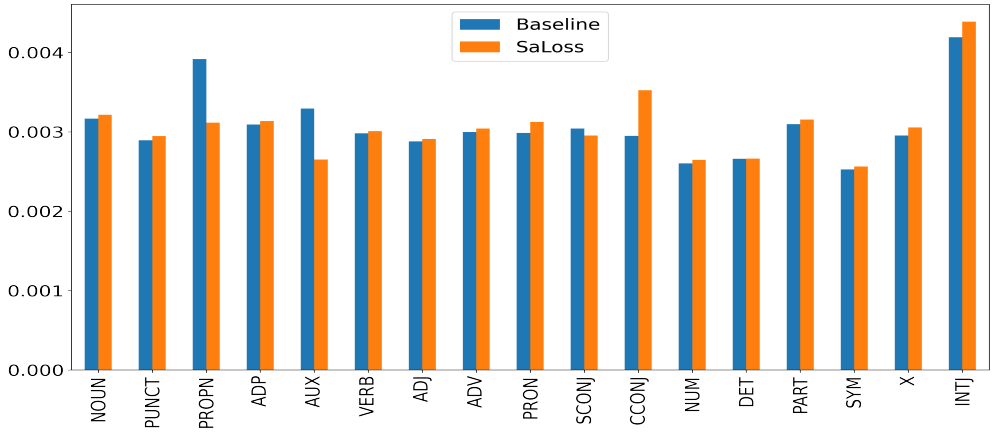
Finally, examining Figure B.1c, we observe that both SaLoss and Baseline place very high importance on particle (PART) words such as *not*. We consider this encouraging, as large parts of the task is to infer if there was a significant difference or *not* based on an observation in the text. Additionally, we observe that SaLoss attends highly to subordinating conjunction (SCONJ) words such as than, which if placed in the context of "*significantly higher than*" directly relates to our task. Also with SaLoss we observe a reduction in attention to pronouns (PRON) compared to Baseline, which we consider encouraging as PRON words are not directly related to the task of infering relationships. This indicates that our proposed objective manages to guide the model's attention away from uninformative tokens such as others and punctation, and towards more informative for the task token types (SCONJ, CCONJ).



(a) (SST)



(b) (AG)



(c) (Ev.Inf.)

Figure B.1: Average importance across Part of Speech (PoS) tags as indicated by $\alpha\nabla\alpha$ with Baseline, with our proposed component SaLoss.

Appendix C

An Empirical Analysis of Faithfulness in Out-of-Domain Settings

C.1 Feature Attribution Correlation Analysis

We examine why $x\nabla x$ and IG, do not perform as well as DeepLift and $\alpha\nabla\alpha$ when using FRESH. We therefore conduct a study to gain better understand this. We first fix the domain of the data we evaluate on and then compute the correlation between importance rankings using any single feature attribution from: (1) a model trained on the same domain with the evaluation data and (2) a model from trained on a different distribution (out-of-domain trained model). High correlations suggest that a feature attribution from an out-of-domain trained model, produce similar importance distributions with that of an in-domain model (i.e. both attend to similar tokens to make a prediction). Therefore, we assume that this will lead to high predictive performance out-of-domain. In Figure C.1 we show Spearman's ranking correlation across dataset pairs, between a model trained on the same distribution as the evaluation data (ID) and an out-of-domain trained model (OOD), such that (ID $<->$ OOD).

As expected, the random baseline produced almost no correlation between models. An interesting observation is that two of the gradient-based methods ($x\nabla x$ and IG) produce strongly correlated rankings. This suggests that these two metrics produce generalizable

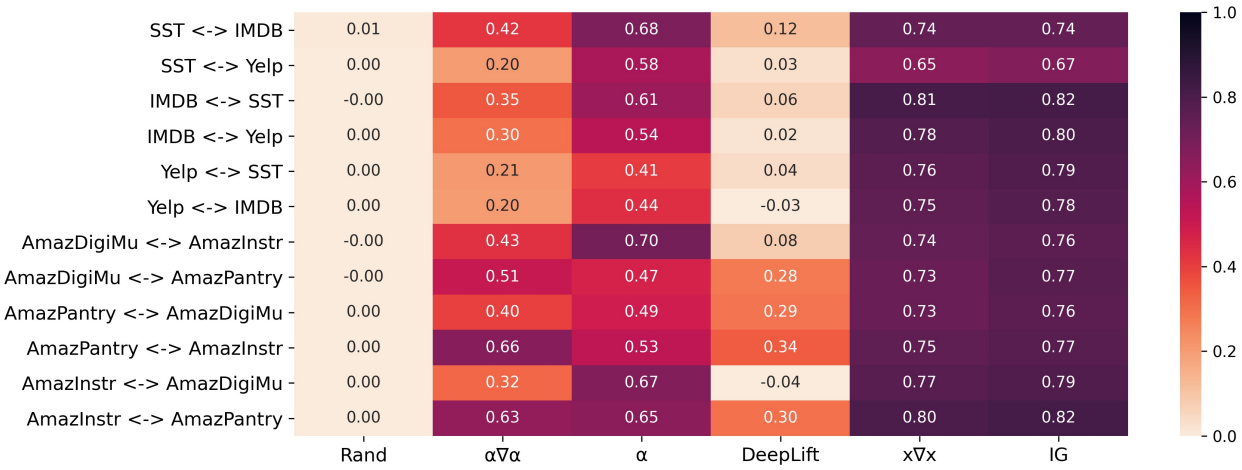


Figure C.1: Average Spearman's ranking correlation coefficient, between feature attribution rankings from: (1) a model trained on the same distribution as the evaluation data (ID) and (2) from a model trained in another domain (OOD), such that ID $< - >$ OOD.

rankings irrespective of the domain shift, when comparing to the remainder of the feature attribution approaches. Surprisingly, Deeplift importance rankings exhibit almost low to no correlation betweenen them, despite being also gradient-based. We hypothesize that this happens because DeepLift considers a baseline input to compute its importance distribution, which highly depends on the model and as such is de-facto normalized and perhaps generalizes better.

α for out-of-domain detection?: An interesting case is that of α, where we observe moderate to strong correlations across all test-cases. What is more evident, is that in the OOD tuples we considered, it appears that stronger correlations appear where the OOD task and the ID task are closer together. For example in the case of SST and IMDB (both sentiment analysis tasks for movie reviews), α produces a strong correlation (0.68). This contrasts the moderate correlation of 0.58 between SST and Yelp, which is for restaurant reviews. This is also evident in the case of AmazDigiMu and AmazInstr, where both tasks are for review classification, but for musical related purchases. They both score strong correlations between them and moderate correlations with reviews for pantry purchases (AmazPantry). This observation might suggest, that *using these correlation metrics with α might be an indicator of the degree of task-domain-shift*. Our observation is also supported by the findings of Adebayo et al. (2020), who show that feature attributions are good indicators of detecting

ID	OOD	Rand	$\alpha\nabla\alpha$	α	DeepLift	$x\nabla x$	IG
SST	IMDB	0.06	0.26	0.39	0.37	0.54	0.55
SST	Yelp	0.07	0.11	0.27	0.29	0.46	0.49
IMDB	SST	0.02	0.13	0.25	0.15	0.43	0.43
IMDB	Yelp	0.02	0.08	0.16	0.09	0.43	0.43
Yelp	SST	0.02	0.08	0.12	0.18	0.37	0.39
Yelp	IMDB	0.02	0.05	0.12	0.10	0.40	0.41
AmazDigiMu	AmazInstr	0.13	0.22	0.38	0.16	0.60	0.61
AmazDigiMu	AmazPantry	0.13	0.30	0.36	0.27	0.60	0.62
AmazPantry	AmazDigiMu	0.14	0.28	0.35	0.27	0.60	0.63
AmazPantry	AmazInstr	0.14	0.39	0.42	0.21	0.62	0.64
AmazInstr	AmazDigiMu	0.08	0.16	0.29	0.12	0.54	0.57
AmazInstr	AmazPantry	0.08	0.29	0.36	0.14	0.57	0.59

Table C.1: Agreement in tokens at 2% rationale length between a feature attribution from an ID model tested on ID and the same feature attribution trained on an OOD dataset and tested on ID.

spurious correlation signals in computer vision tasks.Considering $\alpha\nabla\alpha$ we observe a wide range of correlations, ranging from low in the AmazInstr-AmazDigiMu pair to strong in the AmazPantry-AmazInstr pair, which we cannot interpret as something meaningful.

Correlation values and FRESH: We first observe that the lowest correlated feature attributions $\alpha\nabla\alpha$ and DeepLift perform the better on FRESH, followed by α which displays moderate correlations and at the end of the spectrum the two gradient-based methods which display high correlations. Contrary to our initial assumption, this suggests that the *attributions which generalize better (i.e. return rationales that result in higher FRESH performance) are those which exhibit low to no correlations.*

Agreement at different rationale lengths: As the correlation analysis considers the entire length of the sequence, we now examine a scenario where we have a priori defined rationale lengths. Similarly to the correlation analysis, we now compute the agreement in tokens between ID feature attribution rankings to those of an OOD trained model. In Tables C.1, C.2 and C.3 we therefore show the token agreement between in-domain and out-of-domain post-hoc explanations (on the same data) for 2%, 10% and 20% rationale lengths.

Our findings show that across all rationale lengths, results largely agree with the correlation analysis. The two gradient-based methods exhibit higher agreement than the remainder,

ID	OOD	Rand	$\alpha\nabla\alpha$	α	DeepLift	$x\nabla x$	IG
SST	IMDB	0.10	0.32	0.47	0.33	0.60	0.61
SST	Yelp	0.11	0.19	0.35	0.25	0.54	0.56
IMDB	SST	0.10	0.29	0.41	0.17	0.60	0.61
IMDB	Yelp	0.10	0.21	0.34	0.14	0.59	0.61
Yelp	SST	0.10	0.18	0.28	0.16	0.55	0.57
Yelp	IMDB	0.10	0.16	0.29	0.12	0.56	0.58
AmazDigiMu	AmazInstr	0.17	0.29	0.47	0.16	0.66	0.68
AmazDigiMu	AmazPantry	0.17	0.36	0.44	0.26	0.66	0.69
AmazPantry	AmazDigiMu	0.17	0.33	0.42	0.27	0.66	0.68
AmazPantry	AmazInstr	0.17	0.46	0.49	0.24	0.67	0.69
AmazInstr	AmazDigiMu	0.13	0.24	0.43	0.11	0.64	0.66
AmazInstr	AmazPantry	0.13	0.40	0.50	0.20	0.67	0.68

Table C.2: Agreement in tokens at 10% rationale length between a feature attribution from an ID model tested on ID and the same feature attribution trained on an OOD dataset and tested on ID.

ID	OOD	Rand	$\alpha\nabla\alpha$	α	DeepLift	$x\nabla x$	IG
SST	IMDB	0.20	0.42	0.57	0.34	0.68	0.67
SST	Yelp	0.21	0.31	0.46	0.27	0.61	0.62
IMDB	SST	0.20	0.39	0.52	0.26	0.69	0.69
IMDB	Yelp	0.20	0.32	0.46	0.22	0.67	0.68
Yelp	SST	0.20	0.29	0.41	0.24	0.64	0.66
Yelp	IMDB	0.20	0.27	0.42	0.20	0.65	0.66
AmazDigiMu	AmazInstr	0.23	0.37	0.55	0.21	0.71	0.73
AmazDigiMu	AmazPantry	0.24	0.44	0.51	0.32	0.71	0.74
AmazPantry	AmazDigiMu	0.24	0.40	0.50	0.33	0.71	0.73
AmazPantry	AmazInstr	0.24	0.54	0.57	0.32	0.72	0.73
AmazInstr	AmazDigiMu	0.21	0.33	0.54	0.16	0.70	0.72
AmazInstr	AmazPantry	0.21	0.51	0.60	0.30	0.72	0.74

Table C.3: Agreement in tokens at 20% rationale length between a feature attribution from an ID model tested on ID and the same feature attribution trained on an OOD dataset and tested on ID.

with $\alpha\nabla\alpha$ and DeepLift recording the lowest agreements. Surprisingly, the poorest performers on out-of-domain FRESH record the highest agreement in tokens with in-domain models. Whilst this suggests that they generalize better, we believe that the inhibiting factor to their performance is their limited in-domain capabilities (i.e. they record the lowest in-domain FRESH performance with TopK).

Train	Test	AOPC NormSuff						AOPC NormComp					
		Rand	$\alpha\nabla\alpha$	α	DeepLift	$x\nabla x$	IG	Rand	$\alpha\nabla\alpha$	α	DeepLift	$x\nabla x$	IG
SST	SST	0.42	0.46	0.40	0.42	0.43	0.43	0.11	0.29	0.00	0.11	0.19	0.19
	IMDB	0.35	0.40	0.33	0.35	0.34	0.35	0.11	0.39	0.14	0.13	0.17	0.18
	Yelp	0.36	0.41	0.32	0.37	0.32	0.33	0.10	0.31	0.08	0.10	0.11	0.13
IMDB	IMDB	0.36	0.42	0.39	0.37	0.37	0.37	0.05	0.27	0.14	0.06	0.11	0.12
	SST	0.29	0.30	0.29	0.30	0.30	0.30	0.16	0.33	0.16	0.16	0.21	0.19
	Yelp	0.40	0.45	0.43	0.41	0.40	0.40	0.10	0.35	0.21	0.10	0.13	0.13
Yelp	Yelp	0.12	0.13	0.13	0.13	0.13	0.13	0.02	0.06	0.01	0.02	0.04	0.05
	SST	0.47	0.46	0.46	0.48	0.47	0.47	0.08	0.09	0.00	0.09	0.12	0.12
	IMDB	0.11	0.11	0.11	0.12	0.11	0.11	0.07	0.19	0.10	0.08	0.10	0.10
AmazDigiMu	AmazDigiMu	0.24	0.42	0.16	0.17	0.30	0.29	0.09	0.25	0.04	0.02	0.12	0.13
	AmazInstr	0.17	0.33	0.13	0.13	0.21	0.21	0.14	0.41	0.10	0.06	0.17	0.18
	AmazPantry	0.27	0.45	0.20	0.21	0.30	0.29	0.18	0.43	0.10	0.05	0.20	0.22
AmazPantry	AmazPantry	0.23	0.34	0.27	0.16	0.23	0.22	0.11	0.32	0.19	0.03	0.15	0.15
	AmazDigiMu	0.22	0.35	0.29	0.16	0.22	0.22	0.10	0.29	0.19	0.03	0.12	0.12
	AmazInstr	0.14	0.23	0.18	0.11	0.15	0.14	0.12	0.39	0.23	0.07	0.16	0.17
AmazInstr	AmazInstr	0.13	0.18	0.09	0.11	0.13	0.13	0.16	0.40	0.05	0.08	0.17	0.18
	AmazDigiMu	0.19	0.29	0.12	0.13	0.19	0.18	0.14	0.35	0.04	0.05	0.14	0.15
	AmazPantry	0.20	0.30	0.14	0.15	0.20	0.20	0.19	0.45	0.04	0.08	0.18	0.21

Table C.4: Normalized Sufficiency and Comprehensiveness (higher is better) in-domain and out-of-domain at 2% rationale length, for five feature attribution approaches and a random attribution baseline.

C.2 Post-hoc Explanation Faithfulness - Extended

In Tables C.4, C.5 and C.6, we present post-hoc explanation sufficiency and comprehensiveness scores at 2%, 10% and 20% rationale lengths.

C.3 FRESH Model Performance

Table C.7 presents FRESH F1 macro performance for classifiers trained on Contiguous rationales, with standard deviation in brackets. We include the a priori defined rationale length

Train	Test	AOPC NormSuff						AOPC NormComp					
		Rand	$\alpha\nabla\alpha$	α	DeepLift	$x\nabla x$	IG	Rand	$\alpha\nabla\alpha$	α	DeepLift	$x\nabla x$	IG
SST	SST	0.43	0.55	0.43	0.46	0.44	0.45	0.16	0.42	0.20	0.22	0.25	0.25
	IMDB	0.36	0.65	0.44	0.37	0.36	0.36	0.19	0.69	0.39	0.24	0.25	0.26
	Yelp	0.37	0.67	0.37	0.39	0.33	0.34	0.17	0.58	0.25	0.20	0.22	0.24
IMDB	IMDB	0.37	0.64	0.54	0.40	0.39	0.39	0.10	0.55	0.30	0.17	0.18	0.18
	SST	0.28	0.32	0.29	0.30	0.30	0.30	0.23	0.48	0.29	0.29	0.30	0.29
	Yelp	0.41	0.54	0.46	0.43	0.41	0.41	0.18	0.58	0.36	0.22	0.24	0.24
Yelp	Yelp	0.17	0.22	0.23	0.26	0.19	0.20	0.05	0.15	0.05	0.06	0.08	0.08
	SST	0.48	0.49	0.47	0.50	0.46	0.46	0.13	0.23	0.15	0.16	0.20	0.20
	IMDB	0.13	0.29	0.29	0.22	0.14	0.15	0.13	0.35	0.28	0.16	0.18	0.19
AmazDigiMu	AmazDigiMu	0.33	0.67	0.24	0.25	0.39	0.36	0.11	0.34	0.08	0.06	0.15	0.16
	AmazInstr	0.28	0.67	0.22	0.26	0.29	0.28	0.19	0.57	0.19	0.15	0.22	0.24
	AmazPantry	0.33	0.64	0.25	0.28	0.36	0.34	0.22	0.55	0.17	0.12	0.25	0.26
AmazPantry	AmazPantry	0.23	0.46	0.34	0.17	0.24	0.23	0.15	0.45	0.29	0.10	0.20	0.21
	AmazDigiMu	0.22	0.46	0.35	0.16	0.23	0.22	0.13	0.42	0.29	0.10	0.17	0.17
	AmazInstr	0.14	0.42	0.27	0.12	0.16	0.15	0.18	0.59	0.40	0.17	0.24	0.25
AmazInstr	AmazInstr	0.13	0.28	0.09	0.12	0.13	0.13	0.23	0.58	0.16	0.22	0.24	0.25
	AmazDigiMu	0.19	0.32	0.12	0.14	0.20	0.18	0.18	0.47	0.10	0.14	0.20	0.20
	AmazPantry	0.21	0.35	0.15	0.17	0.21	0.21	0.24	0.57	0.12	0.18	0.24	0.27

Table C.5: Normalized Sufficiency and Comprehensiveness (higher is better) in-domain and out-of-domain at 10% rationale length, for five feature attribution approaches and a random attribution baseline.

in the brackets (.%) and for reference, the ID performance of the Full-Text model.

Comparing with FRESH performance with Contiguous rationales rather than TopK (see Table 5.5), we first observe that performance degrades for most feature attribution methods. These findings are largely in agreement with those of Jain et al. (2020). However, $x\nabla x$ and IG, which perform poorly with TopK, record surprisingly better scores with Contiguous type rationales. For example, in-domain performance with IG becomes comparable with $\alpha\nabla\alpha$ in in-domain IMDB (83.2 with $\alpha\nabla\alpha$ and 82.5 with IG). This is in sharp contrast with TopK, where IG recorded an F1 score of only 59.7, compared to 87.9 of $\alpha\nabla\alpha$.

These findings also hold in out-of-domain settings, where $\alpha\nabla\alpha$, α and DeepLift result in

Train	Test	AOPC NormSuff						AOPC NormComp					
		Rand	$\alpha\nabla\alpha$	α	DeepLift	$x\nabla x$	IG	Rand	$\alpha\nabla\alpha$	α	DeepLift	$x\nabla x$	IG
SST	SST	0.45	0.68	0.51	0.51	0.48	0.49	0.22	0.54	0.34	0.33	0.32	0.34
	IMDB	0.38	0.77	0.55	0.39	0.37	0.38	0.29	0.80	0.54	0.36	0.34	0.36
	Yelp	0.39	0.83	0.57	0.41	0.37	0.38	0.25	0.71	0.44	0.30	0.32	0.34
IMDB	IMDB	0.37	0.75	0.62	0.42	0.41	0.42	0.16	0.73	0.47	0.30	0.27	0.27
	SST	0.26	0.40	0.31	0.31	0.31	0.30	0.32	0.65	0.42	0.41	0.42	0.42
	Yelp	0.42	0.62	0.50	0.43	0.44	0.44	0.28	0.67	0.47	0.35	0.36	0.37
Yelp	Yelp	0.25	0.43	0.41	0.40	0.28	0.30	0.09	0.25	0.12	0.13	0.14	0.15
	SST	0.49	0.55	0.51	0.53	0.48	0.48	0.20	0.35	0.27	0.26	0.28	0.29
	IMDB	0.19	0.53	0.50	0.34	0.24	0.25	0.20	0.46	0.40	0.27	0.28	0.28
AmazDigiMu	AmazDigiMu	0.43	0.81	0.47	0.35	0.52	0.50	0.14	0.41	0.17	0.10	0.19	0.20
	AmazInstr	0.37	0.79	0.49	0.42	0.43	0.42	0.24	0.63	0.33	0.23	0.28	0.30
	AmazPantry	0.42	0.76	0.45	0.37	0.47	0.45	0.26	0.61	0.31	0.20	0.30	0.32
AmazPantry	AmazPantry	0.27	0.63	0.46	0.19	0.30	0.29	0.21	0.57	0.40	0.17	0.28	0.29
	AmazDigiMu	0.25	0.63	0.46	0.18	0.28	0.27	0.19	0.55	0.39	0.16	0.25	0.25
	AmazInstr	0.16	0.61	0.42	0.14	0.21	0.20	0.27	0.72	0.54	0.26	0.35	0.36
AmazInstr	AmazInstr	0.15	0.46	0.15	0.18	0.17	0.16	0.31	0.72	0.33	0.34	0.32	0.34
	AmazDigiMu	0.21	0.46	0.16	0.17	0.23	0.20	0.24	0.60	0.22	0.22	0.26	0.27
	AmazPantry	0.23	0.49	0.18	0.21	0.24	0.23	0.31	0.68	0.28	0.28	0.32	0.35

Table C.6: Normalized Sufficiency and Comprehensiveness (higher is better) in-domain and out-of-domain at 20% rationale length, for five feature attribution approaches and a random attribution baseline.

poorer FRESH performance with Contiguous type rationales, compared to TopK. However, IG and in many cases $x\nabla x$ improves. For example with TopK rationales, evaluating on Yelp using IG from a model trained on IMDB, results on an F1-score of 69.1. On the contrary, with Contiguous rationales and the same set-up, IG results in FRESH performance of 87.0.

Our findings lead us to assume that, *the rationale type has a large impact on FRESH performance, both in-domain and on out-of-domain settings. Certain feature attribution methods benefit from one type of rationales (e.g. DeepLift with TopK), whilst others from another (e.g. IG with Contiguous).*

Train	Test	Full-Text	F1				
			$\alpha\nabla\alpha$	α	DeepLift	$x\nabla x$	IG
SST (20%)	SST	90.1 (0.3)	87.1 (0.8)	80.7 (0.4)	79.7 (1.5)	77.8 (0.6)	79.7 (1.5)
	IMDB	84.3 (0.6)	80.3 (0.5)	58.8 (0.4)	64.9 (1.5)	53.1 (0.7)	64.9 (1.5)
	Yelp	87.9 (2.3)	**88.1**(0.3)	74.8 (1.0)	69.5 (0.9)	71.7 (1.1)	88.1 (0.3)
IMDB (2%)	IMDB	91.1 (0.4)	83.2 (0.1)	75.6 (0.6)	82.5 (0.8)	62.7 (0.2)	82.5 (0.8)
	SST	85.8 (2.0)	80.1 (1.1)	74.7 (1.2)	66.7 (0.6)	71.6 (1.2)	80.1 (1.1)
	Yelp	91.0 (1.2)	87.0 (0.3)	80.8 (1.3)	69.2 (4.4)	73.8 (0.8)	87.0 (0.3)
Yelp (10%)	Yelp	96.9 (0.1)	91.8 (0.5)	81.7 (0.3)	89.0 (0.7)	81.8 (0.2)	89.0 (0.7)
	SST	86.8 (1.7)	65.5 (2.2)	71.3 (1.3)	68.4 (1.0)	68.7 (0.5)	65.5 (2.2)
	IMDB	88.6 (0.3)	75.3 (1.2)	62.1 (0.9)	67.5 (0.2)	55.8 (0.4)	67.5 (0.2)
AmazDigiMu (20%)	AmazDigiMu	70.6 (0.9)	65.8 (1.5)	60.1 (2.3)	59.5 (4.0)	55.9 (2.4)	59.5 (4.0)
	AmazInstr	61.2 (1.8)	57.0 (0.9)	51.8 (2.0)	50.8 (1.8)	47.5 (0.6)	51.8 (2.0)
	AmazPantry	64.6 (1.0)	57.7 (0.6)	51.6 (2.0)	51.4 (2.6)	47.5 (1.2)	47.5 (1.2)
AmazPantry (20%)	AmazPantry	70.2 (1.1)	**63.5**(3.6)	62.0 (0.4)	58.0 (1.0)	50.0 (2.1)	58.0 (1.0)
	AmazDigiMu	59.5 (0.7)	**53.7**(3.6)	52.0 (1.4)	46.7 (0.7)	44.4 (2.7)	53.7 (3.6)
	AmazInstr	64.5 (2.6)	**59.1**(3.9)	56.1 (1.5)	51.4 (0.6)	42.6 (3.6)	56.1 (1.5)
AmazInstr (20%)	AmazInstr	71.5 (0.4)	66.3 (1.1)	52.2 (2.3)	60.9 (0.8)	53.4 (1.2)	60.9 (0.8)
	AmazDigiMu	61.3 (0.3)	56.5 (0.6)	47.0 (1.4)	52.1 (0.3)	48.3 (1.2)	56.5 (0.6)
	AmazPantry	68.2 (0.7)	62.4 (0.9)	49.2 (1.7)	57.4 (0.6)	51.0 (1.3)	51.0 (1.3)

Table C.7: F1 macro performance of FRESH models (Contiguous rationales) with standard deviation in brackets and Expected Calibration Error (ECE) scores. For reference we include the in-domain performance of full-text models. **Bold** denotes no significant difference between FRESH and Full-text (t-test; $p > 0.05$)