How do Decisions Emerge across Layers in Neural Models? Interpretation with Differentiable Masking

De Cao, N.; Schlichtkrull, M.; Aziz, W.; Titov, I.

DOI
10.18653/v1/2020.emnlp-main.262

Publication date
2020

Document Version
Final published version

Published in
2020 Conference on Empirical Methods in Natural Language Processing

License
CC BY

Citation for published version (APA):
How do Decisions Emerge across Layers in Neural Models? Interpretation with Differentiable Masking

Nicola De Cao 1,2, Michael Schlichtkrull 1,2, Wilker Aziz 1, Ivan Titov 1,2
1University of Amsterdam, 2University of Edinburgh
{ nicola.decao, m.s.schlichtkrull, w.aziz } @uva.nl
ititov@inf.ed.ac.uk

Abstract

Attribution methods assess the contribution of inputs to the model prediction. One way to do so is erasure: a subset of inputs is considered irrelevant if it can be removed without affecting the prediction. Though conceptually simple, erasure’s objective is intractable and approximate search remains expensive with modern deep NLP models. Erasure is also susceptible to the hindsight bias: the fact that an input can be dropped does not mean that the model ‘knows’ it can be dropped. The resulting pruning is over-aggressive and does not reflect how the model arrives at the prediction. To deal with these challenges, we introduce Differentiable Masking. DIFFMASK learns to mask-out subsets of the input while maintaining differentiability. The decision to include or disregard an input token is made with a simple model based on intermediate hidden layers of the analyzed model. First, this makes the approach efficient because we predict rather than search. Second, as with probing classifiers, this reveals what the network ‘knows’ at the corresponding layers. This lets us not only plot attribution heatmaps but also analyze how decisions are formed across network layers. We use DIFFMASK to study BERT models on sentiment classification and question answering.1

1 Introduction

Deep neural networks have become standard tools in NLP demonstrating impressive improvements over traditional approaches on many tasks (Goldberg, 2017). Their power typically comes at the expense of interpretability, which may prevent users from trusting predictions (Kim, 2015; Ribeiro et al., 2016), makes it hard to detect model or data deficiencies (Gururangan et al., 2018; Kaushik and Lipton, 2018) or verify that a model is fair and does not exhibit harmful biases (Sun et al., 2019; Holstein et al., 2019).

These challenges have motivated work on interpretability, both in NLP and generally in machine learning; see Belinkov and Glass (2019) and Jacovi and Goldberg (2020) for reviews. In this work, we study post hoc interpretability where the goal is to explain the prediction of a trained model and to reveal how the model arrives at the decision. This goal is usually approached with attribution methods (Bach et al., 2015; Shrikumar et al., 2017; Sundararajan et al., 2017), which explain the behavior of a model by assigning relevance to inputs.

One way to perform attribution is to use erasure where a subset of features (e.g., input tokens) is considered irrelevant if it can be removed without affecting the model prediction (Li et al., 2016; Feng et al., 2018). The advantage of erasure is that it is conceptually simple and optimizes a well-defined objective. This contrasts with most other attribution methods which rely on heuristic rules to define feature salience; for example, attention-based attri-
Question: Where did the Broncos practice for the Super Bowl?
Passage: The Panthers used the San Jose State practice facility and stayed at the San Jose Marriott. The Broncos practiced at Stanford University and stayed at the Santa Clara Marriott.

(a) Integrated Gradient (Sundararajan et al., 2017).
Question: Where did the Broncos practice for the Super Bowl?
Passage: The Panthers used the San Jose State practice facility and stayed at the San Jose Marriott. The Broncos practiced at Stanford University and stayed at the Santa Clara Marriott.

(c) NLP explainer (Guan et al., 2019).
Question: Where did the Broncos practice for the Super Bowl?
Passage: The Panthers used the San Jose State practice facility and stayed at the San Jose Marriott. The Broncos practiced at Stanford University and stayed at the Santa Clara Marriott.

(e) Our DIFFMASK.

Question: Where did the Broncos practice for the Super Bowl?
Passage: The Panthers used the San Jose State practice facility and stayed at the San Jose Marriott. The Broncos practiced at Stanford University and stayed at the Santa Clara Marriott.

(f) Our DIFFMASK non-amortized.

Figure 2: Question Answering token attribution: (b) and (c), are misleading (i.e., not faithful) as they attribute the prediction mostly to the answer span itself (underlined). Our method (d) reveals that the model pays attention to other named entities and the predicate 'practice' in both sentences. Predictions of the path-based methods (a) are more spread-out. Exact search (e) as well as approximate search (f) leads to pathological attributions.

bution (Rocktäschel et al., 2016; Serrano and Smith, 2019; Vashishth et al., 2019) or back-propagation methods (Bach et al., 2015; Shrikumar et al., 2017; Sundararajan et al., 2017). These approaches received much scrutiny in recent years (Nie et al., 2018; Sixt et al., 2020; Jain and Wallace, 2019), as they cannot guarantee that the network is ignoring low-scored features. They are often motivated as approximations of erasure (Baehrens et al., 2010; Simonyan et al., 2014; Feng et al., 2018) and sometimes evaluated using erasure as ground-truth (Serrano and Smith, 2019; Jain and Wallace, 2019).

Despite its conceptual simplicity, subset erasure is not commonly used in practice. First, it is generally intractable, and beam search (Feng et al., 2018) or leave-one-out estimates (Zintgraf et al., 2017) are typically used instead. These approximations may be inaccurate. For example, leave-one-out can underestimate the contribution of features due to saturation (Shrikumar et al., 2017). More importantly, even these approximations remain very expensive with modern deep (e.g., BERT-based; Devlin et al., 2019) models, as they require multiple computation passes through the model.

Second, the method is susceptible to the hindsight bias: the fact that a feature can be dropped does not mean that the model ‘knows’ that it can be dropped and that the feature is not used by the model when processing the example. This results in over-aggressive pruning that does not reflect what information the model uses to arrive at the decision. The issue is pronounced in NLP tasks (see Figure 2d and Feng et al., 2018), though it is easier to see on an artificial example (Figure 3a). A model is asked to predict if there are more 8s than 1s in the sequence. The erasure attributes the prediction to a single 8 digit, as this reduced example yields the same decision as the original one. However, this does not reveal what the model was relying on: it has counted digits 8 and 1 as otherwise, it would not have achieved the perfect score on the test set.

We propose a new method, Differentiable Masking (DIFFMASK), which overcomes the aforementioned limitations and results in attributions that are more informative and help us understand how the model arrives at the prediction. DIFFMASK relies on learning sparse stochastic gates (a.k.a., masks), guaranteeing that the information from the masked-out inputs does not get propagated while maintaining end-to-end differentiability without having to resort to REINFORCE (Williams, 1992). The decision to include or disregard an input token is made with a simple model based on intermediate hidden layers of the analyzed model (see Figure 1). First, this amortization circumvents the need for combinatorial search making the approach efficient at test time. Second, as with probing classifiers (Adi et al., 2017; Belinkov and Glass, 2019), this reveals whether the network ‘knows’ at the corresponding layer what input tokens can be disregarded. During training inputs are truly masked whenever we sample zeros. After training, attribution scores correspond to the expectation of sampling non-zeros.

The amortization lets us not only plot attribution
heatmaps, as in Figure 2e, but also analyze how decisions are formed across network layers. In our artificial example, we see that in the bottom embedding layer the model cannot discard any tokens, as it does not ‘know’ which digits need to be counted (Figure 3e, left). In the second layer, it ‘knows’ that these are 8s and 1s, so the rest gets discarded (Figure 3e, right). In question answering (see Figure 8a), where we use a 24-layer model, it takes 13–16 layers for the model to ‘realize’ that ‘Santa Clara Marriott’ is not relevant to the question and discard it. We also adapt our method to measuring the importance of intermediate states rather than inputs. This, as we discuss later, lets us analyze which states in every layer store information crucial for making predictions, giving us insights about the information flow.

Contributions We introduce DIFFMASK, a technique addressing limitations of attribution-based methods (especially erasure and its approximations), and demonstrate that it is stable and faithful to the analyzed models. We then use this technique to analyze BERT-based models fined-tuned on sentiment classification and question answering.

2 Method

We aim to understand how a trained model processes an input (i.e., a sequence of embedded tokens) to produce an output (e.g., a vector of class probabilities). First, for an input \( x = \langle x_1, \ldots, x_n \rangle \), we obtain the output \( y = f(x) \) of the model along with its hidden states \( \langle h^{(0)}, \ldots, h^{(L)} \rangle \), where \( h^{(0)} = x \). We then probe the model using a shallow interpreter network which takes hidden states up to a certain layer \( \ell \) and outputs a binary mask \( z = \langle z_1, \ldots, z_n \rangle \) indicating which input tokens are necessary and which can be disregarded. To assess whether the masked input \( \hat{x} = \langle \hat{x}_1, \ldots, \hat{x}_n \rangle \) is sufficient, we re-feed the model with it and compute the output \( \hat{y} = f(\hat{x}) \). As long as \( \hat{y} \) approximates the original output \( y \) well, we deem the inputs masked by \( z \) unnecessary.

Masking, however, as in multiplication by zero, makes a strong assumption about the geometry of the feature space, in particular, it assumes that the zero vector bears no information. Instead, we replace some of the inputs by a learned baseline vector \( b \), i.e., \( \hat{x}_i = z_i \cdot x_i + (1 - z_i) \cdot b \).

See Figure 1 for an overview. The interpreter model consists of \( \ell + 1 \) classifiers, the \( \ell \)th of which conditions on the stack of hidden states up to \( h^{(\ell)} \) to predict binary ‘votes’ \( v^{(\ell)} = g^{(\ell)}(h^{(0)}, \ldots, h^{(\ell)}) \) towards keeping or masking input tokens. Each classifier is a one-hidden-layer MLP, details and hyperparameters are provided in Appendix A. For a given depth \( \ell \), the interpreter decides to mask \( x_i \) out as soon as \( v^{(k)}_i = 0 \) for some \( k \leq \ell \), i.e., \( z_i = \prod_{k=0}^{\ell} v^{(k)}_i \). That is, in order to deem \( x_i \) unnecessary, it is sufficient to do so based on any subset of hidden states up until \( h^{(\ell)} \).

Clearly, there is no direct supervision to estimate the parameters \( \phi \) of the probe and the baseline \( b \), thus we borrow erasure’s objective: namely, we train the probe to mask-out as many input tokens as possible constrained to keeping \( f(\hat{x}) \approx f(x) \). Since often, the output of \( f \) parameterizes a likelihood (e.g., a categorical distribution), we formulate the constraint in terms of a divergence \( D \) between the two functions’ outputs. We cast this, rather naturally, in the language of constrained optimization.

Objective A practical way to minimize the number of non-zeros predicted by \( g \) is minimizing the \( L_0 \) ‘norm’.\(^2\) Thus, our \( L_0 \) loss is defined as the

\(^2\)\( L_0 \), denoted \( \|z\|_0 \) and defined as \( \#(i | z_i \neq 0) \), is the number of non-zeros entries in a vector. Contrary to \( L_1 \) or \( L_2 \), \( L_0 \) is not a homogeneous function and, thus, not a proper norm. However, contemporary literature refers to it as a norm, and we do so as well to avoid confusion.
total number of positions that are not masked:
\[
L_0(\phi, b|x) = \sum_{i=1}^{n} 1_{[x \neq 0]}(z_i),
\]
where \(1(\cdot)\) is the indicator function. We minimize \(L_0\) for all data-points in the dataset \(\mathcal{D}\) subject to a constraint that predictions from masked inputs have to be similar to the original model predictions:
\[
\min_{\phi,b} \sum_{x \in \mathcal{D}} L_0(\phi, b|x) \\
\text{s.t.} \quad D_m[y||\hat{y}] \leq m \quad \forall x \in \mathcal{D},
\]
where \(\hat{y} = f(\hat{x}), y = f(x)\), and the margin \(m \in \mathbb{R}_{>0}\) is a hyperparameter. Since non-linear constrained optimisation is generally intractable, we employ Lagrangian relaxation (Boyd et al., 2004) optimizing instead
\[
\max_{\lambda} \min_{\phi,b} \sum_{x \in \mathcal{D}} L_0(\phi, b|x) + \lambda(D_m[y||\hat{y}]-m),
\]
where \(\lambda \in \mathbb{R}_{\geq 0}\) is the Lagrangian multiplier.

**Stochastic masks** Our objective poses two challenges: i) \(L_0\) is discontinuous and has zero derivative almost everywhere, and ii) to output binary masks, \(g\) needs a discontinuous output activation such as the step function. A strategy to overcome both problems is to make the binary variables stochastic and treat the objective in expectation, in which case one option is to resort to REINFORCE (Williams, 1992), another is to use a sparse relaxation to binary variables (Louizos et al., 2018; Bastings et al., 2019). As we shall see (we compare the two aforementioned options in Table 2 and discuss them in Section 3.2), the latter proved more effective. Thus we opt to use the Hard Concrete distribution, a mixed discrete-continuous distribution on the closed interval \([0,1]\). This distribution assigns a non-zero probability to exactly zero while it also admits continuous outcomes in the unit interval via the reparameterization trick (Kingma and Welling, 2014). We refer to Louizos et al. (2018) for details, but also provide a brief summary in Appendix B. With stochastic masks, the objective is computed in expectation, which addresses both sources of non-differentiability. Note that during training inputs are truly masked-out whenever we sample exact zeros. After training, attribution scores correspond to the expectation of sampling non-zero masks since any non-zero value corresponds to a leak of information.

**Masking hidden states** To reveal which hidden states store information necessary for realizing the prediction, we modify the probe slightly. For a given depth \(\ell\), we use a mask \(z^{(\ell)} = g^{(\ell)}(h^{(\ell)})\) to replace some of the states in \(h^{(\ell)} = (h_1^{(\ell)}, \ldots, h_m^{(\ell)})\) by a layer-specific baseline \(b^{(\ell)}\), i.e. \(\hat{h}_i^{(\ell)} = z_i^{(\ell)} \cdot h_i^{(\ell)} + (1-z_i^{(\ell)}) \cdot b^{(\ell)}\). The resulting state \(\hat{h}^{(\ell)}\) is used to re-compute subsequent states, \(\hat{h}^{(\ell+1)}, \ldots, \hat{h}^{(L)}\), as well as the output, which we denote by \(\hat{y}\). Here we do not aggregate ‘votes’ with a product because for this probe we want to discover whether hidden states are predictive of their own usefulness. See Figure 10 in Appendix D for an overview of this variant of DIFFMASK.

### 3 Experiments

The goal of this work is to uncover a faithful interpretation of an existing model, i.e. revealing, as accurately as possible, the process by which the model arrives at the prediction. Human-provided labels, such as human rationales (Camburu et al., 2018; DeYoung et al., 2020), will not help us in demonstrating this, as humans cannot judge if an interpretation is faithful (Jacovi and Goldberg, 2020). More precisely, human-provided labels do not show how the model behaves – e.g., annotations of what parts of the input are relevant for solving a particular task do not constitute a guarantee that a model relies on those parts more than others when making a prediction. When we evaluate an attribution method by comparing its outputs with human annotations, we are not measuring whether it provides faithful attributions but only if they are plausible according to humans. This goes against our goals as we aim to use the interpretation method to detect model deficiencies, which are usually cases where the model does not behave like humans. The ground-truth explanations of how a model makes certain predictions depend not only on the data but also on the model, and, unfortunately, are generally not known for real tasks and with complex models. This makes the evaluation and comparison of attribution methods non-trivial.

Our strategy is to i) show the effectiveness of DIFFMASK in a controlled setting (i.e., a toy task) where ground-truth is available; ii) test the effectiveness of our relaxation for learning discrete masks (on a real model for sentiment classification); and iii) demonstrate that the method is stable and models behave the same when masking is ap-
which digit occurred the most. The prediction must
which digits to keep. Since the model solves the
task almost perfectly (accuracy on test is >99%).

Table 1: Toy task: attribution to hidden states, average
divergence in nats between the ground-truth attrib-
utions and those by different methods. *The Delta dis-
tribution does not share support with the
ground-truth for the method.

<table>
<thead>
<tr>
<th>Methods</th>
<th>D_{KL} ↓</th>
<th>D_{JS} ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact erasure</td>
<td>–*</td>
<td>0.27</td>
</tr>
<tr>
<td>Sundararajan et al. (2017)</td>
<td>1.32</td>
<td>0.27</td>
</tr>
<tr>
<td>Schulz et al. (2020)</td>
<td>1.12</td>
<td>0.18</td>
</tr>
<tr>
<td>Guan et al. (2019)</td>
<td>0.88</td>
<td>0.24</td>
</tr>
<tr>
<td>DIFFMASK</td>
<td>0.01</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 1: Toy task: attribution to hidden states, average
divergence in nats between the ground-truth attrib-
utions and those by different methods. *The Delta dis-
tribution does not share support with the
ground-truth for the method.

plied. Once we have established that DIFFMASK
can be trusted, we use it to analyze BERT-based
models (Devlin et al., 2019) fine-tuned on senti-
ment classification, and on question answering. We
report hyperparameters in Appendix C, and addi-
tional plots, examples and analysis in Appendix D.

3.1 Toy task

Our toy task is defined as: given a sequence $x$ of
digits (i.e., $x_i \in \{0, \cdots, 9\}$), and a query $(n, m)$
of two digits, determine whether $\#n > \#m$ in $x$.

Model The query and input are embedded, con-
catenated, and then fed to a single-layer feed-
forward NN, followed by a single-layer unidirec-
tional GRU (Cho et al., 2014). The classifica-
tion is done by a linear layer that acts on the last hidden
state of the GRU. See Appendix C.1 for all hy-
perparameters and a more precise definition of the
architecture. Unsurprisingly, the model solves the
task almost perfectly (accuracy on test is > 99%).

Ground-truth for hidden-state attribution We
plot the distribution of hidden states (we use dimen-
sionality 2, with the purpose of having a bottleneck
and to support clear visualization) and observe a
linear separation between states of digits present in
the query and states not in the query. This means
that the role of the feed-forward layer is to decide
which digits to keep. Since the model solves the
task, the role of the GRU must then be to count
which digit occurred the most. The prediction must
be attributed uniformly to all the hidden states cor-
responding to either $n$ or $m$. For completeness,
Figure 11 in the Appendix D.1 shows this plot.

Results We start with an example of input attrib-
utions, see Figure 3, which illustrates how DIFF-
MASK goes beyond input attribution as typically
known. The attribution provided by erasure (Fig-
ure 3a) is not informative: for each datapoint the
search always finds a single digit that is sufficient
to maintain the original prediction and discards all the
other inputs. The perturbation methods by Schulz
et al. (2020) and Guan et al. (2019) (Figure 3b
and 3d) are also over-aggressive in pruning. They
assign low attribution to some items in the query
even though those had to be considered when mak-
ing the prediction. Differently from other methods,
DIFFMASK reveals input attributions conditioned
on different levels of depth. Figure 3e shows both
input attributions according to the input itself and
according to the hidden layer. It reveals that at the
embedding layer there is no information regarding
what part of the input can be erased: attribution is
uniform over the input sequence. After the model
has observed the query, hidden states predict that
masking input digits other than $n$ and $m$ will not
affect the final prediction: attribution is uniform
over digits in the query. This reveals the role of
the feed-forward layer as a filter for positions rel-
vant to the query. Other methods do not allow
for this type of inspection. These observations are
consistent across the entire test set.

For attribution to hidden states (i.e., the output
of the feed-forward layer) we can compare meth-
ods in terms of how much their attributions resem-
ble the ground-truth across the test set. Table 1
shows how the different approaches deviate from
the gold-truth in terms of Kullback-Leibler ($D_{KL}$)
and Jensen–Shannon ($D_{JS}$) divergences.5

3.2 Sentiment Classification

We turn now to a real task and analyze models fine-
tuned for sentiment classification on the Stanford
Sentiment Treebank (SST; Socher et al., 2013).

Erasure search as learning masks Before div-
ing into an analysis of a BERT sentiment model,
we would like to demonstrate that we can approxi-
mate the result of erasure well through our differen-
tiable relaxations. For that, we train a single-
layer GRU sentiment classifier and compare the
analyses by DIFFMASK to solutions provided by

---

4To enable comparison across methods, the attributions
in this Section are normalized between 0 and 1.
5We use $D_{KL}[p||q]$ and $D_{JS}[p||q]$ where $p$ is the ground-
truth distribution and $q$ is the predicted attribution distribution.
Table 2: Sentiment classification: optimization with DIFFMASK and REINFORCE (not amortised – with a moving average baseline for variance reduction) vs. erasure with exact search. All metrics are computed at token level; optimality is measured at sentence level.

<table>
<thead>
<tr>
<th>Metric</th>
<th>REINFORCE+</th>
<th>DIFFMASK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>74.69</td>
<td>81.26</td>
</tr>
<tr>
<td>Recall</td>
<td>80.82</td>
<td>85.89</td>
</tr>
<tr>
<td>F₁</td>
<td>73.57</td>
<td>80.75</td>
</tr>
<tr>
<td>Optimality</td>
<td>8.83</td>
<td>32.67</td>
</tr>
<tr>
<td>$L₀$</td>
<td>33.13</td>
<td>30.58</td>
</tr>
</tbody>
</table>

Faithfulness and Plausibility

Now, we get back to the fully-amortized DIFFMASK approach applied to a 12-layers BERTBASE model and verify that there is no performance degradation when applying masking. Training hyperparameters are reported in Appendix C.2. The $F₁$ score of the model on the validation set moved from 37.9% to 38.3% while masking 46.3% input tokens, and to 38.9% while masking 67.6% hidden states. The explanations provided by DIFFMASK are also stable. Across 5 runs with different seeds, the standard deviation of input attributions are 0.05 and 0.03 for inputs and hidden states, respectively.

While we cannot use human labels to evaluate faithfulness of our method, comparing them and DIFFMASK attribution will tell us whether the sentiment model relies on the same cues as humans. Specifically, we compare to SST token level annotation of sentiment. In Figure 4a, we show after how many layers on average an input token is dropped, depending on its sentiment label. This suggests that the model relies more heavily on strongly positive or negative words and, thus, is generally consistent with human judgments (i.e., plausible).

Analysis

We used DIFFMASK to analyse the behavior of our BERT model. In Figure 5, we report the average number of layers that input tokens or hidden states are kept for (or, equivalently, after how many layers they are dropped on average), aggregating by part-of-speech tags (PoS). It turns out that determinants, punctuation, and pronouns can be completely discarded from the input across all validation set, while adjectives and nouns should be kept. Also the [CLS] and [SEP] tokens can be ignored indicating that the model does not need such markers. Examining the POS tags distribution for hidden states leads to further conclusions. Here, the [CLS] and [SEP] tokens are the most important ones. This is not surprising as the classifier on top of BERT uses the [CLS] hidden state which gets progressively updated through all layers. Both these special tokens are not important as inputs because BERT can infer these markers in other layers, however, they are heavily used in the computation.

Figure 6e we show a visual example of that. We see that the model, even in the bottom layers, knows that the punctuation and both separators can be dropped from the input. This contrasts with hidden states attribution (Figure 6f) which indicates that the separator states (especially [SEP]) are very important. By putting this information together, we can hypothesize that the separator is used to aggregate information from the sentence, relying on self-attention. In fact, this aggregation is still happening in layer 12; at the very top layers, states corresponding to almost all non-separator tokens can be dropped.

Comparison to other methods

In Figure 6, we visually compare different techniques on one example form validation set. While previous techniques (e.g., integrated gradient) do not let us test what a model ‘knows’ in a given layer (i.e. attribution to input conditioned on a layer), they can be used to perform attribution to hidden layers. All methods except attention correctly highlight the last hidden...
state of the [CLS] token as important. Its importance is due to the top-level classifier using the [CLS] hidden state. Although for DIFFMASK we show the expectation of keeping states, it assigns much sharper attributions. For instance, on the validation set, it assigns to the last hidden state of the [CLS] the biggest attribution 99% of the times where Schulz et al. (2020) only 71%. Raw attention (Figure 6a) does not seem to highlight any significant patterns in that example except that start and end of sentence tokens ([CLS] and [SEP], respectively) receive more attention than the rest.\footnote{Voita et al. (2019b) and Michel et al. (2019) pointed out that many Transformer heads play no or minor role.} Attributions by Schulz et al. (2020) and Guan et al. (2019) assign slightly higher importance to hidden states corresponding to ‘highly’ and ‘enjoyable’, whereas it is hard to see any informative patterns provided by integrated gradient. Notice that for DIFFMASK, a near-zero attribution has a very clear interpretation: such a state is not used for prediction since in expectation it is dropped (not gated).

3.3 Question Answering

We turn now to QA where we analyse a fine-tuned BERT\textsubscript{LARGE} model on the Stanford Question Answering Dataset (SQuAD; Rajpurkar et al., 2016).

Analysis We start by asking DIFFMASK which tokens does the model keep? We do a similar analysis as for sentiment classification of POS tags over the entire validation set. We summarize the results in Figure 14 in Appendix D.2. It turns out that conjunctions and adpositions are dropped by the embedding and first layer, respectively, on average. On the contrary, proper nouns and punctuation are usually predicted to be dropped only after the 14th layer. We argue that due to the pre-training objective, BERT could infer well missing parts of the input, especially if they are trivial to infer (e.g., as often the case for prepositions). On the contrary, nouns and proper nouns are important as they count for 84% of the answers on SQuAD. For example, in Figure 8a, we can see that it takes 13–16 layers for the model to ‘realize’ that ‘Santa Clara Marriot’ is not relevant to the question and discard it.

Unlike in sentiment classification, separator tokens as well as punctuation assume a central role as inputs (i.e., punctuation is considered the most important POS tag as for both questions and passages is usually dropped after the 17th layer). Punctuation serves to demarcate sentence boundaries, useful for QA but not for sentiment classification.

Tokens from questions are generally masked by higher layers than tokens from passages as we show in Figure 7a, which suggests that they are more important. We highlight that even in higher
layers when DIFFMASK masks > 95% of the tokens, the original model prediction is almost always kept > 90%. Noticeably, when the original BERT makes wrong predictions, the tokens annotated as the ground truth answer are kept ~60% of the time. This may suggest that when this happens the model still considers other options (e.g., valid options such as the ground truth) as plausible, thus DIFFMASK detects them as important.

Now, we inspect hidden states attributions to answer where is the information stored? In Figure 7b we can see a similar trend as for masking input, i.e., question’s hidden states are kept more on average and deeper in the computation. States on layers 2–3 are dropped less than from the embedding and first layer. This is consistent with findings of Voita et al. (2019a) which show that frequent tokens, such as determiners, accumulate contextual information. However, they are not important as inputs as we show in an example in Figure 8b.

The hidden states corresponding to separator tokens are always kept across all layers except the last one across the validation set. Notice that, this token is also used as a delimiter between the question and the passage, and hence indicates where questions as well as passages end.

The level of hidden states pruning is quite incremental (after layer 3) and gets strong, after layer 9 more than 50% of them can be masked out. A steep increase in superfluous states 13–14 (visible on both parts of Figure 7) may indicate that some states, at that point in computation, contain enough information needed for the classification while all the others can indeed be removed without affecting the model prediction. Our observation that higher layers are more predictive is in line with findings of Kovaleva et al. (2019). They pointed out that the final layers of BERT change most and are more task-specific. Again, the fact that states correspond-

Figure 7: QA: average expectation of keeping input (a) and hidden states (b) from different layers.

Figure 8: QA: attribution the inputs (a) and hidden states (b). The correct answers is highlighted in bold.

Comparison to other methods As we do not have access to the ground-truth, we start by contrasting DIFFMASK qualitatively to other attribution methods on a few examples. We highlight some common pitfalls that afflict other methods (such as the hindsight bias) and how DIFFMASK overcomes those. This helps demonstrate our method’s faithfulness to the original model.

Figure 2 shows input attributions by different methods on an example from the validation set. Erasure (Figure 2d), as expected, does not provide useful insights, it essentially singles out the answer discarding everything else including the question. This cannot be faithful and is a simple consequence of erasure’s hindsight bias: when only the span that contains the answer is presented as input, the model predicts that very span as the answer, but this does not imply that the model ignores everything else when presented with the complete document as input. The methods of Schulz et al. (2020) and Guan et al. (2019) optimize attributions on single examples and thus also converge to assigning high importance mostly to words that support the current prediction and that indicate the question type. Integrated gradient does not seem to highlight any
discernible pattern, which we speculate is mainly because a zero baseline is not suitable for word embeddings. Choosing a more adequate baseline is not straightforward and remains an important open issue (Sturmfels et al., 2020). Note that, DIFF-MASK without amortization (Figure 2f) resembles erasure (as shown in §3.2 for SST).

Differently from all other methods, our DIFF-MASK probes the network to understand what it ‘knows’ about the input-output mapping in different layers. In Figure 2e we show the expectation of keeping input tokens conditioned on any one of the layers in the model to make such predictions (see Figure 8a for a per-layer visualization). Our input attributions highlight that the model, in expectation across layers, wants to keep words in the question, the predicate ‘practice’ in both sentences as well as all potential candidate answers (i.e., named entities). But eventually, the most important spans are in the question and the answer itself.

4 Related Work

While we motivated our approach through its relation to erasure, an alternative way of looking at our approach is considering it as a perturbation-based method. This recently introduced class of attribution methods (Ying et al., 2019; Guan et al., 2019; Schulz et al., 2020; Taghanaki et al., 2019), instead of erasing input, injects noise. Besides back-propagation and attention-based methods discussed in the introduction, another class of interpretation methods (Murdoch and Szlam, 2017; Singh et al., 2019; Jin et al., 2020) builds on prior work in cooperative game theory (e.g., Shapley value of Shapley, 1953). These methods are not trivial to apply to a new model, as they are architecture-specific. Their hierarchical versions (e.g., Singh et al., 2019; Jin et al., 2020) also make a strong assumption about the structure of interaction (e.g., forming a tree) which may affect their faithfulness. Also Chen et al. (2018) share some similarities to our work as they also do amortization but use the Gumbel softmax trick (Maddison et al., 2017; Jang et al., 2017) to approximate minimal subset selection. They assume that the subset contains exactly \( k \) elements where \( k \) is a hyperparameter. Moreover, their explainer is a separate model predicting input subsets, rather than a ‘probe’ on top of the model’s hidden layers, and hence cannot be used to reveal how decisions are formed across layers.

A large body of literature analyzed BERT and Transformed-based models. For example, Tenney et al. (2019) and van Aken et al. (2019) probed BERT layers for a range of linguistic tasks, while Hao et al. (2019) analyzed the optimization surface. Rogers et al. (2020) provides a comprehensive overview of recent BERT analysis papers.

There is a stream of work on learning interpretable models by means of extracting latent rationales (Lei et al., 2016; Bastings et al., 2019). Some of the techniques underlying DIFF-MASK are related to that line of work. They employ stochastic masks to learn an interpretable model, which they train by minimizing a downstream loss subject to constraints on \( L_0 \), whereas we employ stochastic masks to interpret an existing model, and for that, we minimize \( L_0 \) subject to constraints on that model’s output distribution. In our very recent work Schlichtkrull et al. (2020), we also employ stochastic masks and \( L_0 \) regularization for analyzing graph neural networks. We learn which edges are relevant in multi-hop question answering and graph-based semantic role labeling (Marcheggiani and Titov, 2017; De Cao et al., 2019).

5 Conclusion

We have introduced a new post hoc interpretation method which learns to completely remove subsets of inputs or hidden states through masking. We circumvent an intractable search by learning an end-to-end differentiable prediction model. To overcome the hindsight bias problem, we probe the model’s hidden states at different depths and amortize predictions over the training set. Faithfulness is validated in a controlled experiment pointing more clearly to some flaws of other attribution methods. We used our method to study BERT-based models on sentiment classification and question answering. DIFF-MASK sheds light on what different layers ‘know’ about the input and where information about the prediction is stored in different layers.

Acknowledgements

The authors want to thank Christos Baziotis, Elena Voita, Diewu Hupkes, and Naomi Saphra for helpful discussions. This project is supported by SAP Innovation Center Network, ERC Starting Grant BroadSem (678254), the Dutch Organization for Scientific Research (NWO) VIDI 639.022.518, and the European Union’s Horizon 2020 research and innovation programme under grant agreement No 825299 (Gourmet).
References


