Integrated Directional Gradients: Feature Interaction Attribution for Neural NLP Models

Sandipan Sikdar*
RWTH Aachen University
sandipan.sikdar@cssh.rwth-aachen.de

Parantapa Bhattacharya*
University of Virginia
parantapa@virginia.edu

Kieran Heese
University of Virginia
kh8fb@virginia.edu

Abstract

In this paper, we introduce *Integrated Directional Gradients* (IDG), a method for attributing importance scores to groups of features, indicating their relevance to the output of a neural network model for a given input. The success of Deep Neural Networks has been attributed to their ability to capture higher level feature interactions. Hence, in the last few years capturing the importance of these feature interactions has received increased prominence in ML interpretability literature. In this paper, we formally define the feature group attribution problem and outline a set of axioms that any intuitive feature group attribution method should satisfy. Earlier, cooperative game theory inspired axiomatic methods only borrowed axioms from solution concepts (such as Shapley value) for individual feature attributions and introduced their own extensions to model interactions. In contrast, our formulation is inspired by axioms satisfied by *characteristic functions* as well as solution concepts in cooperative game theory literature. We believe that characteristic functions are much better suited to model importance of groups compared to just solution concepts. We demonstrate that our proposed method, IDG, satisfies all the axioms. Using IDG we analyze two state-of-the-art text classifiers on three benchmark datasets for sentiment analysis. Our experiments show that IDG is able to effectively capture semantic interactions in linguistic models via negations and conjunctions.

1 Introduction

In the last decade Deep Neural Networks (DNN) have been immensely successful. Much of this success can be attributed to their ability to learn from complex higher order interactions from raw features (Goodfellow et al., 2016). This success of DNNs has led to them being increasingly adopted for algorithmic decision making. This in turn has led to increasing concerns over explainability and interpretability of these models, given the important role they are beginning to take in society (Selbst and Barocas, 2018).

One area of work that has emerged in recent years is that of black box model explanation strategies that "explain" the output of a DNN for a given input using feature attribution scores or saliency maps (Sundararajan et al., 2017; Shrikumar et al., 2017). Numerous studies have been published in recent years proposing different strategies to answer the question "which features in the input were most important in deciding the output of the DNN?" However, modern DNNs take as input raw data as features, and learn from higher order interaction of those features. Thus in the past year a number of studies have instead focused on explaining feature interactions rather than explaining individual features (Chen and Jordan, 2020; Jin et al., 2019; Sundararajan et al., 2020; Chen et al., 2020; Tsang et al., 2020).

One issue that remains, however, is that given two methods for attributing importance scores, it is not entirely straight forward to objectively compare them. As has been noted by earlier studies (Sundararajan et al., 2017), if the output of an attribution method seems non-intuitive it is not easy to answer if that is caused by (i) limitations of the attribution method, (ii) limitations of the DNN model being explained, or (iii) limitation of the data on which the DNN model was trained. Like multiple previous studies (Chen and Jordan, 2020; Sundararajan et al., 2020; Tsang et al., 2020) we take an axiomatic approach to this problem, whereby we first define the set of properties/axioms that a "good" solution must satisfy, followed by development of a solution that satisfies those axioms.

The method for computing feature group attribution (interchangeably referred to as feature inter-

*Equal contribution

action attribution) presented in this study is called *Integrated Directional Gradients* or IDG. Like multiple earlier methods in this area, IDG is a co-operative game theory inspired method. However, unlike earlier cooperative game theory inspired axiomatic methods which only borrowed axioms from solution concepts (such as Shapley value) for individual feature attributions and introduced their own extensions to model interactions, our formulation is inspired by axioms satisfied by well behaved *characteristic functions* as well as solution concepts in cooperative game theory literature. We find that well behaved *characteristic functions* provide a much simpler and intuitive framework for defining axioms for group attributions.

We apply IDG on state-of-the-art models on the NLP domain. As part of its input IDG requires a set of meaningful feature sets, that have a hierarchical structure (Section 2.1). In this paper we use parse tree of sentences to construct the meaningful feature structures. Figure 1 shows an illustrative example of the nature of explanations and attributions computed using IDG.

The major contributions of the current work are as follows:

- First, we formally define the feature group attribution problem as an extension to the feature attribution problem (Section 2.1).
- Second, we state a set of axioms that a well behaved feature group attribution method should satisfy (Section 2.2).
- Third, we present the method of *Integrated Directional Gradients* or IDG as a solution to the feature group attribution problem that satisfies the stated axioms (Section 2.3).
- Fourth, we propose an efficient algorithm to compute IDG for a given set of feature groups with a hierarchical structure (Section 2.4).
- Finally, we compare IDG with other recently proposed related methods for computing feature interactions attribution. (Section 3).
- To facilitate reproducibility, the implementation of IDG has been made publicly available[1].

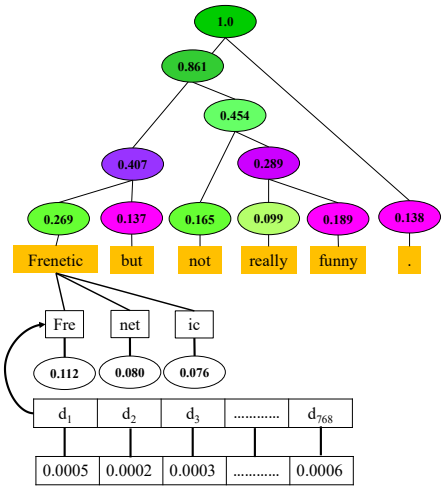


Figure 1: Computation of attribution score (value function v) for an example sentence `Frenetic but not really funny`. Magenta and green respectively denote negative or positive contribution to the inferred class and the importance is represented by the color intensity. Constituency parse tree is used to obtain meaningful feature groups. Note that each word is further divided into tokens (owing to byte pair encodings) each of which has 768 dimensions. IDG computes importance scores in a bottom-up manner starting from the individual embedding dimensions (d_i) working its way up to tokens, words, phrases and finally the sentence.

2 Methodology

2.1 Problem Definition

In this section we formally state the problem of assigning attribution scores to meaningful feature groups.

Let $f(x)$ be a deep neural network function, that takes as input a n dimensional real valued vector $x \in \mathbb{R}^n$ and produces a real valued scalar output. Let $A = \{a_1, a_2, \ldots, a_n\}$ refer to the set of features, with x_i referring to the value of feature a_i in feature vector x.

Then the **feature group attribution problem** is defined as follows: Given an input x, a baseline $b \in \mathbb{R}^n$, and a family of meaningful feature subsets $M \subseteq \mathcal{P}(A)$, assign to every subset of features $S \subseteq A$ a value/importance score $v(S)$. Here, $\mathcal{P}(A)$ represents the power set of the feature set.

The above formulation is inspired by cooperative game theory literature. Intuitively, we think of features as players in a co-operative game trying to "help" the DNN model reach its output. The objective then is to design a "good" value/importance function (characteristic function in cooperative game theory literature) for each feature subset (coalition of players).

Note that the above formulation is very different from existing cooperative game theory inspired feature attribution methods. Most existing methods assume that the value/characteristic function exists

[1] https://github.com/parantapa/integrated-directional-gradients

and then compute a payoff assignment vector for individual features, typically using Shapley values.

Similar to earlier studies, in our formulation we assume that the baseline b represents the "zero" input or absence of contribution from any feature.

The "family of meaningful feature subsets" M captures the notion that not all subsets of features represent "meaningful" parts of input. Another intuitive way to think about this is that not all features can collaborate directly, but need to be part of groups that can directly collaborate.

In general we will assume that M has a hierarchical containment structure, that is feature groups in M can be represented as a directed acyclic graph — with tree being a special case. Further, we will also assume that every individual feature is in M — that is $\{a_i\} \in M$ for $i \in \{1, 2, \ldots, n\}$ — and represents the leaf nodes in the hierarchy, while the set of all features is also in M — that is $A \in M$ and represents the root of the hierarchy.

2.2 Solution Axioms

In this section we present a set of axioms that a well behaved value/importance function should satisfy. Note that, the following four axioms are variants of standard axioms for characteristic functions in cooperative game theory literature.

Axiom 1 (Non-Negativity) *Every feature subset has a non-negative value,* $v(S) \geq 0$.

Axiom 2 (Normality) *The value of the empty set of features is zero,* $v(\emptyset) = 0$.

Axiom 3 (Monotonicity) *The value of a set of features is greater than or equal to the value of any of its subsets; if* $S \subseteq T$, *then* $v(S) \leq v(T)$.

Axiom 4 (Superadditivity) *The value of the union of two disjoint sets of features is greater than or equal to the sum of the values of the two sets; if* $S \cap T = \emptyset$ *then* $v(S \cup T) \geq v(S) + v(T)$.

Since the value function represents the importance of a set of features, which is intuitively a direction less quantity, the Non-Negativity axiom ensures that every feature has a non-negative value/importance score. Similarly, the Normality axiom ensures that the importance score assigned to the empty set of features is zero. Since in the current framework the features in a deep neural network "collaborate", with the assumption that collaboration can only be beneficial, the axioms of Monotonicity and Superadditivity ensure that collaboration doesn't lead to diminished value/importance. Note that Superadditivity together with Non-Negativity implies Monotonicity.

In a cooperative game, players cooperate to generate the maximum value. A sometimes implicit assumption in these games is that it is always possible for a player to do nothing, in which case they generate zero value. Thus if doing something generates negative value a rational player will always choose to do nothing. This is the essence of Axiom 1. In axiomatic ML explanation literature, features are thought of as players cooperating to predict the output. One can also think of the value provided by a feature (importance of the feature) as the information contained in the feature that is effectively used by the model. This view also supports assumption of Axiom 1 as quantities of information (entropy) is also a non-negative quantity.

Axioms 1–3 are some of the foundational axioms of cooperative game theory (Chalkiadakis et al., 2011). While much mathematical theory has been published for computing solution concepts in games where these assumptions do not hold, we argue that those games themselves can be difficult to interpret and thus are less suitable for developing interpretability/explainability methods.

The following three axioms are variations of axioms of the same name presented in the (Sundararajan et al., 2017). The modifications presented here are necessary to incorporate the complexities resulting from assigning attribution scores to groups of features rather than individual features.

Axiom 5 (Sensitivity (a)) *Let there be a feature* a_i *such that,* $f(x) \neq f(b)$ *for every input feature vector* x *and baseline vector* b *that only differ in* a_i. *Then* $v(\{a_i\}) > 0$ *and* $v(S) > 0$ *for every set of features* S *such that* $a_i \in S$.

Axiom 6 (Sensitivity (b)) *Let there be a feature* a_j *such that,* $f(x) = f(b)$ *for every input feature vector* x *and baseline vector* b *that only differ in* a_j. *Then* $v(\{a_j\}) = 0$ *and* $v(S) = v(S \setminus \{a_j\})$ *for every set of features* S *such that* $a_j \in S$.

In essence the axiom Sensitivity (a) ensures that features that *does effect* the output of the DNN are not assigned a zero value/importance. Consequently, any feature group that includes such a feature must also be assigned a non-zero value. Conversely, the axiom Sensitivity (b) ensures that any feature that *does not effect* the output of the DNN is assigned a zero value, and that it doesn't contribute any value to any feature group that it is included in.

Axiom 7 (Symmetry Preservation) *Two features a_i and a_j are said to be functionally equivalent if $f(x) = f(y)$ for every pair of input vectors x and y such that $x_i = y_j$, $x_j = y_i$, and $x_k = y_k$ for $k \notin \{i, j\}$. Two features a_i and a_j are said to be structurally equivalent with respect to a family of meaningful feature subsets M if $a_i \in S$ and $S \neq \{a_i\}$ implies $a_j \in S$ for all feature subsets $S \in M$ and vice versa.*
If two features a_i and a_j are both functionally and structurally equivalent and if the given input vector x and baseline vector b are such that $x_i = x_j$ and $b_i = b_j$ then $v(S \cup \{a_i\}) = v(S \cup \{a_j\})$ for every subset of features $S \subseteq A \setminus \{a_i, a_j\}$.

The Symmetry Preservation axiom first defines two different types of feature equivalence: functional and structural. Two features are said to be functionally equivalent if swapping the values of those features doesn't effect the output of the DNN. Where as structural equivalence of features on the other hand refers to them having equivalent position in the structure imposed by the set of meaningful features M. Finally, the Symmetry Preservation axiom ensures that features that are both functionally and structurally equivalent contribute equal value/importance to all feature subsets they are included in.

Axiom 8 (Implementation Invariance) *Two neural networks $f'()$ and $f''()$ are functionally equivalent if $f'(x) = f''(x)$ for all x. Let the value functions for them be denoted by $v'()$ and $v''()$ respectively. Then $v'(S) = v''(S)$ for all subset of features $S \subseteq A$.*

The Implementation Invariance axiom simply ensures that different implementations of the same DNN function result in same value/importance assignment to all feature subsets.

2.3 Our Method: Integrated Directional Gradients

In this section we present a solution to the "feature group attribution problem" that we call the Integrated Directional Gradients method or IDG. This method is inspired by the Integrated Gradients method (Sundararajan et al., 2017) and by Harsanyi dividends (Harsanyi, 1963) in cooperative game theory. The high level idea of the method is to construct the value function in terms of the "dividends" generated by each meaningful feature subset. In this formulation, each meaningful feature group contributes "additional value" to the DNN model, that we call "dividend" of the group. The dividend of a feature group S is represented by $d(S)$ and $d(S) \in [0, 1)$.

The dividend of a single feature is also its value and a measure of its importance. One of the simplest measures of importance of a feature is the partial derivative of the DNN function with respect to the feature. The partial derivative also has an intuitive notion that it represents the amount of change in the output of the DNN function per unit change in the input, in the direction of the feature. However, as noted in the earlier studies (Sundararajan et al., 2017), due to effects such as gradient saturation, partial derivatives can't be directly used for measuring the importance of a feature. To alleviate this issue the authors of the Integrated Gradients method recommend taking a path integral of the partial gradient over the straight line path connecting the baseline b to the input x. For this study, we take a similar approach, and take the absolute value of the path integral of the partial gradient as the dividend of a single feature.

The dividend of a group of features is distinct from its value and is the measure of the importance of the interaction of the features in the group. For this study we consider the directional derivative of the DNN function in the direction of the given set of features to be representative of the importance of the interaction of the given set of features. Similar to the single feature case this also has the intuitive notion that it represents the amount of change in the output of DNN function per unit change in input, in the direction of the subset of features. However, as in the case with single features, issues such as gradient saturation still need to be addressed for directional gradients as well. Thus we propose to use absolute value of IDG, which is the path integral of the directional gradient over the straight line path from the baseline b to the input x as the dividend of the feature group. Further, the sign of IDG may be used to signify the nature of contribution (positive or negative) to model output.

$$z_i^s = \begin{cases} x_i - b_i & \text{if } a_i \in S \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$\nabla_S f(x) = \nabla f(x) \cdot \hat{z}^s \quad \text{where } \hat{z}^s = \frac{z^s}{\|z^s\|} \tag{2}$$

$$\text{IDG}(S) = \int_{\alpha=0}^{1} \nabla_S f\left(b + \alpha(x - b)\right) \mathrm{d}\alpha \tag{3}$$

$$d(S) = \begin{cases} \frac{|\text{IDG}(S)|}{Z} & \text{if } S \in M \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$$Z = \sum_{S \in M} |\text{IDG}(S)| \quad (5)$$

$$v(S) = \sum_{T \in \{T | T \subseteq S \wedge S \in M\}} d(T) \quad (6)$$

Equations 1 to 6 describe the process of computing the value/importance $v(S)$ of a subset of features using the IDG method. Given a feature subset S first the feature subset difference vector z^s is computed from the input feature vector x and the baseline vector b. Next, $\text{IDG}(S)$ is computed by integrating over the directional derivative, in the direction of z^s over the straight line path from the baseline b to the input x. The dividend $d(S)$ of the feature subset S is then computed by normalizing the absolute value of $\text{IDG}(S)$ over all meaningful subsets, such that the sum of the dividends of all meaningful features subsets add up to 1. Finally the value $v(S)$ of the given feature subset S is computed by adding up the dividends of all the meaningful subsets contained in S, including itself.

Proposition 1 *$v(s)$ satisfies axioms 1 to 8*[2].

2.4 Efficiently computing Integrated Directional Gradients

Similar to (Sundararajan et al., 2017), we approximate the integral in IDG, by simply summing over the gradients at points occurring at small intervals along the path from baseline b to the input x. The approximated $\text{IDG}(S)$ is computed as:

$$\text{AIDG}(S) = \frac{1}{m+1} \sum_{k=0}^{m} \nabla_S f \left(b + \frac{k}{m}(x - b) \right) \quad (7)$$

Here m denotes the number of steps in the Reimann approximation of the integral. We now propose a polynomial time dynamic programming Algorithm (1) for calculating the attribution score (i.e., value function v) for all the meaningful subsets in M for a given input x and a baseline b.

First, ∇f is calculated for each of the $m+1$ intermediate positions between x and b. Next we compute $\text{AIDG}(S)$ for all feature groups in M. This is followed by the computation of Z, which is simply the sim of the $AIDG(S)$ scores for reach of the meaningful subsets. Given Z and the individual scores the divided $d(S)$ can easily be computed using Eq. 4. Finally, given the dividend of all meaningful subsets of S is known, the value function $v(S)$ for each of the meaningful subsets of S can be computed using Eq. 6.

Algorithm 1

procedure COMPUTEATTRIBUTION(x, b, M, m)
  for $k \in \{0, 1, \ldots, m\}$ **do**
    Compute $\nabla f(b + \frac{k}{m}(x - b))$
  end for
  for $S \in M$ **do**
    Compute $\text{AIDG}(S)$ ▷ Using Eq. 7
  end for
  $Z \leftarrow \sum_{S \in M} |\text{AIDG}(S)|$
  for $S \in M$ **do**
    Compute $d(S)$ ▷ Using Eq. 4
  end for
  for $S \in M$ **do**
    Compute $v(S)$ ▷ Using Eq. 6
  end for
end procedure

We illustrate the computation of attribution scores using an example sentence `Frenetic but not really funny` taken from SST dataset (Figure 1). The task is sentiment classification and the inferred class for this sentence is negative. The model used for classification is XLnet-base (refer to Section 3 for details on dataset, model and training procedure). We leverage the constituency parse tree of the sentence to obtain meaningful feature groups. Note that XLnet tokenizer uses byte pair encoding. Hence the word "Frenetic" is further decomposed into "Fre", "net" and "ic". Each token is further represented by an embedding of size 768. The value function is calculated in a bottom-up manner starting from each embedding dimension of the constituent tokens (referred as d_i in Figure 1). These are then combined to obtain the value function score for each token. We then follow the parse tree to calculate the score for each phrase. For example the score for phrase `Frenetic but` is 0.407 while that of `not really funny` is 0.454.

The overall time complexity of Algorithm 1 is $O\left(m(F + B + V \cdot |A|) + V + E\right)$, where F and B are the time complexity of a single forward and backward pass of the neural network, V and E are, respectively, the number vertices and edges in the graph structure induced by the family of meaningful feature subsets M, $|A|$ is the number of features, and m the number of approximation steps used to compute $\text{AIDG}(S)$. For more details

[2]Detailed proofs are available in Appendix.

on the complexity result, refer to Appendix.

3 Evaluation

3.1 Comparison with existing methods

It has been noted that when a DNN explanation method returns a non-intuitive result, it is not possible to disentangle which part of the pipeline — training data, trained model, or the explanation method — is to blame for the result (Sundararajan et al., 2017). Thus many studies (Sundararajan et al., 2017; Chen and Jordan, 2020; Sundararajan et al., 2020; Tsang et al., 2020) have taken the axiomatic strategy instead to compare methods qualitatively. Taking a similar approach, we present in Table 1 a qualitative comparison of recent feature interaction attribution methods most similar to our work.

We group the comparison into four major categories. *First*, in most cooperative game theory literature players are assumed to cooperate. It is thus intuitive that more cooperation will not lead to lesser benefit, and it is generally assumed that the grand coalition will form (Chalkiadakis et al., 2011). While there are mathematical formulations that work in absence of this assumption, we argue that they lead to non-intuitive results when applied to the task of feature interaction attribution. These assumptions are manifested by well-behavedness properties of the characteristic/value function. In Table 1 we see that existing cooperative game theory inspired methods generally ignore this aspect when computing importance attributions.

Second, to compute the effect of a model in absence of a feature, attribution methods generally mask out the feature, generally replacing it with a ZERO or PAD token. It has been noted that this requires the DNN model to be evaluated in an region of the input space for which it has not received any training data and for which its accuracy was never evaluated (Sundararajan et al., 2017; Kumar et al., 2020). Thus the results that model produces for these out-of-distribution inputs is questionable. In Table 1 we see that all existing methods compute their attributions by evaluating the model for these out-of-distribution inputs.

Third, in a cooperative game theoretic setting when players (here features) are assumed to cooperate, it is intuitive that as the size of the coalition grows the coalition will not become less important. This is the key intuition behind Axioms 1–4. However, In Table 1 we see that none of the existing methods ensure that their attributions adhere to this key intuition.

Finally, cooperative game theory based methods generally ensure that axioms of Completeness (a.k.a. Efficiency), Symmetry Preservation, Linearity, and Sensitivity (a.k.a Null/Dummy player) are warranted by their attributions. In this paper we follow the lead of (Sundararajan et al., 2017) and use the nomenclature from (Aumann and Shapley, 2015), which additionally introduces the axiom of Implementation Invariance. In Table 1, we see that for LS-Tree (Chen and Jordan, 2020), Shapley-Taylor Interaction Index (Sundararajan et al., 2020), and Archipelago (Tsang et al., 2020), which are cooperative game theory inspired methods, these assumptions hold. However for SCD/SOC (Jin et al., 2019) and HEDGE (Chen et al., 2020) which are not axiomatic formulations, these assumptions do not hold. For our method, IDG, all but the axiom of Linearity holds. In Section 5.2 we argue that this is not a major limitation and refer to existing literature that even argues for doing away with the Linearity axiom.

3.2 Evaluating IDG on state-of-the-art models

We deploy our model for the task of sentiment classification across three different datasets - Stanford Sentiment Treebank (SST) (Socher et al., 2013), Yelp reviews (Zhang et al., 2015) and IMDB (Maas et al., 2011). For each dataset, we train three state-of-the-art models - XLnet-base (Yang et al., 2019), XLnet-large (Yang et al., 2019) and BERT-itpt (Sun et al., 2019). We use the same hyperparameter configuration as mentioned in the original papers. They are summarized in Appendix as well. The performance of these models are summarized in Table 2.

4 Results

To precisely visualize the interactions between phrases, we search over the test examples for instances of negations. We follow the methodology proposed in (Murdoch et al., 2018). In specific, we look into the parse tree for each review and check if the left child consists of a negation phrase (e.g., lacks, never etc.) in the first two words and the right child has a positive or a negative sentiment. Since for SST, each phrase is also annotated with their corresponding sentiment labels in the form of a constituency parse tree, this can be easily ob-

Axioms/Properties	SCD/SOC	HEDGE	LS-Tree	STI	Archipelago	IDG
Well-Behaved Characteristic Function	NA	NA	✗	✗	✗	✓
In Distribution Evaluations	✗	✗	✗	✗	✗	✓
Non-Negativity	✗	✗	✗	✗	✗	✓
Normality	✗	✗	✗	✗	✗	✓
Monotonicity	✗	✗	✗	✗	✗	✓
Superadditivity	✗	✗	✗	✗	✗	✓
Sensitivity	✗	✗	✓	✓	✓	✓
Symmetry Preservation	✗	✗	✓	✓	✓	✓
Linearity	✗	✗	✓	✓	✓	✗
Completeness	✗	✗	✓	✓	✓	✓
Implementation Invariance	✗	✗	✓	✓	✓	✓

Table 1: A comparison of axiomatic guarantees / properties of feature interaction attribution methods: SCD/SOC (Jin et al., 2019), HEDGE (Chen et al., 2020), LS-Tree (Chen and Jordan, 2020), Shapley-Taylor Interaction Index (STI) (Sundararajan et al., 2020), Archipelago (Tsang et al., 2020), and IDG (proposed method). Note since $v(\emptyset) = 0$ and $v(A) = 1$, IDG satisfies completeness trivially.

	Test/train split	XLnet-base	XLnet-large	BERT-itpt
SST	6920/872	0.915	0.916	0.769
Yelp	560K/38K	0.979	0.983	0.947
IMDB	25K/25K	0.967	0.967	0.957

Table 2: Accuracy of the trained models on the three datasets.

tained. For Yelp and IMDB, we look for presence of negation phrases in the reviews and then manually select 100 such examples from the filtered set. Since the parse trees for the reviews are not explicitly available for Yelp and IMDB, we deploy a state-of-the-art constituency parser (Mrini et al., 2019) to obtain them.

We illustrate with one example each from SST and Yelp datasets in Figures 2(a) and 2(b) respectively. Additional examples can be found in Appendix. For Figure 2(a) the classification model is XLnet-base and the ground truth as well as the inferred class is negative. The first part (`Though everything might be literate and smart`) has a positive sense. But when appended with the second part (`it never took off and always seemed static`), a negative sense is manifested. This is captured by the classification model as demonstrated by our framework. For the example in Figure 2(b), the classifier model is BERT-itpt and the inferred as well as the ground-truth class is negative. This example consists of two sentences while the first one `Nice atmosphere` has a positive sense, when combined with the second sentence `Cheeseburger was not at all that`, the overall sense turns negative. This is again conveniently manifested in the scores assigned by our framework. We also report the results on IMDB reviews (Maas et al., 2011) in Appendix.

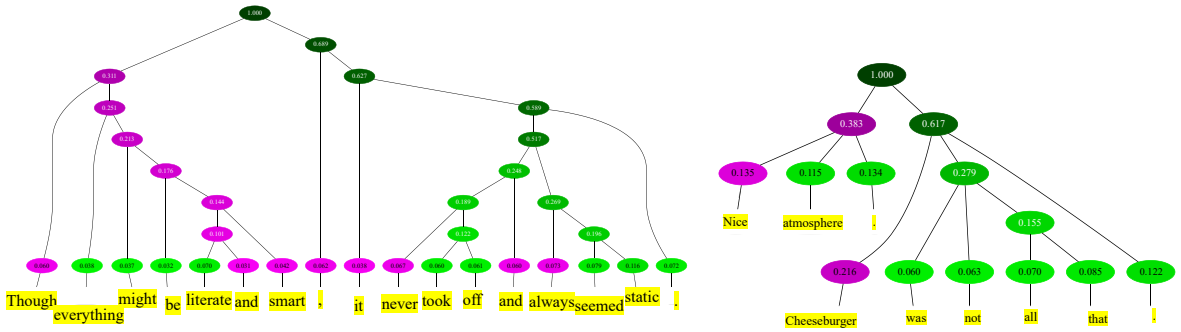
5 Discussion

5.1 Quantitative Evaluations and Human Judgement Experiments

As noted by (Sundararajan et al., 2017), when the results of an explanation method is non-intuitive, it is not obvious which part of the ML pipeline — the data, the model being explained, the explanation method — is to be blamed and by how much. Due to this issue many authors (Sundararajan et al., 2017; Chen and Jordan, 2020; Sundararajan et al., 2020; Tsang et al., 2020) have chosen to take the axiomatic/theoretical path, where they state the properties of the proposed method and compare explanation methods based on the axioms/properties they satisfy.

Nevertheless, many recent studies (Singh et al., 2018; Jin et al., 2019; Chen et al., 2020) have proposed new explanation methods and provided evaluations using quantitative metrics such as AOPC (Nguyen, 2018), Log Odds (Shrikumar et al., 2017), and Cohesion Score (Jin et al., 2019).

One common strategy is to perturb the input — such as removing of Top-K most important words/features — followed by measuring the drop in performance. We argue that these methods of evaluation have issues because they generally involve measuring model performance on out-of-distribution inputs. And as stated earlier, measuring the outputs of models on out-of-distribution inputs, that is inputs, on which the model has neither been trained or tested on, is questionable.



(a) (b)

Figure 2: The value function scores assigned by our framework for different coalitions (interactions) between phrases for two reviews from SST (a) and Yelp (b) respectively. Magenta and green respectively denote negative or positive contribution to the inferred class and the magnitude of importance is represented by the color intensity. Note that the interactions are correctly captured by the classifier model in both the cases as demonstrated by IDG.

The other strategy is to perturb the model — such as by adding noise to model weights — followed by measuring the drop in performance. (Hooker et al., 2019) proposed a similar solution for the input perturbation case as well, that is by retraining the model after perturbing all training samples. However, in this scenario if two explanation methods provided different explanations/attributions for the different models, it is not obvious if the models are to blame or the explanation methods. Similar issues exist for human judgement experiments as well. Due to the above issues for the current work we too have chosen to take the qualitative comparison path.

5.2 Linearity and Uniqueness

One of the common axioms of solution concepts in cooperative game theory is Linearity. The axiom of Linearity (a.k.a Additivity) states that if the characteristic/value function has the form $v(S) = v_1(S) + v_2(S)$ and $\phi_1(S)$ and $\phi_2(S)$ are the attributions due to $v_1(S)$ and $v_2(S)$ then the attribution due to $v(S)$ should be given by $\phi(S) = \phi_1(S) + \phi_2(S)$.

During our design and experimentation we found that having the attributions normalized, that is $v(\emptyset) = 0$ and $v(A) = 1$, provided much more intuitive results. Such normalization, however, runs counter to the possibility of an attribution method that satisfies Linearity.

Further, it has been argued by some game theorists that the axiom of Linearity was added as a mathematical convenience and also to constrain the attributions such that it is unique (Osborne and Rubinstein, 1994). Further, (Kumar et al., 2020) argue that enforcing such uniqueness constraints by this method limits the kind of models that can be explained by these attributions.

Thus, IDG is also not an unique solution to the feature group attribution problem, due to its sacrifice of Linearity. However, given that recent studies have found (Sundararajan and Najmi, 2020) that Shapley values can and have been used in many different ways, each of which claiming uniqueness, the importance of uniqueness claims is significantly diminished.

6 Related work

Feature attribution based method. These methods essentially assign importance scores to individual features thereby explaining the decisions of the classifier model. The scores are mostly calculated by either backpropagating a custom relevance score (Sixt et al., 2020) or directly using the gradients. The gradient based methods aim to calculate the sensitivity of the inference function with respect to the input features and thereby measuring its importance. The method was first introduced in (Springenberg et al., 2015) and further investigated in (Selvaraju et al., 2017; Kim et al., 2019). (Sundararajan et al., 2017) adopts an axiomatic approach and deem it to be more suitable as the feature attribution methods are hard to evaluate empirically. The other set of methods usually backpropagates their custom relevance scores down to the input to identify relevance of an input feature (Bach et al., 2015; Shrikumar et al., 2017; Zhang et al., 2018). Unlike the gradient based methods, these are not implementation invariant

(i.e., the back propagation process is architecture specific).

Game theoretic aspect. (Lundberg and Lee, 2017) adopts results (shapely values in specific) from coalition game theory to obtain feature attribution scores. The key idea is to consider the features as individual players involved in a coalition game of prediction which is considered the payout. The payout then can be fairly distributed among the players (features) to measure their importance. This has been further explored in (Lundberg et al., 2020; Ghorbani and Zou, 2020; Sundararajan and Najmi, 2020; Frye et al., 2020).

Quantifying feature interactions. The methods mentioned above fail to properly capture the importance of feature interaction. (Janizek et al., 2020) proposes to capture pair-wise interaction by building upon Integrated gradients framework. (Cui et al., 2020) learns global pair-wise interactions in bayesian neural networks. (Murdoch et al., 2018) introduces contextual decomposition to capture interaction among words in a text for a LSTM-based classifier. (Singh et al., 2018) further extends the method to other architectures. More recent research endeavors in this direction include (Tsang et al., 2020; Liu et al., 2020; Chen et al., 2020). We elaborate more on the methods closest to our work in section 3.

7 Conclusion

In this paper we investigated the problem of feature group attribution and proposed a set of axioms that any framework for feature group attribution should fulfill. We then introduced IDG, a novel method, as a solution to the problem and demonstrated that it satisfies all the axioms. Through experiments on real-world datasets with state-of-the-art DNN based classifiers we demonstrated the effectiveness of IDG in capturing the importance of feature groups as deemed by the classifier.

Acknowledgements

Sandipan Sikdar was supported in part by RWTH Aachen Startup Grant No. StUpPD384-20. Parantapa Bhattacharya was supported in part by the Dense Threat Reduction Agency (DTRA) under Contract No. HDTRA1-19-D-0007, by the National Science Foundation (NSF) under Grant No. CCF-1918656, and by the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8650-19-C-7923. The authors would also like to thank the Research Computing Center at University of Virginia for compute time grant on the Rivanna cluster.

References

Robert J Aumann and Lloyd S Shapley. 2015. *Values of non-atomic games*. Princeton University Press.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge. 2011. Computational aspects of cooperative game theory. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(6):1–168.

Hanjie Chen, Guangtao Zheng, and Yangfeng Ji. 2020. Generating hierarchical explanations on text classification via feature interaction detection. In *Annual Meeting of the Association for Computational Linguistics*, pages 5578–5593.

Jianbo Chen and Michael Jordan. 2020. Ls-tree: Model interpretation when the data are linguistic. In *AAAI Conference on Artificial Intelligence*, volume 34, pages 3454–3461.

Tianyu Cui, Pekka Marttinen, Samuel Kaski, et al. 2020. Learning global pairwise interactions with bayesian neural networks. In *European Conference on Artificial Intelligence*. IOS Press.

Christopher Frye, Colin Rowat, and Ilya Feige. 2020. Asymmetric shapley values: incorporating causal knowledge into model-agnostic explainability. *Advances in Neural Information Processing Systems*, 33.

Amirata Ghorbani and James Zou. 2020. Neuron shapley: Discovering the responsible neurons. *arXiv preprint arXiv:2002.09815*.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*, volume 1. MIT press Cambridge.

John C Harsanyi. 1963. A simplified bargaining model for the n-person cooperative game. *International Economic Review*, 4(2):194–220.

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*.

Joseph D Janizek, Pascal Sturmfels, and Su-In Lee. 2020. Explaining explanations: Axiomatic feature interactions for deep networks. *arXiv preprint arXiv:2002.04138*.

Xisen Jin, Zhongyu Wei, Junyi Du, Xiangyang Xue, and Xiang Ren. 2019. Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models. In *International Conference on Learning Representations*.

Beomsu Kim, Junghoon Seo, Seunghyeon Jeon, Jamyoung Koo, Jeongyeol Choe, and Taegyun Jeon. 2019. Why are saliency maps noisy? cause of and solution to noisy saliency maps. In *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4149–4157. IEEE.

I Elizabeth Kumar, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle Friedler. 2020. Problems with shapley-value-based explanations as feature importance measures. In *International Conference on Machine Learning*, pages 5491–5500. PMLR.

Zirui Liu, Qingquan Song, Kaixiong Zhou, Ting-Hsiang Wang, Ying Shan, and Xia Hu. 2020. Detecting interactions from neural networks via topological analysis. *Advances in Neural Information Processing Systems*, 33.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. 2020. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1):2522–5839.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774.

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

Khalil Mrini, Franck Dernoncourt, Trung Bui, Walter Chang, and Ndapa Nakashole. 2019. Rethinking self-attention: An interpretable self-attentive encoder-decoder parser. *arXiv preprint arXiv:1911.03875*.

W James Murdoch, Peter J Liu, and Bin Yu. 2018. Beyond word importance: Contextual decomposition to extract interactions from lstms. In *International Conference on Learning Representations*.

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

Martin J Osborne and Ariel Rubinstein. 1994. *A course in game theory*. MIT press.

Andrew D Selbst and Solon Barocas. 2018. The intuitive appeal of explainable machines. *Fordham L. Rev.*, 87:1085.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE international conference on computer vision*, pages 618–626.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR.

Chandan Singh, W James Murdoch, and Bin Yu. 2018. Hierarchical interpretations for neural network predictions. In *International Conference on Learning Representations*.

Leon Sixt, Maximilian Granz, and Tim Landgraf. 2020. When explanations lie: Why many modified bp attributions fail. In *International Conference on Machine Learning*, pages 9046–9057. PMLR.

Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. 2020. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on empirical methods in natural language processing*, pages 1631–1642.

J Springenberg, Alexey Dosovitskiy, Thomas Brox, and M Riedmiller. 2015. Striving for simplicity: The all convolutional net. In *ICLR (workshop track)*.

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.

Mukund Sundararajan, Kedar Dhamdhere, and Ashish Agarwal. 2020. The shapley taylor interaction index. In *International Conference on Machine Learning*, pages 9259–9268. PMLR.

Mukund Sundararajan and Amir Najmi. 2020. The many shapley values for model explanation. In *International Conference on Machine Learning*, pages 9269–9278. PMLR.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.

Michael Tsang, Sirisha Rambhatla, and Yan Liu. 2020. How does this interaction affect me? interpretable attribution for feature interactions. *Advances in Neural Information Processing Systems*, 33.

Junlin Wang, Jens Tuyls, Eric Wallace, and Sameer Singh. 2020. Gradient-based analysis of nlp models is manipulable. In *Conference on Empirical Methods in Natural Language Processing: Findings*, pages 247–258.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. 2018. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

8 Appendix

8.1 Detailed proof of theorems

Given dividend $d(S)$ is constructed to be non-negative, it is straight forward to show that $v(S)$ satisfies Axioms 1 to 4, given it is a sum of one or more non-negative dividends.

Lemma 1 *$v(S)$ satisfies Sensitivity (a)*

Proof 1 *Let there be a feature a_i such that, $f(x) \neq f(b)$ for given input x and baseline b that only differ in a_i. To prove $v(S)$ satisfies Sensitivity (a) it is sufficient to prove that in the above scenario $\mathrm{IDG}(\{a_i\}) \neq 0$. Then from (Eq 2)*

$$\hat{z}_j^{\{a_i\}} = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{otherwise} \end{cases}$$

Since, in the given case, x_i is the only feature that varies on the straight line path connecting b and x, we can rewrite $f(x) = g(x_i)$. Therefore

$$\nabla_{\{a_i\}} f(x) = \frac{\partial}{\partial x_i} f(x) = \frac{\mathrm{d}}{\mathrm{d}x_i} g(x_i)$$

Thus

$$\begin{aligned} \mathrm{IDG}(\{a_i\}) &= \int_{\alpha=0}^{1} \frac{\partial}{\partial x_i} f\left(b + \alpha(x - b)\right) \mathrm{d}\alpha \\ &= \int_{\alpha=0}^{1} \frac{\mathrm{d}}{\mathrm{d}x_i} g\left(b_i + \alpha(x_i - b_i)\right) \mathrm{d}\alpha \\ &= \frac{1}{x_i - b_i} \int_{x_i=b_i}^{x_i} \frac{\mathrm{d}}{\mathrm{d}x_i} g(x_i) \, \mathrm{d}x_i \\ &= \frac{g(x_i) - g(b_i)}{x_i - b_i} \\ &= \frac{f(x) - f(b)}{x_i - b_i} \\ &\neq 0 \end{aligned}$$

■

Lemma 2 *$v(S)$ satisfies Sensitivity (b)*

Proof 2 *Let there be a feature a_i such that, $f(x) = f(y)$ for every input x and y that only differ in a_i. To prove $v(S)$ satisfies Sensitivity (b) it is sufficient to prove that $\mathrm{IDG}(S) = \mathrm{IDG}(S')$, for all S such that $a_i \in S$, and $S' = S \setminus \{a_i\}$. The precondition of Sensitivity (b) implies that*

$$\frac{\partial}{\partial x_i} f(x) = 0$$

Therefore for any S and S' such that $S' = S \setminus \{a_i\}$

$$\nabla_S f(x) = \nabla_{S'} f(x)$$

Which implies that $\mathrm{IDG}(S) = \mathrm{IDG}(S')$. ■

Lemma 3 *$v(S)$ satisfies Symmetry Preservation*

Proof 3 *To prove that $v(S)$ satisfies Symmetry Preservation, it is sufficient to prove that for any feature subset $S \subseteq A \setminus \{a_i, a_j\}$, $\mathrm{IDG}(S \cup \{a_i\}) = \mathrm{IDG}(S \cup \{a_j\})$. The precondition of functional equivalence implies that if in a given feature vector x, $x_i = x_j$ then*

$$\frac{\partial}{\partial x_i} f(x) = \frac{\partial}{\partial x_j} f(x)$$

Additionally, when considering $x_i = x_j$ and $b_i = b_j$, we have

$$\nabla_{S \cup \{a_i\}} f(x) = \nabla_{S \cup \{a_j\}} f(x)$$

Further, this also implies that $x_i = x_j$ on every point on the straight line connecting b and x. The above imples that $\mathrm{IDG}(S \cup \{a_i\}) = \mathrm{IDG}(S \cup \{a_j\})$. ■

Lemma 4 *$v(S)$ satisfies Implementation-Invariance*

Proof 4 *$v(S)$ satisfies Implementation Invariance since they only depend on gradients of the nerual network function and its evaluations.* ■

8.2 Complexity of Algorithm 1

In Algorithm 1, the for loop on line 2 computes $m+1$ forward and backward backward passes of the neural network. Let the graph structure induced by M contain V vertices and E edges. Then the loop of line 5 requires V computations of $\mathrm{AIDG}(S)$ each of which requires $O(m \cdot |A|)$ computation time. Next, Z can be computed in $O(V)$ time. Each iteration of the loop on line 9 takes $O(1)$ time. Finally the loop on line 12 can be computed in $O(E)$ time.

Thus, the overall time complexity of Algorithm 1 is $O\left(m(F + B + V \cdot |A|) + V + E\right)$, where F and B are the time complexity of a single forward and backward pass of the neural network, V and E are, respectively, the number vertices and edges in the graph structure induced by the family of meaningful feature subsets M, $|A|$ is the number of features, and m the number of approximation steps used to compute $\mathrm{AIDG}(S)$.

8.3 Additional results

IMDB. The dataset (Maas et al., 2011) consists of $25K$ positive labeled and $25K$ negatively labeled reviews posted on IMDB.

For evaluation, we deploy the same procedure as in case of Yelp to obtain 100 representative examples. Two illustrative examples are provided in Figures 3 and 4.

Negative example. We consider an example from the SST dataset where the classifier model made wrong inference. The ground truth class was negative while the inferred class was positive. The value function scores for all the valid coalitions are provided in Figure 5. The results show that although the classifier was able to distinguish between the positive sense manifested in the first part and the negative sense in the second, it made a positive inference overall. This might be due to the low confidence of the classifier in inferring the final class as demonstrated by the probabilities - 0.44 for negative and 0.56 for positive class. However, further investigations are required before stronger claims can be made.

8.4 Training models

SST. The XLnet-base model was trained with batch size 24 for 4 epochs. We use AdamW (Loshchilov and Hutter, 2018) as optimizer with learning rate $2e^{-05}$ and weight decay 0.01. The model achieved an accuracy of 0.915 on the test set. The XLnet-large model was trained with same batch size, for same number of epochs and with same optimizer. The learning rate and weight decay were $5e^{-06}$ and 0.01 respectively. An accuracy of 0.916 was obtained on the test set for this model. BERT-itpt was trained with a batch size of 24 and optmized with AdamW with learning rate $1e^{-5}$ and weight decay 0.01. The embedding layers were not frozen during training.

Yelp. The Bert-itpt model was trained with training batch size of 24, for 3 epochs and with AdamW (learning rate $1e^{-05}$, weight decay 0.01) and achieved an accuracy of 0.947 on the test set. We further trained an XLnet models with similar training hyperparameters and achieved an accuracy of 0.983.

IMDB. The two models Bert-itpt and XLnet-large were both trained on $25K$ training examples and tested on the rest. The batch sizes were 24 and 32 respectively. AdamW was used as optimizer for both models with same weight decay of 0.01 but learning rates $2e^{-05}$ and $2e^{-05}$ respectively for Bert-itpt and XLnet-large. We could obtain testing accuracy of 0.957 and 0.967 respectively for the two models.

All these models were trained on cluster with 2 CPUs each with 20 cores, 384 GB DDR4 RAM and Inter Xeon Gold 6148 processor. The distributed set up was connected through Mellanox ConnectX-5 network and used Lustre file system. The set up also utilized 4 NVIDIA Tesla V100 GPUs each with 32 GB memory.

Experiments with IDG were performed on a system with Intel Core i7-8550U 1.80GHz CPU with 16 GB RAM.

8.5 Adversarial attacks against explanations

In (Selbst and Barocas, 2018) the authors argue that one of the main reasons to develop explanation techniques is to enable humans to understand how automated decision systems work which in turn enable us to debate on whether the model's rules for decision making are justifiable. On the flip side security researchers (Slack et al., 2020) have have shown that such efforts can be stifled using adversarial attack techniques. In particular (Slack et al., 2020) showed that models can be trained to deceive blackbox explanation methods, such that it provides 'unfair' results on in-distribution samples while exhibiting different behavior when explained using KernelSHAP. In a recent study (Wang et al., 2020) the researchers have explored creation of deceptive models that can fool gradient based methods such as IntGrad (Sundararajan et al., 2017). In (Slack et al., 2020) the authors showed that evaluating models on out-of-distribution inputs, that is the inputs that the original model was not tested on, is a large potential attack surface for such deceptive techniques. While unlike existing studies, IDG doesn't evaluate out-of-distribution values, it seems certainly possible to use adversarial training methods to deceive IDG. While for the current work evaluation against adversarial attack was out of scope, we consider it as an important future direction.

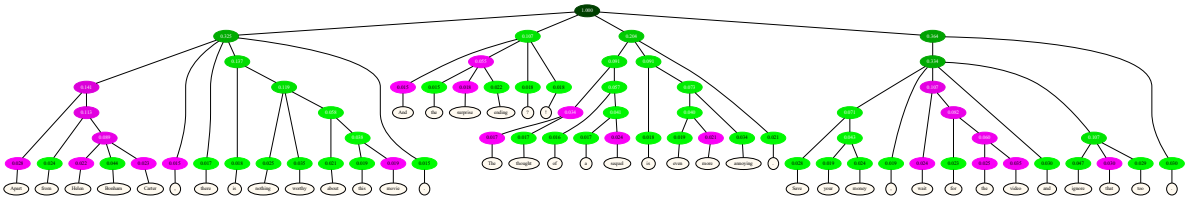


Figure 3: The value function scores assigned by our framework for different coalitions (interactions) between phrases for the review `Apart from Helena Bonham Carter, there is nothing worthy about this movie. And the surprise ending?! The thought of a sequel is even more annoying. Save your money, wait for the video and ignore that too.` The inferred class is negative. IDG correctly captures the positive sense (even though the overall sense is negative) of the phrase `Apart from Helena Bonham Carter` as it contributes oppositely to the overall inference result.

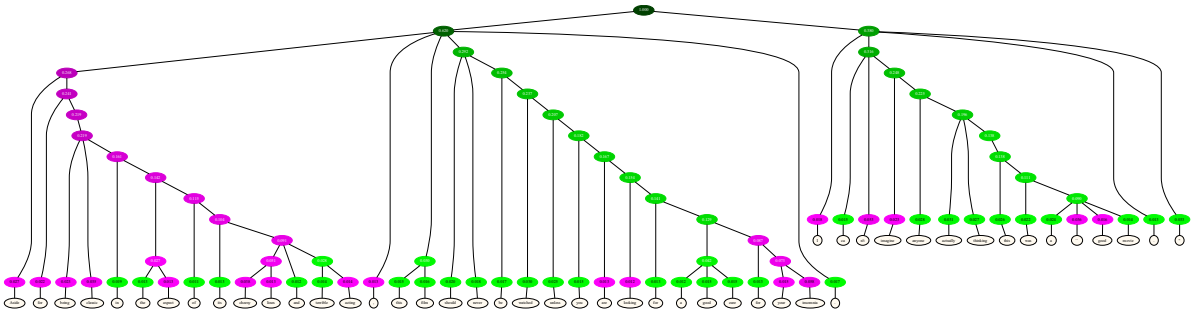


Figure 4: The value function scores assigned by our framework for different coalitions (interactions) between phrases for the review `Aside for being a classic in the aspect of its cheesy lines and terrible acting, this film should never be watched unless you are looking for a good cure for your insomnia. I can't imagine anyone actually thinking this was a ''good movie''.` The inferred class is negative. IDG shows how the classifier captures the positive sense (even though the overall sense is negative) of the phrase `Aside for being a classic in the aspect of cheesy lines and terrible acting` as it contributes oppositely to the overall inference result.

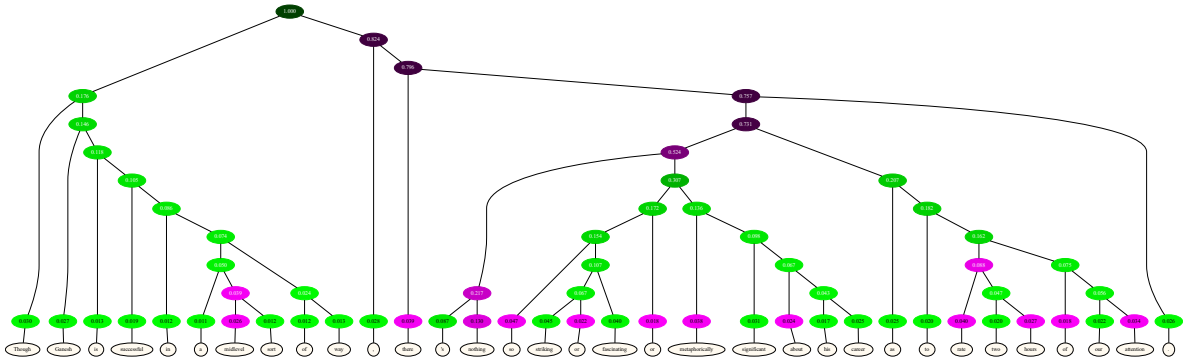


Figure 5: The value function scores assigned by our framework for different coalitions (interactions) between phrases for the review `Though Ganesh is successful in a midlevel sort of way, there's nothing so striking or fascinating or metaphorically significant about his career as to rate two hours of our attention.` The inferred class is positive while the ground truth class is negative.