

EXSUM: From Local Explanations to Model Understanding

Yilun Zhou
MIT CSAIL

Marco Tulio Ribeiro
Microsoft Research

Julie Shah
MIT CSAIL

{yilun, julie_a_shah}@csail.mit.edu marcotcr@microsoft.com

<https://yilunzhou.github.io/exsum/>

Abstract

Interpretability methods are developed to understand the working mechanisms of black-box models, which is crucial to their responsible deployment. Fulfilling this goal requires both that the explanations generated by these methods are correct *and* that people can easily and reliably understand them. While the former has been addressed in prior work, the latter is often overlooked, resulting in informal model understanding derived from a handful of local explanations. In this paper, we introduce explanation summary (EXSUM), a mathematical framework for quantifying model understanding, and propose metrics for its quality assessment. On two domains, EXSUM highlights various limitations in the current practice, helps develop accurate model understanding, and reveals easily overlooked properties of the model. We also connect understandability to other properties of explanations such as human alignment, robustness, and counterfactual minimality and plausibility.

1 Introduction

Understanding a model’s behavior is often a prerequisite for deploying it in the real world, especially in high-stake scenarios such as financial, legal, and medical domains. Unfortunately, most high-performing models, such as neural networks, are black-boxes. Thus, model-agnostic interpretability techniques have been developed, with the majority being “local” – algorithms that produce an explanation for a specific input at a time (e.g., Li et al., 2016; Ribeiro et al., 2016).

Even with these local explanations, there are still two hurdles to overcome before achieving the ultimate goal of complete understanding of a model. First, some local explanations may not correctly (or faithfully) represent the model’s reasoning process (Jacovi and Goldberg, 2020), as has been demonstrated both theoretically (Nie et al., 2018) and empirically (Adebayo et al., 2018) in prior work. As a

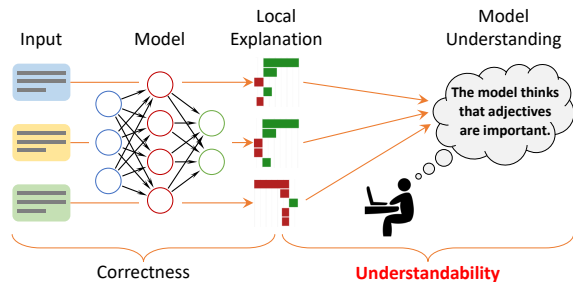


Figure 1: Local model explanations need to be both correct and easily understandable. While much prior work (e.g., Zhou et al., 2022) has studied the former property, this paper focuses on the latter, which has thus far been largely ignored.

result, correctness evaluation has received much attention in the community (e.g., Samek et al., 2016; Arras et al., 2019; Zhou et al., 2022).

Another mostly overlooked property of explanations is their *understandability*. As the model understanding pipeline depicted in Fig. 1 shows, explanations need to be both correct and easily understandable, since even correct explanations are not as valuable if they lead to incorrect understanding. However, the concept of understandability has yet to be formalized, and instead users often derive model understanding from few examples in a non-rigorous (and potentially incorrect) manner.

Consider the sentiment classification task shown in Fig. 2. On a test input, the model makes the correct prediction of positive sentiment. Obviously, this evidence is insufficient to conclude that “*in general*, the model classifies positive inputs correctly”, because even a random-guess model is correct 50% of the time on a single instance. Instead, statistics such as the confusion matrix serve to rigorously support (or refute) generalization claims about model *performance* – for example, “the model is correct 97.6% of the time on positive inputs” – ensuring an accurate understanding of model performance.

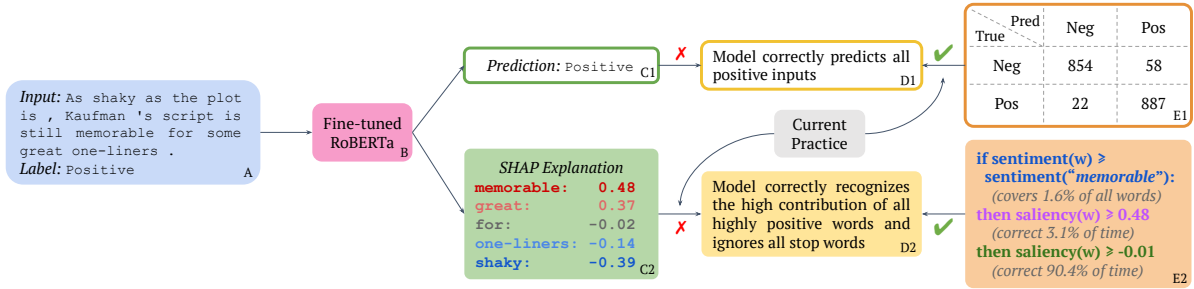


Figure 2: An analogy between understanding model prediction (top route) and model explanation (bottom route). A test input (A) is fed into a fine-tuned RoBERTa model (B), which generates a correct prediction (C1) and reasonable explanation (C2). While generalized claims of understanding model *performance* (D1) are made rigorously from quantitative statistics such as the test set confusion matrix (E1), claims of understanding model *behavior* (D2) are predominantly derived informally from one or few explanations (C2). In this paper, we argue the necessity of formalizing this process, and propose the explanation summary (EXSUM) framework (E2), which reveals the severe limitations of the *ad hoc* model understanding (D2).

Do we understand model *behaviors* in the same rigorous way? Fig. 2 shows that the SHAP score (Lundberg and Lee, 2017) of the word “*memorable*” is highest at 0.48, while that of “*for*” is negligible at -0.02. Therefore, it is tempting to conclude that “*in general*, the model recognizes the high positive contribution of highly positive words and ignores stop words” – as expected for an accurate sentiment classifier. However, this is a generalization from a single instance, and thus potentially unreliable. We need the “confusion matrix” analogue for such claims, which to the best of our knowledge does not exist, making it hard to derive model understanding from local explanations.

In this paper, we propose EXSUM, a mathematical framework to formalize model understanding. In EXSUM, each piece of “model understanding” is specified precisely via a rule that links inputs to attribution values. For example, the tentative understanding described in the previous paragraph could be formalized as “words more positive than *memorable* (as measured by the word sentiment score given in the dataset, e.g., *flawless*, *charming*, etc) have SHAP attribution value in the [0.48, 1] range.” This precise definition allows for quantitative evaluations. For example, this rule covers 1.6% of all words in the corpus, and is only correct 3.1% of the time. For the rule to be 90% correct, we need a wide and uninformative range of [-0.01, 1], indicating that a hasty generalization from “*memorable*” is unwarranted. Similarly, a saliency range of [-0.05, 0.05] for stop words is only correct 64% of the time: over 1/3 of stop words have *non-negligible* saliency – an understanding that is easily available with EXSUM, but might be missed with informal

explanation inspection. We define metrics to establish the quality profile of each rule and present a tool that makes it easy for users to construct EXSUM rules from local explanations. Finally, we demonstrate how EXSUM reveals the various drawbacks in the current practices of *ad hoc* model understanding, and allows for better understanding of model behavior in two separate tasks.

2 On Generalized Model Understanding

Besides the practical example above, we start from first principles and argue that *generalized* model understanding is the central concept for explanation usefulness. Local explanations are *mathematical descriptions (MD) of some aspect of model behavior, for specific inputs*. For example, gradient saliency (in the embedding space) is the sensitivity of the prediction to infinitesimal changes in the token embedding; occlusion saliency is the prediction change if individual embeddings are zeroed out. It is with these mathematical descriptions that people associate *high-level interpretations (HL) of model behavior*, such as associating the above two metrics with word importance. This (unconscious) train of thought can be described as follows:

$$x \rightarrow \text{MD} \rightarrow \text{HL}.$$

Crucially, people rarely study MD or HL for one *specific* input, as explanations are often used to understand broader model behaviors, such as reliance upon spurious correlation, non-discrimination of a protected class, or usage of unknown scientific principles. We elaborate upon these use cases in App. A to demonstrate that people implicitly or explicitly seek generalized model understanding.

From another perspective, analogous to why people ultimately focus on the *generalization* accuracy of a model, they (should) focus on *generalized* model understanding derived from local explanations.

For example, after observing that *some* highly polar words have high contribution for a sentiment classification model, people conclude that *all* highly polar words have high contribution. This process can be formalized as follows:

$$\left. \begin{array}{l} x_1 \rightarrow \text{MD}_1 \rightarrow \text{HL}_1 \\ \dots \\ x_n \rightarrow \text{MD}_n \rightarrow \text{HL}_n \end{array} \right\} \rightarrow \text{HL}^{(g)},$$

where $\text{HL}^{(g)}$ is the *generalized* high-level model understanding. This generalization is too informal, not least because the step from MD_i to HL_i is itself already informal. Alternatively, we propose to generalize at the MD level, as follows:

$$\left. \begin{array}{l} x_1 \rightarrow \text{MD}_1 \\ \dots \\ x_n \rightarrow \text{MD}_n \end{array} \right\} \rightarrow \text{MD}^{(g)} \rightarrow \text{HL}^{(g)}.$$

Since MDs are rigorously defined mathematical quantities (e.g., the prediction of the sentence drops by 32% after the embedding of “great” is zeroed out), we can define and evaluate the quality of their generalization, and $\text{HL}^{(g)}$ can also include any failures and anomalies. As each MD is a local explanation, we call $\text{MD}^{(g)}$ the *explanation summary* (EXSUM), and proceed by instantiating this principle for feature attribution explanations.

3 The EXSUM Framework

3.1 Setup and Notation

We focus on the classification setting, but all the ideas below can extend straightforwardly to regression. We have an input space \mathcal{X} and output space $\mathcal{Y} = \{1, \dots, K\}$ of K classes. A data point is an input-output pair $d = (x, y) \in \mathcal{D} = \mathcal{X} \times \mathcal{Y}$, distributed as \mathbb{P}_D . We consider a model $m : \mathcal{X} \rightarrow \Delta^{K-1}$ where $m(x)$ is the predicted class distribution on the probability simplex.

Feature attribution explainers assign an attribution, also known as saliency or importance, to each input feature, such as a token in a text input. For an instance (x, y) , each feature of x is called a fundamental explanation unit (FEU), defined as $u = (x, y, l) \in \mathcal{U}$ with $1 \leq l \leq L_x$ as the feature index. $e(u) \in \mathcal{E}$ represents the attribution value assigned to it, where \mathcal{E} is the attribution space, such as $[-1, 1]$ for normalized explanations.

$e(u_-) = (e_x^{(1)}, \dots, e_x^{(l-1)}, e_x^{(l+1)}, \dots, e_x^{(L_x)}) \in \mathcal{E}_-$ denotes the explanations on all other FEUs of x .

We *define* a distribution \mathbb{P}_U over \mathcal{U} such that the probability (or probability density) of $u = (x, y, l)$ is $1/L_x$ of that of d under the data distribution \mathbb{P}_D . In other words, sampling of u can be performed in two steps: first draw an instance $d = (x, y) \sim \mathbb{P}_D$, then a feature index $l \sim \text{Unif}(\{1, \dots, L_x\})$.

3.2 EXSUM Rules

An EXSUM rule formalizes a piece of model understanding, such as that for positive words in Fig. 2, which we use as the running example.

Definition 3.1 (EXSUM rule). An EXSUM rule r is defined by two functions. A binary-valued *applicability function* $a : \mathcal{U} \rightarrow \{0, 1\}$ determines whether the rule applies to a given FEU, with 1 being applicable and 0 otherwise. We use $a(\mathcal{U}) = \{u \in \mathcal{U} : a(u) = 1\}$ to denote the *applicability set*. A set-valued *behavior function* is defined as $b : a(\mathcal{U}) \times \mathcal{E}_- \rightarrow \mathcal{P}(\mathcal{E})$ where $\mathcal{P}(\mathcal{E})$ is the power set (i.e., the set of all subsets) of \mathcal{E} . This function predicts a set of possible explanation values for the FEU, called the *behavior range*. The rule is written as $r = \langle a, b \rangle$. We abbreviate $b(u, e(u_-))$ as $b(u)$ and refer to the two functions as a - and b -functions.

For FEU $u = (x, y, l)$, the a -function typically depends only on x_l , but could depend on the entire input x (e.g., for long sentences) or the output y (e.g., for positive class). In our example, it tests whether the sentiment score is greater than that of the word “memorable” (0.638). The b -function usually outputs a constant range. Since “memorable” has a saliency of 0.479, the range is $[0.479, 1.0]$.

3.3 Additional Examples

While we expect most rules to use rather simple a - and b -functions, they can also be more complex with more nuanced aspects. For the following examples, recall that $u = (x, y, l)$. An applicability function can target words only in long sentences using a conjunction with $\text{len}(x) \geq L$, where L is the threshold. We can also target inputs with ambivalent predictions with $\max_c m(x)_c \leq 0.6$, where $\max_c m(x)_c$ is the probability of the predicted class. For behavior functions, to indicate the first word of the sentence has higher saliency than the rest, we can define $b(u, e_-) = (\max_{l' \geq 2} e_-^{(l')}, 1.0]$, where the a -function selects the first word (i.e. $a(u) = \mathbb{1}_{l=1}$). Similarly, to describe that an FEU has higher saliency than all

the verbs in a sentence, we can use $b(u, e_-) = (\max_{v:\text{is_verb}(x_v)} \{e_-^{(v)}\}, +\infty)$.

3.4 EXSUM Rule Unions

Since a single EXSUM rule is designed to capture one aspect of model understanding, multiple rules are often necessary for comprehensive understanding. However, conflicts can occur when multiple rules apply to the same FEU but the b -functions are different. We resolve them by defining the composition of two or more rules into a *rule union*.

Definition 3.2 (Precedence-Mode Composition). Two rules, $r = \langle a, b \rangle$ and $r' = \langle a', b' \rangle$, can be composed into a precedence-mode rule union $r^* = r > r'$ defined as $r^* = \langle a^*, b^* \rangle$ where

$$a^*(u) = \mathbb{1}\{a(u) + a'(u) \geq 1\}, \quad (1)$$

$$b^*(u) = \begin{cases} b(u) & \text{if } a(u) = 1, \\ b'(u) & \text{if } a(u) = 0, a'(u) = 1, \end{cases} \quad (2)$$

represent the a - and b -functions of rule union r^* , with semantics similar to those for rules.

For example, if we want to split positive adjectives into a separate rule from other positive words, we create a rule to test for part-of-speech and sentiment score, and assign a higher precedence to this rule, such that the original rule is only applicable to the remaining non-adjectives. One useful practice is to include a lowest-precedence catch-all rule that covers everything not addressed by other rules, with a constant $a(u) = 1$ function, which leaves no FEUs unaccounted for.

Definition 3.3 (Intersection-Mode Composition). Two rules, $r = \langle a, b \rangle$ and $r' = \langle a', b' \rangle$, can be composed into an intersection-mode rule union $r^* = r \& r'$ defined as $r^* = \langle a^*, b^* \rangle$ where

$$a^*(u) = \mathbb{1}\{a(u) + a'(u) \geq 1\}; \quad (3)$$

$$b^*(u) = \begin{cases} b(u) & \text{if } a(u) = 1, a'(u) = 0, \\ b'(u) & \text{if } a(u) = 0, a'(u) = 1, \\ b(u) \cap b'(u) & \text{if } a(u) = a'(u) = 1. \end{cases} \quad (4)$$

Unlike precedence-mode, intersection-mode composition is symmetric with respect to the two rules. This mode is helpful when each property of an FEU has a corresponding behavior range, and the final behavior range of an FEU depends on FEU’s properties. For example, if verbs have a behavior range of $[-0.4, 0.4]$ and strongly positive words have a behavior range of $[0.3, 1]$, a strongly positive verb would have a behavior range

$[0.3, 0.4]$, or the intersection of the two constituent ranges. In our case studies, however, we do not encounter any situations in which intersection-mode compositions were preferable.

Since rule unions are also defined by a - and b -functions, they can form other rule unions in the same way. Recursively, this results in a list of rules composed into a single rule union, written as $r^* = (r_3 > r_1) \& ((r_4 \& r_2) > r_5)$. This rule union represents our *generalized model understanding*.

3.5 Quality Metrics

We propose three metrics for establishing the quality profiles of EXSUM rules or rule unions.

Definition 3.4 (Coverage). The coverage of a rule (union) $r = \langle a, b \rangle$ is defined as follows:

$$\kappa(r) = \mathbb{E}_{U \sim \mathbb{P}(U)} [a(U)]. \quad (5)$$

This represents the fraction of FEUs that we attempt to understand. While individual rules may have low coverage because they specialize in aspects of the model behavior, we want their union to have high coverage to achieve a comprehensive understanding of the model and prevent model prediction from being excessively affected by the uncovered (i.e. unexplained) input features. For our positive word rule, the coverage is the frequency of those words in the corpus and not surprisingly is only 1.6%. By contrast, including a catch-all rule in the union maxes out its coverage value at 100%.

Definition 3.5 (Validity). Let $\mathbb{P}_{a(U)}$ be \mathbb{P}_U truncated to the set of applicable FEUs. The validity of a rule (union) $r = \langle a, b \rangle$ is then defined as follows, capturing the intuitive notion of a “correct” understanding:

$$\nu(r) = \mathbb{E}_{U \sim \mathbb{P}_{a(U)}} [\mathbb{1}\{e(U) \in b(U)\}]. \quad (6)$$

For our example, we compute it as the frequency that the saliency of those words is actually in the range of $[0.479, 1]$ – which turns out to be only 3.1% of the time. However, validity alone is not sufficient, as it increases with wider behavior range. We thus establish sharpness as a competing metric.

Definition 3.6 (Sharpness). Let \mathbb{P}_E be the probability measure corresponding to the marginal distribution over explanation values generated by the explainer on $u \sim \mathbb{P}_U$. The sharpness of a rule (union) $r = \langle a, b \rangle$ is defined as follows:

$$\sigma(r) = \mathbb{E}_{U \sim \mathbb{P}_{a(U)}} [1 - \mathbb{P}_E(b(U) \setminus U)], \quad (7)$$

where $b(U) \setminus U = b(U) \setminus \{U\}$ removes the actual attribution value U from the behavior range to pre-

vent penalizing sharpness simply because the attribution value is very common (e.g., zero for sparse explanations), in which case \mathbb{P}_E is discrete at U .

Sharpness represents precision in the understanding, as $1 - \sigma(r)$ gives the probability that a random FEU explanation value is correct. Thus, a lack of precision represented by a wide behavior range has minimal sharpness. We use the probability measure \mathbb{P}_E to define the “size”, as it is consistent across all explanation distributions, most of which are non-uniform. A more general interpretation of sharpness is the consistency of the described model behavior: if a behavior range is wide (e.g., containing very positive *and* negative saliencies), then it is less sharp, and hence less useful. \mathbb{P}_E could be replaced by an application-specific diversity measure, though the precision notion may be lost.

There is generally a trade-off between validity and sharpness, as more precise rules (i.e., those with narrower behavior ranges) are less likely to be valid. For our rule, the probability of a *random* word saliency being in $[0.479, 1.0]$ is 0.2%, indicating that explanation values are rarely higher than 0.479. This makes sharpness very high at 99.8%. However, the rule is not useful because of its low validity; i.e., it is almost never correct. By comparison, the looser range of $[-0.01, 1.0]$ has 90.4% validity but 28.6% sharpness. There is another trade-off between coverage and the two, since a larger set of covered FEUs tends to be more diverse, making it harder to write a *b*-function that remains as valid and sharp simultaneously.

Since these metrics are all expected values, we can estimate them by their empirical estimate from a dataset (i.e., a simple average), and \mathbb{P}_E can be constructed by kernel density estimation.

4 EXSUM Development Process and GUI

We describe a systematic procedure for authoring EXSUM rule unions from scratch and utilize it in Sec. 5. Starting from an empty rule union with

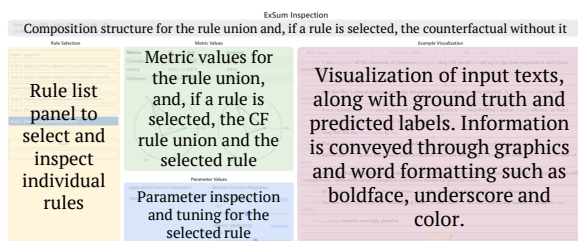


Figure 3: EXSUM inspection GUI.

no FEUs covered, we iteratively create rules that target uncovered FEUs. Each rule describes one model behavior, such as that for highly positive words. For a rule, the *a*- and *b*-functions need to be defined, which may involve setting and tuning parameters, such as the sentiment threshold. Last, we add a lowest precedence catch-all rule if any FEUs remain uncovered. During this process, we may also merge or split rules and change the composition structure according to the metric values.

To support these steps, we developed a Python Flask-based (Grinberg, 2018) graphical user interface (GUI, Fig. 3). Users can visualize the FEUs, with font formatting for their coverage and validity. Users can also filter for uncovered or invalid FEUs, iteratively constructing and refining the rule union. EXSUM rule definitions usually include parameters such as the sentiment threshold. Manually selecting correct values for the parameters is tedious, so the lower middle panel of the GUI implements automatic parameter tuning for a given target metric value. Installation and usage instructions for the GUI are available on the project page¹.

5 Evaluation

We construct EXSUM rule unions for SST and QQP models (details in App. B). We split the test set into a *construction set* to create the rule union and tune its parameters (analogous to the training and validation set in supervised model training) and an *evaluation set* to compute unbiased estimates of the metric values (analogous to the test set).

5.1 Sentiment Classification

Setup We use SHAP explanations (Lundberg and Lee, 2017) for fine-tuned RoBERTa (Liu et al., 2019), and take 300 random sentences as the construction set, with the remaining 1910 sentences as the evaluation set. We compute five features for each FEU: sentiment score, part of speech (POS), named entity recognition (NER), dependency tag (DEP) and word frequency. For example, the word “same” in the sentence “*They felt like the same movie to me .*” has sentiment score of 0.028, POS = ADJ, NER = O, DEP = amod, and frequency of $7.14e-4$, with SHAP saliency of -0.82.

Current Practice We evaluate the current practice of extracting *informal* model understanding from local explanation inspection against the three

¹<https://yilunzhou.github.io/exsum/>

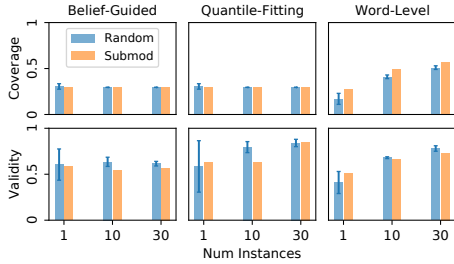


Figure 4: Coverage and validity metrics for the three current practice modes. Tab. 4 of App. B.1.1 presents the complete numerical data (also with sharpness).

metrics. We assess three values of K , the number of inspected instances: 1, for the typical *ad hoc* setting of generalization from a single explanation, 10, for a more careful investigation, and 30, which is quite cumbersome for manual inspection. These examples are selected either randomly or by submodular pick (Ribeiro et al., 2016). Next, we consider three ways to extract model understanding – belief-guided (BG), quantile-fitting (QF) and word-level (WL) – and apply them to create rules on strongly positive words and stop words introduced in Sec. 1. For the strongly positive word rule, BG mandates that words more positive than the average sentiment score should have an above-average saliency score, representing the belief of a positive correlation between the two. For the stop word rule, a saliency range belief of $[-0.05, 0.05]$ is averaged with the observed range. For both rules, QF extracts the 5%-95% quantile interval of the saliencies for words covered by the respective rule. WL, by contrast, creates a behavior range *for each word seen*, with 0.03 margin on both sides. App. B.1.1 presents technical details for these.

We formalize the understanding derived from the selected instances and plot their coverage and validity metrics on the evaluation set in Fig. 4. For BG and QF, the bars represent the average metric value of the positive word and stop word rules. For WL, the bars represent the metric for the rule union consisting of an individual rule for each unique word. Error bars for the random pick represent the standard deviation across five iterations. Tab. 4 of App. B.1.1 presents the complete statistics for all metric values, and we highlight several findings.

- A very small number of samples (e.g., 1) exhibit large variance for random pick, and low validity for both pick methods. This confirms the intuition that model understanding from very few explanations should be avoided.

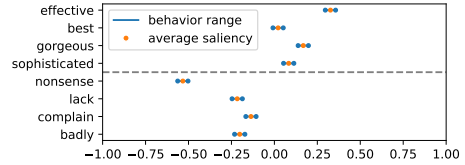


Figure 5: Behavior ranges can vary widely and unpredictably on similar words for WL rules.

- BG overall yields low validity, because its “beliefs” turn out to be quite incorrect. This suggests a strong prior belief about how the model works could lead to incorrect conclusions.
- While submodular pick can select a more diverse set of words, to the particular benefit of the coverage of WL², its validity is generally lower due to under-representation of common words.
- Although WL achieves highest coverage *and* validity, it has > 500 rules at $K=30$, with similar words having very different ranges, as shown in Fig. 5 – a conglomerate (almost) impossible to make sense of. It also overfits, as the evaluation set validity is much lower than the construction set validity (which is 100% by construction).
- At $K=10$, only the stop word rule with random pick QF achieves validity $> 80\%$, indicating that even the more careful practices are unreliable.

All the drawbacks call for a principled way to derive robust model understanding with enforceable metric values (e.g. validity). As we demonstrate next, given a large construction set and automatic parameter tuning assistance, we can create such a EXSUM rule union. Finally, as a meta-point, the above discussion above of various limitations would not be possible without the proposed EXSUM formalization and metric definitions.

EXSUM Construction We create a rule union consisting of nine rules, with target validity of 90% and tune the sharpness accordingly. Tab. 1 summarizes the individual and aggregate metrics.

Clearly, high validity comes at the cost of low sharpness. Since $(1 - \text{sharpness})$ is the probability that a random FEU has an explanation value within the behavior range, this around 90.7% validity should be put into a context where the random baseline achieves a validity of around 75%. In this sense, we attain only a crude understanding of the local explanations that misses many subtleties.

Nonetheless, Rule 3 (strongly positive words)

²The other two are less affected because the subject of the rule (e.g., stop words) largely dictates which words it covers.

Idx	Rule	Cov%	Val%	Shp%
1	Negation	1.2	89.5	65.1
2	Strongly neg. adj	3.2	91.6	83.5
3	Strongly pos. words	5.1	91.9	40.0
4	Strongly neg. non-adj	1.2	89.9	71.4
5	Person name	2.4	90.9	28.4
6	Stop words	47.5	90.8	23.5
7	Zero-sentiment words	17.1	90.0	15.6
8	Weakly pos. words	15.4	91.2	11.3
9	Weakly neg. words	5.7	91.7	31.4
Un-	On construction set	100	90.7	26.1
ion	On evaluation set	100	89.4	26.2

Table 1: Metrics for SST rules and rule union.

and Rule 6 (stop words) achieve better validity-sharpness trade-off than their counterparts created using the *ad hoc* BG and QF methods above. Moreover, the WL rules cover all words seen in the analyzed instances – analogous, in a sense, to our EXSUM rule union. While the validity-sharpness trade-off is comparable between the two, ours has 100% coverage due to the effectively “catch-all” Rule 7, while WL rules have less than 60%. Most importantly, as our rule union is composed of nine semantically organized rules, it is much more interpretable than WL, which include more than 500 unpredictably varying rules (Fig. 5).

The fact that the EXSUM rule union reveals the imprecision and limitations of our model understanding while still performing better than current practice emphasizes the need for more formal and quantitative model understanding, as well as the development of methods that are easier to *understand*, in addition to being correct. Below, we highlight two sets of rules that quantitatively support or refute our intuition, and cover the rest in App. B.1.2. **Rule 2, 3, 4, 8, 9: Sentiment-carrying words.** We expect a sentiment classifier to recognize sentiment-laden words. To test our intuition, we create rules for positive and negative words, and further split each set of words into two according to sentiment strength, resulting in four rules. For the two rules on strong words, we find that wide behavior ranges of $[0.01, 1]$ and $[-1, -0.01]$ are necessary to achieve 90% validity, suggesting the looseness of the model understanding. However, we do observe that negative adjectives (but not positive ones) are modeled much better, where a range of $[-1, -0.06]$ is sufficient for the same validity. Thus, we create a separate Rule 2, with very high sharpness of 84.2%. For the two rules on words of weaker sentiment,

Idx	Rule	Cov%	Val%	Shp%
1	Matching words neg. pred	11.7	90.9	39.5
2	Matching words pos. pred	12.4	90.3	38.6
3	Non-matching words neg. pred	18.7	90.0	35.5
4	Question mark neg. pred	5.2	90.2	36.5
5	Question mark pos. pred	3.8	90.0	23.1
6	Stop words neg. pred	22.3	90.0	32.8
7	Stop words pos. pred	12.6	90.5	12.5
8	Negation words neg. pred	0.3	90.0	36.0
9	Negation words pos. pred	0.1	95.7	7.2
10	All else neg. pred	4.0	92.1	23.5
11	All else pos. pred	8.8	90.3	5.7
Un-	On construction set	100	90.3	29.3
ion	On evaluation set	100	90.0	29.1
Word	On construction set	100	90.8	29.4
Avg	On evaluation set	82.3	84.4	29.4

Table 2: Metrics for QQP rules and rule union. The last two rows are for the baseline at the end of Sec. 5.2.

even wider ranges of $[-0.11, 1]$ and $[-1, 0.05]$ are necessary. Since both ranges encroach upon the other side, the model often considers these words to have an impact opposite to their intrinsic meaning, but we fail to extract further understanding. In addition, negative rules are much sharper than positive ones, suggesting that the model considers a negative word to be stronger evidence for a negative prediction than its positive counterpart.

Rule 6: Stop words. While stop words (e.g., “the”, “of”) *should* have negligible impact on prediction (and saliency values close to zero), a narrow behavior range of $[-0.05, 0.05]$ only has 64% validity. We create this rule for all stop words with 90% target validity and use different ranges on different words for better sharpness. On average, we get $[-0.07, 0.12]$, demonstrating that they can sometimes be more influential than even strong sentiment words. The ranges also tilt to the positive side, uncovering a grammaticality bias wherein prediction is more negative for grammatically incorrect sentences with stop words masked out by SHAP.

5.2 Paraphrase Detection

Setup We use LIME explanations (Ribeiro et al., 2016) for a fine-tuned BERT model (Devlin et al., 2019), with 500 random test sentences as the construction set and the remaining $\approx 40k$ as the evaluation set. We remove the word sentiment feature but add the question ID (1 or 2) of each FEU.

EXSUM Construction QQP is a more complex domain than SST, since the label is the semantic equivalence of *two* sentences. The metric values for the EXSUM are summarized in Tab. 2. Below,

we describe how expectations for the model are validated, but a hidden – and somewhat surprising – phenomenon is also uncovered. All other rules are documented in App. B.2.1.

Rule 1, 2: Matching words. Due to the nature of the task, we expect the model to rely heavily on matching words. For such a word u , defined as (proper) noun, verb, adjective or pronoun that has exactly one case-insensitive match v in the other question, we expect similar saliency to their match due to symmetry, or formally its saliency $s_u \in [s_v - \alpha, s_v + \beta]$, where α and β are lower and upper margins. This behavior function is non-constant, with output depending on the saliency values of other words in the sentence.

For the same margin, FEUs for pairs of negative predictions have much higher validity than positive ones, so we split the rule into two based on the prediction. Despite a less than 1% difference in sharpness (Tab. 2), we have $\alpha = \beta = 0.07$ for the negative rule, but 0.18 for the positive rule, suggesting that the matching words make a much larger and more unpredictable contribution to positive predictions. Interestingly, all other rules had wider intervals for positive predictions as well.

Rule 3: Non-matching words. Next we study model behaviors for non-matching words, defined analogously to matching ones. Following the previous split based on predicted label, we designed two rules. The negative rule has a reasonably sharp behavior range of $[-0.35, 0.01]$ at 90% validity. Given that LIME saliency is the linear regression coefficient on a neighborhood created by word erasure, we conclude that the *presence* of these non-matching words mostly causes the prediction to tilt toward the non-paraphrase (i.e. negative) class, indeed a very reasonable behavior. However, we cannot find a range with 10% sharpness at 90% validity for the positive rule and thus discard it.

With regard to the sharpness contrast by predicted label, one explanation is that the model defaults to a negative prediction, since many negative pairs consist of completely unrelated questions and the model decision is largely insensitive to input perturbations, leading to stable LIME coefficients. On the other hand, a positive prediction requires the cooperation of all parts of both questions. Depending on the exact sentence structure, the importance of each word to the match are different and hard to predict, which prevents the rules from being sharp.

Word Average Baseline Here, we introduce a new baseline as an “automated” version of WL rules in SST. Specifically, for each word in the construction set, we compute a behavior range around its average saliency, with sharpness of 29.4% (matching that of our EXSUM rule union). As Tab. 2 shows, the resulting rule union is much worse than our manual one on both evaluation set coverage and validity, which is not surprising as the word saliency *should* be more context-dependent, due to the matching mechanism of paraphrase detection. Moreover, with more than 2,000 constituent rules, the rule union barely qualifies as any sort of *generalized* model understanding.

6 Related Work

As discussed in Sec. 1, explanation evaluation usually has a focus on correctness (or faithfulness) – i.e., whether the explanation truly reflects the model’s reasoning process. This includes sanity checks (Adebayo et al., 2018), proxy metrics (Samek et al., 2016; Arras et al., 2019), and explicit ground truth (Zhou et al., 2022). The understandability issue has been much less studied, with the exception by Zheng et al. (2021), who proposed an evaluation specifically for rationale models (Lei et al., 2016). EXSUM, however, addresses *post hoc* explanations of general black-box models.

In addition, a few prior works have attempted to capture the “end-to-end” utility of explanations: whether access to explanations leads to performance increase in certain tasks. Hase and Bansal (2020) propose a model-teaching-human setup, subsequently extended by Pruthi et al. (2022) into an automated evaluation procedure. Bansal et al. (2021) study whether explanations can improve human-machine teaming performance. While these studies report mostly negative results, pinpointing the root cause is difficult due to their end-to-end nature. Poor *understanding* of the explanations may be a major reason, as indicated by EXSUM.

Last, some authors have proposed methods for understanding model predictions beyond individual instances. For example, the anchor method (Ribeiro et al., 2018) generates an explicit domain of applicability for each explanation, while Lakkaraju et al. (2016) and Lakkaraju et al. (2019) proposed to learn “patches” of the input space specified by logical predicates. EXSUM also emphasizes the need to understand models that generalizes across instances, and uses logical predicates in

the formulation, but focuses on model understanding via *explanations* instead of direct *predictions*, which can capture a wider variety of *behaviors* (e.g. the matching and non-matching behaviors of the QQP model). Furthermore, the fine-grained analysis of behaviors allows us to investigate whether models are “correct for the correct reason.”

7 The Many Faces of Understandability

The central thesis of this paper is quite simple and intuitive: in order to understand a model from local explanations, we need to understand those local explanations. While EXSUM is the first framework to explicitly formalize and quantify the notion of understandability, we argue that it is connected to many often-discussed and desirable properties of explanation (further details in App. C).

Human Alignment Users sometimes expect explanations that are aligned with their expectations. For example, the fact that highly salient words convey strong sentiment is taken as evidence for the quality of an explanation method by Li et al. (2016). In image classification, this concept is typically implemented as a pointing game between the high-saliency region and the segmentation mask of the predicted class (Fong and Vedaldi, 2017). However, alignment does not imply correctness, as the model could use any spurious correlations, which should be faithfully highlighted by the explainer. However, higher-alignment explanations are indeed *more understandable*, since by definition they agree more with human intuition. Thus, an alignment-based evaluation can be considered as one of understandability. Nonetheless, understandability can also be achieved by correcting human expectations, e.g., users realizing that punctuations are actually important for predictions (contrary to expectations).

Robustness It is often argued that explanations should be robust (Ghorbani et al., 2019) – similar inputs should induce similar explanations. However, robustness can be at odds with correctness: if the model truly applies vastly different logic for two very close inputs – such as a pair of inputs that only differ in the root feature of a decision tree – then their explanations should be distinct, as they are routed down two different sub-trees. Nonetheless, slow-varying explanations are generally easier to understand than those that change erratically and unpredictably (independent of their correctness), and thus robustness is related to understandability.

Counterfactual Similarity and Plausibility

Counterfactual explanations (e.g. Ross et al., 2021) indicate how the input should change in order to alter the model prediction. Besides the success rate of achieving target prediction, they are often evaluated on similarity (the magnitude of input change) and plausibility (the naturalness of the changed input). Both properties can serve as proxies for understandability: it is easier to relate an input to another similar and natural input than to a totally different or abnormal one. However, App. C presents two cases where they should *not* be similar or plausible but remain understandable, to highlight certain model behaviors.

8 Discussion and Conclusion

Traditionally, model explanations are evaluated on correctness (or faithfulness), i.e., whether they correspond to how models actually make predictions, e.g., reliance on spurious correlations (Zhou et al., 2022; Adebayo et al., 2022). Such evaluation, however, does not answer the equally important question of whether these (presumably correct) explanations are understandable. Even faithful explanations can lead users into error, if misunderstood (e.g., trusting a model incorrectly).

In a sense, the most correct explanation for an input is the literal trace of model computation, but it is also arguably the least understandable (or useful). As we abstract away from low-level details and use higher-level concepts such as word sentiment, the resulting explanation loses correctness but gains understandability. At the other extreme are explanations that are trivially understandable but completely wrong, such all attribution values being 0 (i.e., no feature impacts the model prediction). Thus, a trade-off often occurs between these two desiderata, and we need to choose a sweet spot.

Concretely, we propose EXSUM rules and rule unions, along with three quality metrics to formalize and evaluate understandability. Such rigorous investigations stand in contrast to current *ad hoc* practices, which are prone to yielding unreliable and coarse model understanding. For SST and QQP datasets, EXSUM demonstrates that our model understanding is quite limited and imprecise, even with very reasonable explanations. *Being aware of this is an asset*. While EXSUM helps us to recognize that our understanding is incomplete, it still helps uncover unexpected model behaviors that warrant further investigation.

Limitations and Ethical Impacts

Limitations

One notable requirement of EXSUM is the extensive human involvement in constructing and optimizing its rules. However, this process is necessary, as the alternative of generalizing from a few explanations has various flaws, depicted in Fig. 2 and Fig. 4. Practically, we spent about 3 hours on each rule union in Sec. 5, and our effort was streamlined by the systematic process and GUI presented in Sec. 4, which could be further improved by methods that automatically propose candidate rules.

In addition, another area requiring human involvement is the FEU feature definitions, which are often domain-dependent: both the sentiment score and the matching word features reflect the nature of the tasks. Other features may be necessary for other tasks. For example, in question-answering, one important FEU feature could be the kind of interrogative word used in the question (e.g., “what” vs. “when” vs. non-interrogative words). If important features are missed, the quality of the EXSUM rules – and, hence, the model understandings – will suffer accordingly.

Last, the difficulty of obtaining overall high-quality model understanding may result from the fundamental limitations of word-level attribution-based explanations, which cannot account for higher-level interactions. EXSUM could aid in the development of new explanation methods that are easier for humans to understand. As a first step, we explore defining and evaluating model understanding obtained from instance-based explanations with whole input as FEUs. App. D details the investigation, which raises questions such as the reliability of such explanations.

Ethical Impacts

As interpretability methods are increasingly deployed for quality assurance, auditing and knowledge discovery purposes, it is important to ensure the legitimacy of any conclusions drawn from explanations. While the correctness of these explanations is often studied, we argue their understandability should be equally emphasized, and evaluations with our newly proposed EXSUM framework and GUI reveal many problems of existing *ad hoc* procedures. Thus, a more careful treatment on the understandability aspect is necessary for well-calibrated model understandings and responsible model deployment in the real world.

Acknowledgment

This research is supported by the National Science Foundation (NSF) under the grant IIS-1830282. We thank the reviewers for their efforts.

References

- Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. *Advances in Neural Information Processing Systems*, 31.
- Julius Adebayo, Michael Muelly, Harold Abelson, and Been Kim. 2022. Post hoc explanations may be ineffective for detecting unknown spurious correlation. In *International Conference on Learning Representations*.
- David Alvarez-Melis and Tommi S Jaakkola. 2018a. On the robustness of interpretability methods. In *ICML Workshop on Human Interpretability in Machine Learning*.
- David Alvarez-Melis and Tommi S Jaakkola. 2018b. Towards robust interpretability with self-explaining neural networks. *Advances in Neural Information Processing Systems*, 31.
- Leila Arras, Ahmed Osman, Klaus-Robert Müller, and Wojciech Samek. 2019. [Evaluating recurrent neural network explanations](#). In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 113–126, Florence, Italy. Association for Computational Linguistics.
- Gagan Bansal, Tongshuang Wu, Joyce Zhou, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel Weld. 2021. Does the whole exceed its parts? The effect of AI explanations on complementary team performance. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–16.
- Jasmijn Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2963–2977.
- Ian Covert, Scott M Lundberg, and Su-In Lee. 2020. Understanding global feature contributions with additive importance measures. In *Advances in Neural Information Processing Systems*, volume 33.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

- Ruth C Fong and Andrea Vedaldi. 2017. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3429–3437.
- Amirata Ghorbani, Abubakar Abid, and James Zou. 2019. Interpretation of neural networks is fragile. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3681–3688.
- Miguel Grinberg. 2018. *Flask web development: Developing web applications with Python*. " O'Reilly Media, Inc."
- Peter Hase and Mohit Bansal. 2020. [Evaluating explainable AI: Which algorithmic explanations help users predict model behavior?](#) In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5540–5552, Online. Association for Computational Linguistics.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.
- Alon Jacovi and Yoav Goldberg. 2020. [Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness?](#) In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online. Association for Computational Linguistics.
- Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. 2016. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1675–1684.
- Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. 2019. Faithful and customizable explanations of black box models. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 131–138.
- Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. [Rationalizing neural predictions](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas. Association for Computational Linguistics.
- Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. [Visualizing and understanding neural models in NLP](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691, San Diego, California. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30.
- Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. [StereoSet: Measuring stereotypical bias in pre-trained language models](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5356–5371, Online. Association for Computational Linguistics.
- Weili Nie, Yang Zhang, and Ankit Patel. 2018. A theoretical explanation for perplexing behaviors of backpropagation-based visualizations. In *International Conference on Machine Learning*, pages 3809–3818. PMLR.
- Danish Pruthi, Rachit Bansal, Bhuwan Dhingra, Livio Baldini Soares, Michael Collins, Zachary C. Lipton, Graham Neubig, and William W. Cohen. 2022. [Evaluating Explanations: How Much Do Explanations from the Teacher Aid Students?](#) *Transactions of the Association for Computational Linguistics*, 10:359–375.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. “Why should I trust you?” Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 32.
- Alexis Ross, Ana Marasović, and Matthew Peters. 2021. [Explaining NLP models via minimal contrastive editing \(MiCE\)](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3840–3852, Online. Association for Computational Linguistics.
- Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. 2016. Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2660–2673.
- Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations*.
- Robert Wolfe and Aylin Caliskan. 2021. [Low frequency names exhibit bias and overfitting in contextualizing language models](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 518–532, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. 2021. [Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6707–6723, Online. Association for Computational Linguistics.

Yiming Zheng, Serena Booth, Julie Shah, and Yilun Zhou. 2021. The irrationality of neural rationale models. *arXiv preprint arXiv:2110.07550*.

Yilun Zhou, Serena Booth, Marco Tulio Ribeiro, and Julie Shah. 2022. Do feature attribution methods correctly attribute features? In *AAAI Conference on Artificial Intelligence*.

A Real World Use Cases for Explanations

Here, we discuss several scenarios in which people use local explanations to understand models, and argue that people invariably derive *generalized* model understanding from these explanations.

A.1 Spurious Correlation Identification

Natural datasets can contain many spurious correlations. For example, in a COVID-19 chest X-ray dataset, most positive images (i.e., patients diagnosed with COVID-19) come from a pneumonia-specializing hospital and contain a watermark of the hospital name, while most negative images from other hospitals do not. Thus, a model could achieve very high accuracy by simply detecting the watermark rather than genuine medical signals. Similar spurious correlations could also be present in the text domain, such as the correlation between an exclamation mark and the positive sentiment class, or between the word “*not*” and the contradiction class in natural language inference.

It is crucial for people to be aware of the shortcuts that models may take, and one possible way to highlight such behaviors is via feature attribution, which in the examples above would assign an abnormally high score to the watermark region, exclamation mark, or the word “*not*.” Assuming the explanations do indeed exhibit such patterns, when people claim a model relies on spurious correlation, they mean this in a general sense: for example, the model is likely to focus on the watermark in *any* image that contains it, rather than in only a specific set of images.

A.2 Fairness Assurance

Similar to spurious correlation features, other features should not have a high impact, but for reasons of fairness. For example, decisions made by a loan approval model should not be affected by gender³, therefore the gender feature should not have a high attribution score.

If we observe that the gender of one applicant heavily impacts the model’s decision, we may suspect the model is discriminative; conversely, observing that it has minimal impact could increase our assurance of the model’s fairness. However, such single-instance observations are fundamentally exploratory, and claims about the model’s fairness or discrimination must be established using a *population* of instances to determine whether the trend persists generally.

A.3 Model-Guided Human Learning

In some cases, a very accurate and “super-human” model could be a source for knowledge discovery. Consider the task of early-stage cancer detection from CT scans, which is challenging for doctors. If a label is generated from follow-up visits tracking whether patients develop cancer after a certain number of years, a model achieving better test accuracy than doctors is likely to use certain cues that would be missed by humans or not known to be linked to cancer.

For these models, explanation methods such as saliency maps could be used to help doctors make better diagnoses, or assist scientists in the creation of new pathological theories. Similarly to the above two use cases, *generalized* model understanding across different inputs are necessary, because doctors need to apply what they have learned to new patients, and scientists require new theories to hold broadly.

B Additional Evaluation Details

Tab. 3 summarizes the key parameters of our experiment. Both saved models are publicly accessible from Huggingface Hub, and the model names in the table are links to the respective model checkpoints. For normalization, we divided all explanation values for all test set instances by a single scaling factor such that the maximum magnitude of new explanations is 1.

B.1 SST Sentiment Classification

For the explainer, we used the PartitionSHAP algorithm implemented by the shap repository⁴. Fig. 6 shows the explanations on three sentences (after normalization).

³There could be other features that correlate with gender, such as job title, but we ignore such possibilities for simplicity.

⁴<https://shap.readthedocs.io/en/latest/>

Task	Dataset	Model	Acc.	F1	Explainer
Sentiment	SST-2	RoBERTa	95.6%	0.957	SHAP
Paraphrase	QQP	BERT	90.7%	0.875	LIME

Table 3: A summary of tasks, models (fine-tuned on respective datasets), and explainers for the two case studies.

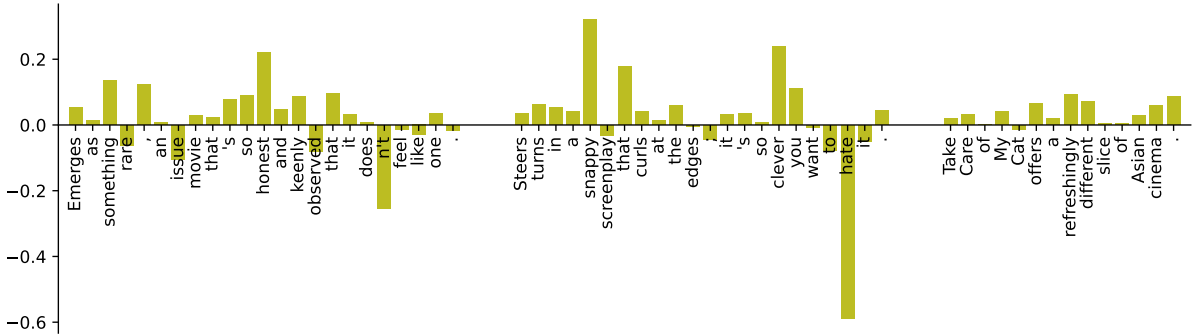


Figure 6: SHAP explanation visualization for three SST inputs.

B.1.1 Details of Current Practices

Here, we provide an extended description of the three current practices, and how they are applied on the handful of selected examples, collectively called the “sample” below.

The first method, “belief guided” (BG), represents the practice wherein the user has some expectations (or beliefs) about the attributions of certain words, and modifies (or updates) them after observing explanations on some actual test inputs. It operates differently for the two rules on positive-sentiment and stop words, as follows.

1. For positive-sentiment words, the prior belief is that a word with a higher sentiment score (one of the FEU features provided by the SST dataset) should also receive more positive attribution. This leads to a rule that applies to all words with a sentiment score greater than α , and has a behavior function that outputs a constant range of $[\beta, 1]$ (recall that SHAP attribution values are normalized to $[-1, 1]$ range). It then computes the value of α as the mean sentiment score and β as the mean attribution value for all words in the sample with positive sentiment scores.
2. For the stop words – defined as those with parts of speech AUX, DET, ADP, CCONJ, SCONJ, PRON, PART, and PUNCT – it has a prior belief that they should have a attribution value range of $[-0.05, 0.05]$ (i.e., not important to model prediction), and computes the observed attribution range $[\alpha, \beta]$ for stop words in the sample. The final behavior range as predicted by the behavior function of this rule is the average of these two: $[-(0.05 + \alpha)/2, (0.05 + \beta)/2]$.

The second method, “quantile fitting” (QF), represents the practice wherein the user fully follows the observed data without any prior beliefs. Specifically, for a set of words, it collects all attribution values for words within the set and then creates a rule that applies to this set, with the behavior function predicting a constant range of 5% to 95% quantile of these attribution values. For the two rules for positive-sentiment and stop words, the set of words (and hence the applicability functions) is defined in the same way as for the BG method above.

The last method, “word-level” (WL), can be considered a more extreme version of QF, where the user not only lacks any prior expectations for the explanations but also considers each word individually. For example, if the user observes that the word “brilliant” has an attribution value of 0.5 in one sentence and the word “fantastic” has attribution of 0.8 in another, they would *not* conclude that other, similarly positive words would have attributions approximately within the range of $[0.5, 0.8]$. Specifically, for every distinct word w in the sample, this method builds a rule that applies only to that word, with a constant behavior

function that outputs a range of $[\min(s_w) - 0.03, \max(s_w) + 0.03]$, where s_w is the list of attributions received by different occurrences of w . In many cases, especially given a small sample, word w only appears once, in which case s_w is a list containing only that attribution value.

Tab. 4 presents the metric values of the above methods. Fig. 4 of Sec. 5.1 depicts a graphic summary.

K	pick	belief-guided		quantile-fitting		word-level
		positive	stop word	positive	stop word	seen words
1	SP	10, 72, 50	49, 45, 65	10, 63, 44	49, 63, 45	28, 51, 61
	RND μ	12, 63, 57	49, 58, 53	12, 45, 56	49, 73, 33	17, 41, 68
	RND σ	6, 25, 27	0, 9, 6	6, 32, 29	0, 24, 21	6, 12, 10
10	SP	10, 61, 61	49, 47, 63	10, 78, 34	49, 72, 38	49, 66, 48
	RND μ	10, 71, 52	49, 56, 56	10, 75, 32	49, 84, 25	41, 68, 48
	RND σ	0, 6, 7	0, 4, 4	0, 9, 10	0, 3, 2	2, 1, 2
30	SP	10, 64, 59	49, 50, 60	10, 88, 17	49, 82, 29	57, 73, 42
	RND μ	10, 66, 56	49, 57, 55	10, 82, 26	49, 86, 24	51, 78, 39
	RND σ	0, 4, 5	0, 1, 2	0, 6, 7	0, 2, 2	2, 3, 2

Table 4: Coverage, validity, and sharpness (percentage) of model understanding with *ad hoc* current practice. “SP” refers to the submodular pick procedure, and “RND” refers to the random sampling procedure. The latter also shows mean μ and stdev σ across five runs.

B.1.2 Complete Rule Union Description

Below, we present the details of the construction process for rules not discussed in Sec. 5.1.

- **Rule 1: Negation words have negative saliency.** We found that negation words – *not*, *n’t*, *no*, *nothing* and those with NEG dependency tag – almost invariably receive (sometimes highly) negative saliency, regardless of the sentence label or sentiment of the word being modified. We create a rule that predicts a constant behavior range $[-1.0, 0.002]$, with 89.5% validity and 65.1% sharpness. Although the validity is under our 90% target, we found that to make it higher, the upper limit of the behavior range needs to be 0.1, which results in an extremely low sharpness of 11%. Thus, we decided against it.
- **Rule 5: Person names have positive saliency.** During our initial inspection, we found several cases where the name of a person (e.g. director or actor) have positive saliency values. Thus, we create this rule from the NER tag, covering 2.3% of words. However, after parameter tuning, we found that while many of the words have positive saliency, the correct characterization is that they all have small saliency values, as a behavior range of $[-0.06, 0.1]$ achieves 91.6% validity. However, since SHAP saliencies are mostly concentrated around 0, this range achieves a meager sharpness of 26.8%. Despite this, we still decide to keep it.
- **Rule 7: Zero-sentiment words have small saliency.** Besides stop words, we should expect words that do not carry sentiment, such as most nouns and verbs (e.g., *movie* and *get*), to have small saliency magnitudes. Due to the wide range of words applicable under this rule, we choose the saliency range to be $[-0.15, 0.15]$ for $\geq 90\%$ validity, but this range yields lowest sharpness of 13.5%.
- **Rule 8, 9: Weakly positive/negative words have weakly positive/negative saliency.** Finally, we set up two rules to capture words that have sentiment of neither zero (covered by Rule 19) nor high-magnitude (covered by Rule 3 – 5). To achieve 90% validity, we require a behavior range of $[-0.11, 1]$ for weakly positive words and $[-1, 0.05]$ for weakly negative words, unfortunately again with quite low sharpness. Notably, both ranges need to “spill over” to the other side of zero for the required validity.

B.2 QQP Paraphrase Detection

For the explainer, we used the LIME algorithm implemented by the `lime` repository⁵. Fig. 7 depicts the explanations on two pairs (after normalization).

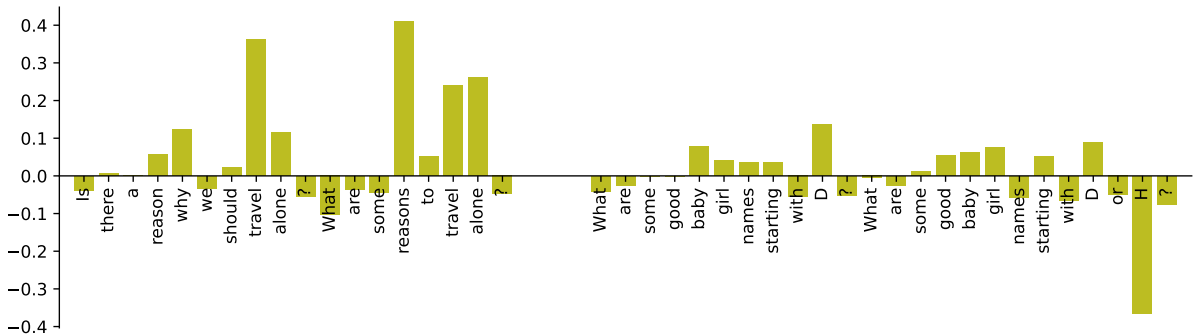


Figure 7: LIME explanation visualization for two QQP pairs.

B.2.1 Complete Rule Union Description

Below, we present details of the construction process for rules not discussed in Sec. 5.2.

- **Rule 4, 5: Saliency for trailing question marks.** Since the dataset is composed of pairs of questions, the vast majority of sentences conclude with question marks. These should be purely decorative and syntactic, and so should have small saliency, similar to stop words. However, we observe that the saliencies assigned to them for positive and negative predictions are very different, so we create two rules for these two cases. With a 90% validity target, the saliency range is $[-0.04, 0.03]$ for negative predictions and $[-0.07, 0.06]$ for positive predictions. Again, the saliencies for positive predictions demonstrate more variation than those for negative ones.
- **Rule 6, 7: Saliency for stop words.** Similar to SST, we use these two rules to ensure stop words should *not* be influential. We split the stop word group into finer segments by part of speech, to achieve higher sharpness. On average, the range is $[-0.07, 0.03]$ for negative predictions and $[-0.09, 0.1]$ for positive predictions, which again demonstrate a much higher degree of variation.
- **Rule 8, 9: Saliency for negation words.** In the SST case, we found that negation words typically have negative saliency regardless of the sentiment label, and test whether this holds for QQP as well. Following on our previous findings, we use two rules to separately model inputs of positive and negative predictions. We find that the range is $[-0.1, 0.24]$ for positive predictions and $[-0.21, 0.01]$ for that for negative predictions. Curiously, the same negative saliency trend is preserved here as well, but only for inputs with negative predictions.
- **Rule 10, 11: Saliency for everything else.** Finally, we designed two lowest-precedence “catch-all” rules to complete the coverage. The range for positive prediction FEUs is $[-0.13, 0.25]$. For negative prediction inputs, we find that breaking them according to different parts of speech (nouns, verbs, adjectives, and everything else) is helpful, with verbs having a particularly narrow saliency range of $[-0.05, 0.05]$. On average, the saliency range is approximately $[-0.09, 0.05]$.

C Understandability as a Unified Theme

In this section, we elaborate on how understandability is the unified theme behind many properties of explanations that seem “orthogonal” to correctness. Specifically, we discuss three properties: human alignment, robustness, and counterfactual similarity and plausibility.

⁵<https://github.com/marcotcr/lime/>

C.1 Human Alignment

Many prior works have assessed how much explanations agree with human expectation. For example, Li et al. (2016) observed that the word “hate” contributes the most to a negative sentiment prediction in many inputs, and used it to argue the explanation is correct. In a similar sentiment classification task, Bastings et al. (2019) used the high degree of overlap between the extracted rationale and strong-sentiment words to argue the superior quality of a neural rationale model (Lei et al., 2016). In computer vision, this alignment is often implemented as a pointing game that computes the intersection-over-union (IoU) metric between the salient region and the semantic segmentation mask of the predicted class (Simonyan et al., 2014; Fong and Vedaldi, 2017), as shown in Fig. 8. For a model that predicts breast cancer onset using patients’ genetic information, Covert et al. (2020) demonstrated that many of the influential genes identified by their explainer were indeed known to be associated with the disease.

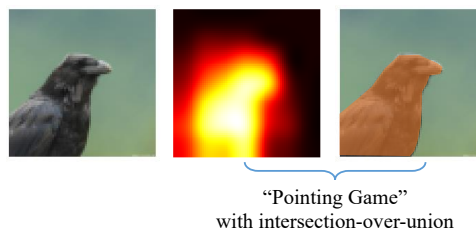


Figure 8: A pointing game used to quantify human alignment for visual explanations.

As discussed in App. A.1, models could use any unexpected spurious correlation, such as the green background in Fig. 8. For these models, correct explanations should have low alignment scores. When correctness (or faithfulness) is the sole desideratum of interpretability methods, it is unclear what purposes these alignment evaluations serve. Some authors (e.g. Jacovi and Goldberg, 2020) have even argued they are fundamentally misleading and flawed in nature as they focus on *plausibility*, which is sometimes at odds with the goal of correctness.

However, from the perspective of *understandability*, high-alignment explanations are arguably very understandable, simply because they align closely with human expectation. Thus, given the same level of correctness, a higher-alignment explainer may be preferable.

C.2 Robustness

Besides human alignment, robustness – i.e. that similar inputs should have similar explanations – is also argued to be a favorable property for explanation. For example, Ghorbani et al. (2019) argued that explanations are fragile due to their adversarial vulnerability, Alvarez-Melis and Jaakkola (2018a) empirically estimated the Lipschitz constant for many explainers, and Alvarez-Melis and Jaakkola (2018b) proposed an inherently interpretable model that is explicitly regularized for explanation robustness.

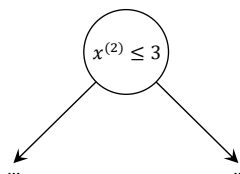


Figure 9: A decision tree that splits on the second feature at the root node.

Robustness generally conflicts with correctness. If, for two inputs, the model is using distinct reasoning patterns, the correct explainer should faithfully report distinct explanations for them. One straightforward example is the decision tree model shown in Fig. 9, where the root node splits on the second feature at a threshold value of 3. For two inputs x_1 and x_2 that agree on all features except the second one, with $x_1^{(2)} = 2.99$ and $x_2^{(2)} = 3.01$, since they are sent down two different sub-trees at the very beginning, the model is likely to use for totally different features.

Nonetheless, as implicitly argued by the works above, erratic model behaviors are less understandable because they make it more difficult to identify generalizable patterns compared with slowly varying explanations in the input space. Thus, robustness is another aspect of the same understandability desideratum.

C.3 Counterfactual Similarity and Plausibility

Unlike feature attribution explainers that assign importance to individual features, counterfactual (CF) explainers (e.g., Ross et al., 2021) directly generate whole inputs but for a target predicted class. Thus, a CF explanation indicates how to cross the decision boundary from the input.

Naturally, the fundamental requirement of CF explanations is achieving the target prediction, which is typically known as validity. However, this is trivially satisfiable by simply finding a training instance with the target prediction, along with other ways such as creating adversarially perturbed or nonsensical inputs. Thus, two additional requirements are often enforced: similarity and plausibility. The former says that the CF explanation should be close to the original input (with regard to, for example, edit distance), and the latter says the CF explanation should be plausible, or natural. Tab. 5 depicts various CF explanations and their satisfaction of the three requirements.

Input: This restaurant is the best I have been, with especially great food.				
CF	Type	Val.	Sim.	Plau.
This restaurant is the <i>worst</i> I have been, with especially <i>terrible</i> food.	“good” CF	✓	✓	✓
Rude service!	training set look-up	✓	✗	✓
This <i>resturant</i> is the best I have been, with especially great food.	adversarial typo injection	✓	✓	✗
Fjwpeaf faweekl fka erj sfdlk erjlm adl erio fd	nonsensical inputs	✓	✗	✗

Table 5: CF explanations that are all valid but differ in similarity and plausibility metrics.

Validity for CF can be considered as the correctness analogy for feature attribution, but the purposes of similarity and plausibility are not readily apparent. As CF explanations represent ways to cross the decision boundary, people need to meaningfully understand how the CF instance is related to the original input. It is difficult to relate two dissimilar instances, and an implausible CF instance is generally unexpected. Thus, similarity and plausibility are required to make CF explanations more understandable.

Interestingly, if our true goal is the understandability of the relationship between the input and its CF explanation, there are cases where similarity or plausibility is *not* desirable. First, consider a sentence length classifier that predicts positive for sentences of at least 10 words, and negative otherwise. Given an input of three words, the CF explainer should generate *dissimilar* CF instances of at least 10 words in order to correctly illustrate the decision boundary, while instances of even more words would be helpful for understanding the “at least 10 words” logic. Second, consider a classifier trained on a typo-free dataset and having high probability of making mistakes on inputs that contain typos. To illustrate this behavior, CF explanations should contain randomly (not adversarially) injected typos, which are *implausible*, but useful as long as the typo injection is understood by people.

D Additional Details on Instance-Based Explanations

In this section, we describe our initial attempt at extending the EXSUM framework to another type of explanations: instance-based explanations (IBE). The IBE for an input x is a set of instances and their predictions $\{(x_i, \hat{y}_i)\}$, where x and x_i are semantically related (e.g., negation). We define \hat{y}_i as the predicted probability of positive class.

Type	$b(\hat{y})$	ν	σ
Entity change	$\hat{y} \pm 0.05$	91.4	56.5
Minor insert	$\hat{y} \pm 0.05$	89.1	57.3
Negation	other-side(\hat{y})	30.4	50.0
Negation	same-side(\hat{y})	69.6	49.9
Negation (≤ 6 words)	other-side(\hat{y})	56.2	49.7

Table 6: Instance-based explanation metrics on SST.

We use POLYJUICE (PJ, Wu et al., 2021) to generate instances of three semantic operations. *Entity change* replaces a proper noun (e.g. actor name) with another using “lexical” mode of PJ. *Minor insert* adds a short text to the sentence using “insert” mode. *Negation* generated a negated version of the input using “negation” mode. For each operation type, our expectation for model behavior is formalized as a range $b(\hat{y})$ on \hat{y} . We expect the prediction to be unchanged by the first two operations allowing for a margin of 0.05, but changed to the other side of 0.5 by negation. We then define validity $\nu = \mathbb{E}_{\hat{Y}, \hat{Y}_i} [\mathbb{1}_{\hat{Y}_i \in b(\hat{Y})}]$ and sharpness $\sigma = 1 - \mathbb{P}_{\hat{Y}}[b(\hat{Y})]$ analogously.

Tab. 6 summarizes the results. While our expectation is mostly confirmed for entity change and minor insert, it is notably violated in the case of negation, with only 30.4% validity, indicating model prediction is on the *same* side 69.6% of time. Upon further evaluation, we find that validity drops with sentence length, with short sentences of six words or fewer having much higher validity (for other-side). Since the PJ rewriting model is learned rather than manually defined and negation is more complex than the other two operations, there are two failure modes, as presented in Tab. 7. In the first, a negation is applied to the input sentence, but on a part irrelevant to the sentiment. In the second, the generated sentence is not a negation of the input by any reasonable standard.

These examples highlight the importance of clearly defining the operation: rather than a generic

Input sentence	“Negated” sentence
Human Nature initially succeeds by allowing itself to go crazy , but ultimately fails by spinning out of control .	Human Nature initially succeeds by allowing itself to go crazy , but ultimately fails by not coming to consciousness .
This may be the dumbest , sketchiest movie on record about an aspiring writer ’s coming-of-age .	This may be the dumbest , sketchiest movie on record , not an aspiring writer ’s coming-of-age .
Before long , the film starts playing like General Hospital crossed with a Saturday Night Live spoof of Dog Day Afternoon .	Before long , the film starts playing like nothing crossed with a Saturday Night Live spoof of Dog Day Afternoon .
A startling and fresh examination of how the bike still remains an ambiguous icon in Chinese society .	A startling and fresh examination of how the bike still seems to be an ambiguous icon in Chinese society .
Never engaging , utterly predictable and completely void of anything remotely interesting or suspenseful .	Not engaging , utterly predictable and completely void of anything remotely interesting or suspenseful .
Between the drama of Cube ?	Are there no interesting problems?
Tailored to entertain !	No tails !

Table 7: Failure cases of POLYJUICE negations. The first half shows examples where the negation is irrelevant to the sentiment. The second half includes examples where the negation fails to appear.

negation, we would need the negation to happen on the “sentiment-carrying” part. It is also crucial to ensure that the generator is of a high quality in order to minimize the chance of generating nonsensical outputs. Despite many advances in generative language modeling, it has been shown to be undesirable in many ways (e.g., [Holtzman et al., 2019](#)), all of which affect the quality of the counterfactual explanation.

At a high level, IBE explains the local prediction by illustrating ways to cross (e.g., negation) or not cross (e.g., entity change) the decision boundary in the (very) high-dimensional input space. However, as the negation case indicates, we must be careful about the exact definition of the rewriting (e.g., negating any part of the input or the “sentiment-carrying” part only), as it could have a significant impact on the conclusion. Furthermore, it is difficult for any rewriting mechanism to achieve 100% validity due to the high dimensionality, the multitude of possible ways of rewriting, and the imperfection of the model. Focusing only on the mistakes (or ignoring them altogether) yields incomplete model understanding. Instead, the validity metric, which indicates the *generalized* model behavior, should be used to.

There are many potentially fruitful directions for future work. First, the quality of instances obviously depends on the generative models, which, while impressive, are known to be flawed in many ways (e.g., [Holtzman et al., 2019](#); [Nadeem et al., 2021](#); [Wolfe and Caliskan, 2021](#)). Second, each rule essentially covers the entire input space. Partitioning the input space in some way may allow for identification of both more and less consistent areas, which makes the applicability function much more difficult to define as it now takes whole sentences rather than individual words. Finally, unlike feature attribution, which conveys the single notion of “importance,” different instances of the same input can reveal different aspects of model behavior, calling for a potentially different definition of coverage, which measures completeness of understanding.