

Transparent Neighborhood Approximation for Text Classifier Explanation by Probability-based Editing

Yi Cai

Dept. of Math. and Comp. Science
Freie Universität Berlin
Berlin, Germany
yi.cai@fu-berlin.de

Eirini Ntoutsi

Research Institute CODE
Universität der Bundeswehr München
Munich, Germany
eirini.ntoutsi@unibw.de

Arthur Zimek

Dept. of Math. and Comp. Science
University of Southern Denmark
Odense, Denmark
zimek@imada.sdu.dk

Gerhard Wunder

Dept. of Math. and Comp. Science
Freie Universität Berlin
Berlin, Germany
gerhard.wunder@fu-berlin.de

Abstract—Recent literature highlights the critical role of neighborhood construction in deriving model-agnostic explanations, with a growing trend toward deploying generative models to improve synthetic instance quality, especially for explaining text classifiers. These approaches overcome the challenges in neighborhood construction posed by the unstructured nature of texts, thereby improving the quality of explanations. However, the deployed generators are usually implemented via neural networks and lack inherent explainability, sparking arguments over the transparency of the explanation process itself. To address this limitation while preserving neighborhood quality, this paper introduces a probability-based editing method as an alternative to black-box text generators. This approach generates neighboring texts by implementing manipulations based on in-text contexts. Substituting the generator-based construction process with recursive probability-based editing, the resultant explanation method, XPROB (explainer with probability-based editing), exhibits competitive performance according to the evaluation conducted on two real-world datasets. Additionally, XPROB’s fully transparent and more controllable construction process leads to superior stability compared to the generator-based explainers.

Index Terms—Explainable AI, Local explanation, Counterfactual, Neighborhood approximation, Text classification

I. INTRODUCTION

Advances in machine learning, particularly deep learning, have enabled applications in various real-world scenarios [1]. Accompanying improvements in performance, the increasing model complexity and the exploding number of parameters [2] prohibit human users from comprehending the decisions made by these data-driven models, as the decision rules are implicitly learned from the data presented. The absence of reasoning for model decisions keeps raising concerns about the transparency of AI-driven systems [3]. Encouraged by these

concerns, extensive research in explainable AI (XAI) emerges, intending to uncover the decision-making process hiding under the machine learning black boxes.

A major focus in enhancing explainability is on identifying evidence supporting model decisions. Known as attribution methods, these approaches assign scores to individual input features of a specific decision, referred to as the *explicand*. These attribution scores quantify each feature’s contribution to the target decision. A common solution for deriving feature attribution involves analyzing the correlation between input features and model outcomes [4]. Requiring only query-level access to the target model, this kind of method delivers explanations in a post-hoc manner without delving into model internal details. Query-level access means that the to-be-explained model can only be interacted with through its input and output interfaces. Making minimal, if any, assumptions about the model being explained, these solutions are termed model-agnostic approaches.

In the explanation process, a model-agnostic approach begins by creating a synthetic set (termed the *neighborhood*) containing instances surrounding the explicand [5], [6]. These instances are then labeled with outputs from the target black box, providing pseudo labels that reflect the model’s behaviors. During the synthetic set construction, the locality constraint [7] emphasizes the spatial closeness of the generated instances to the explicand, ensuring that the derived explanation concentrates exclusively on model behaviors relevant to the inquired decision. Once the neighborhood is determined, explanatory information is indirectly extracted by investigating the impacts of altered features on model outcomes.

The quality of generated instances, which are the sole reflections of model behaviors, significantly affects the performance of model-agnostic explainers. Although neighborhood construction appears relatively simple for structured data, such as tabular [7] and image [8] data, solving the same task for textual data is nontrivial. The strong inherent feature interaction in

Yi Cai and Gerhard Wunder were supported by the Federal Ministry of Education and Research of Germany (BMBF) in the program of “Souverän. Digital. Vernetzt.”, joint project “AlgenCY: Chances and Risks of Generative AI in Cybersecurity”, project identification number 16KIS2013. Gerhard Wunder was also supported by BMBF joint project “6G-RIC: 6G Research and Innovation Cluster”, project identification number 16KISK020K.

texts degrades the effectiveness of straightforward generation approaches like random word dropping [5]. A recent line of works [9], [10] proposes using generative models for neighborhood construction, which produces realistic neighboring texts – semantically meaningful and grammatically correct – thereby enhancing explanation quality. However, the deployment of non-self-explainable components introduces additional opacity to the constructed neighborhoods and, consequently, to the explanations derived from them.

In response to transparency concerns in the explanation procedure, this paper proposes XPROB¹ (explainer with probability-based editing), an approach generating realistic texts for neighborhood construction through editing guided by local n -gram contexts. Our experiments on two real-world datasets empirically show the competitive performance of the proposed explainer without relying on black box generators. The experimental results also validate concerns about the inherent opacity of generator-based explainers by highlighting their complicated dependencies on generative models. Furthermore, XPROB’s neighborhood construction is more controllable as its probability-based editing is a deterministic process following explicit rules, contrasting with generator-based solutions, where the effects of latent space perturbations on generated texts are unforeseeable.

II. RELATED WORK

Constructing neighborhoods is a fundamental task in deriving model-agnostic explanations. The earlier attempts to accomplish the task can be summarised as input perturbation [11], which is well-suited for structured data. This approach generates neighboring instances by altering the feature values of an explicand. Specifically, input perturbation applies slight fluctuation in the vicinity of a given input for numeric features [7] or samples categorical feature values from some empirical distribution [12]. While the strategy appears straightforward, perturbing input adheres to the locality constraint.

Building on the concept of input perturbation, LIME [5] constructs neighborhoods of text by randomly dropping words from an explicand, repeating the process until sufficient instances are collected. Similarly, input occlusion [13] perturbs text by replacing words with padding tokens. Although both methods generate synthetics similar to the original explicand, they face limitations with textual data due to strong inherent feature interactions. While adhering to the locality constraint, the randomly perturbed texts can drive the concentration of explanations away from the natural data manifold [14], [15], potentially leading to explanations that do not accurately represent model behaviors [16]. Besides, the feasibility of these methods is constrained by the length of the target text. For explicands with only a few words, input perturbation struggles to gather sufficient unique samples for further analysis.

XSPELLS [10], [17] addresses the limitations of input perturbation by deploying a variational autoencoder [18] for

textual neighborhood construction. This generator-based explainer samples instances in a numeric latent space maintained by the generative autoencoder, selecting neighbors based on their proximity to the latent representation of the explicand. The sampled latent vectors are then decoded into texts by the generator for succeeding explanation processes. A follow-up work, XPROAX [9], refines the latent space sampling with a two-staged progressive approximation. Specifically, it employs iterative latent space interpolations between points with distinct labels, which effectively highlights decision boundaries relevant to the queried decision. XPROAX also specifies the locality-preserving and reconstruction properties as key criteria in selecting generative models for model-agnostic explanations. Prior applications of the generator-based construction strategy include ABELE [19], which explains image classifiers by refining embedding sampling with a genetic algorithm, and methods for tabular data classifiers [16], [20], which aims at aligning the distribution of generated instances with the data manifold. Moreover, recent research extends the use of generators to craft counterfactuals [21], [22], providing an alternative form of explanation that explores minimal changes needed to alter the original outcome.

Despite the popularity of generators in tasks related to explainability, the heavy reliance of generator-based explainers on these opaque components undermines the transparency of the explanations derived, which requires further discussion. In fact, explaining generative models remains an open question [23], particularly challenging due to the complex nature of their outputs, such as texts and images. Moreover, generative models generally require more information than classifiers to operate on the same data manifold. This is evidenced by the fact that generative models can be more readily fine-tuned for classification tasks [24], but not the other way around, implying that generators often possess capabilities beyond classifiers. Consequently, generator-based attribution methods risk complicating the explanation process by explaining one black box with another, potentially more complicated one.

III. EXPLAINING WITH PROBABILITY-BASED EDITING

Given a text classifier $f(\cdot)$ and an explicand consisting of a sequence of tokens $\mathbf{x} = (w_1, w_2, \dots, w_n)$, the objective of feature attribution methods is to determine a vector ξ , whose elements indicate the contributions of input features to the decision² $y = \arg \max f(\mathbf{x})$. To extract explanatory information under query-level access, XPROB constructs a neighborhood $N(\mathbf{x})$ around the explicand, then quantifies feature contributions indirectly using a surrogate $g(\cdot)$ (details about the extraction are presented in Section III-C). The surrogate model $g(\cdot)$ is a self-explainable weak learner, trained on the pseudo-labeled neighborhood, and designed to emulate the local behavior of the black box model $f(\mathbf{x})$. Consistent with prior research [25], neighboring texts are defined as texts sharing similar components and meanings.

¹Source code is available at: <https://github.com/caiy0220/XPROB>.

²For simplicity, we assume a binary classification problem: $y \in \{+, -\}$.

A. Probability-based Editing

While input perturbation fails to generate synthetics representing the underlying data manifold of a classification task, adopting generative models produces realistic neighboring texts. This can be formally described as creating a synthetic set where each element $\forall \hat{x} \in N(\mathbf{x})$ satisfies $\hat{x} \sim \mathcal{X}$, with \mathcal{X} denoting the target text domain. A text generator is trained to learn the conditional probability distribution, allowing it to extend a given prompt with the most likely word:

$$w_{l+1} = \arg \max_w P(w|w_1, w_2, \dots, w_l)$$

Given that text generators function as probability distribution estimators, we propose an editing approach driven directly by empirical probability distributions as a transparent alternative.

Different from the objective of extending given prompts, text manipulation for neighborhood construction involves more operation than appending tokens to existing texts and should align with the locality constraint. Therefore, we redefine the editing task to identify the optimal manipulation operation that integrates a selected token from the explicand $w \in \mathbf{x}$ into a prototype $\hat{x} = (\hat{w}_1, \hat{w}_2, \dots, \hat{w}_l)$. The integration maximizes the probability of the resultant text:

$$\mathcal{O}(w, \hat{x}) = \arg \max_{1 \leq i < j \leq l} P(\hat{w}_1, \dots, \hat{w}_i, w, \hat{w}_j, \dots, \hat{w}_l) \quad (1)$$

where the probability $P(\hat{w}_1, \dots, \hat{w}_i, w, \hat{w}_j, \dots, \hat{w}_l)$ is empirically estimated on a corpus $X_p \sim \mathcal{X}$ sampled from the target text domain. The manipulation operation, defined by the indices i and j , refers to either inserting at position i of the prototype when $j = i + 1$, or replacing a contiguous subsequence $\hat{x}_{i+1}^{j-1} = (\hat{w}_{i+1}, \dots, \hat{w}_{j-1})$ in \hat{x} (denoted by $\hat{x}_{i+1}^{j-1} \prec \hat{x}$) otherwise. A prototype, which is a neighboring counterfactual, is selected from the retained corpus X_p (see further details about prototype selection in Section III-B). By recursively integrating explicand components to prototypes, XPROB produces a series of neighboring texts that share components with \mathbf{x} , performing a gradual transition from one class (the label assigned to \hat{x}) to another (the label of \mathbf{x}) that highlights the target decision boundary.

Implementing manipulation as outlined in Eq. 1 ensures the grammatical correctness of the generated texts. However, using the entire sequence as context restricts the process to only reproducing texts present in X_p . To enable exploration beyond existing token combinations, the editing process considers local n -gram contexts [26] instead of whole sequences. Specifically, the n -gram context consists of the preceding sequence \hat{x}_{i-n+1}^i and the succeeding sequence \hat{x}_j^{j+n-1} for a manipulation operation specified by i and j . Updating the editing objective with the local n -gram context yields:

$$\mathcal{O}(w, \hat{x}) = \arg \max_{1 \leq i < j \leq l} P_{pre}(w|\hat{x}_{i-n+1}^i) P_{suc}(w|\hat{x}_j^{j+n-1})$$

The optimal operation can be identified in quadratic time w.r.t. the prototype length. Denoting word sequence concatenation

as $\mathbf{x}_a^b + \mathbf{x}_c^d = (w_a, \dots, w_b, w_c, \dots, w_d)$, the conditional probability of integrating w given the *preceding* and *succeeding* words can be written as:

$$P_{pre}(w|\hat{x}_{i-n+1}^i) = \frac{P(\hat{x}_{i-n+1}^i + (w))}{P(\hat{x}_{i-n+1}^i)}$$

$$P_{suc}(w|\hat{x}_j^{j+n-1}) = \frac{P((w) + \hat{x}_j^{j+n-1})}{P(\hat{x}_j^{j+n-1})}$$

Albeit not explicitly given, the above conditional probabilities can be estimated using the retained corpus X_p :

$$P_{pre}(w|\hat{x}_{i-n+1}^i) = \frac{|\{\mathbf{x}|\mathbf{x} \in X_p, (\hat{x}_{i-n+1}^i + (w)) \prec \mathbf{x}\}|}{|\{\mathbf{x}|\mathbf{x} \in X_p, \hat{x}_{i-n+1}^i \prec \mathbf{x}\}|}$$

$$P_{suc}(w|\hat{x}_j^{j+n-1}) = \frac{|\{\mathbf{x}|\mathbf{x} \in X_p, ((w) + \hat{x}_j^{j+n-1}) \prec \mathbf{x}\}|}{|\{\mathbf{x}|\mathbf{x} \in X_p, \hat{x}_j^{j+n-1} \prec \mathbf{x}\}|}$$

For subsequences $\hat{x}_{i-n+1}^i + (w)$ absent in X_p , we assign a minimal value $\epsilon = (|X_p| + 1)^{-1}$ instead of 0 to avoid the probability collapsing to zero during the multiplication of P_{pre} with P_{suc} . Given the minimal value, we set the threshold for a valid edit at ϵ^2 . This threshold value implies that an edit is unfeasible if the target token w never appears in the context formed by either the preceding or succeeding texts according to X_p . The choice of n , which determines the range of local contexts, balances between semantic correctness and feasibility of the edit.

To allow manipulation at the beginning or the end of a text, n padding tokens $\langle PAD \rangle$ are concatenated to both the head and the tail of a prototype. However, this introduces a bias towards the padding tokens due to their omnipresence, which inadvertently induces the manipulation process to replace an entire text sequence with w , considering only the padding tokens as the local context. Such an abrupt change, which replaces a whole sentence, contradicts the intention of implementing gradual modifications. To address this issue, a penalty based on edit length is incorporated into the manipulation objective:

$$\mathcal{O}(w, \hat{x}) = \arg \max_{1 \leq i < j \leq l} \frac{P_{pre}(w|\hat{x}_{i-n+1}^i) P_{suc}(w|\hat{x}_j^{j+n-1})}{e^{j-i}} \quad (2)$$

This soft constraint on edit length encourages operations that affect fewer prototype components.

B. Neighborhood Approximation with Recursive Editing

Guided by probability-based editing, XPROB builds neighborhoods through recursive text manipulations. The process is initialized with real counterfactuals from X_p that serve as prototypes \hat{x} , since the manipulation formulated by Eq. 2 requires local n -gram contexts and does not generate texts from scratch. By integrating explicand components into counterfactuals, XPROB implements a gradual transition in model predictions from one class to another, resulting in synthetics that reveal the decision boundary situated in between.

To ensure the locality of the generated texts, prototype selection follows their spatial closeness to the explicand. Specifically, the proximity of text pairs is measured by the

Algorithm 1 Recursive probability-based Edit

Input: x : explicand; S : prototype set

Output: $N(x)$: neighborhood of x ;

```
1: while terminate criterion NOT met do
2:    $N(x) = S, S_{new} = \emptyset$ 
3:   for  $w \in x, \hat{x} \in S$  do
4:      $i, j = \text{Edit}(w, \hat{x})$ 
5:      $\hat{x} = (\hat{w}_1, \dots, \hat{w}_i, w, \hat{w}_j, \dots, \hat{w}_l)$ 
6:      $S_{new}.\text{add}(\hat{x})$ 
7:   end for
8:    $S = S_{new} - S_{new} \cap N(x)$ 
9:    $N(x) = N(x) \cup S_{new}$ 
10: end while
11: return  $N(x)$ 
```

cosine distance between their *tf-idf* vectors [27] with a vectorizer fitted on X_p . The distance measure reflects text similarity in terms of components, aligning with the definition of neighboring texts [25]. The texts closest to x form the initial prototype set S . Although there are other text similarity measures, such as the distance of text embeddings [28], *tf-idf* is preferred in this context to avoid introducing any external opacity to the explanation framework.

Taking as input an explicand x and the corresponding prototype set S , the detailed neighborhood construction process is outlined in Algorithm 1. For each word-prototype pair (w, \hat{x}) (line 3), the recursive process attempts probability-based editing to produce a synthetic text \hat{x} (lines 4, 5). Manipulation may be skipped if the prototype already contains w or if the optimal probability obtained by editing does not exceed the validity threshold ϵ^2 . The synthetics produced in one iteration are used to update S for the subsequent round, enabling recursive editing (lines 6, 8). Throughout the iterations, all intermediate texts are collected to shape the final neighborhood $N(x)$ (line 9). The generation process terminates when there are no feasible operations on the updated prototype set or when the neighborhood has collected sufficient instances.

C. Extraction of Local Explanations

Once neighborhood construction is completed, explanatory information is extracted using a linear regression model $g(\cdot)$, trained on $N(x)$ with a weighted square loss:

$$\mathcal{L}(N(x), x) = \sum_{\hat{x} \in N(x)} \exp\left(\frac{-d(\hat{x}, x)^2}{\sigma^2}\right) \cdot (f(\hat{x}) - g(\hat{x}))^2$$

where each synthetic is weighted by its cosine distance to the explicand $d(\hat{x}, x)$, further strengthening locality. The surrogate mimics the behaviors of $f(\cdot)$ as reflected by $N(x)$, serving as a proxy for explanation extraction. Specifically, the coefficients determined by $g(\cdot)$ quantify the attributions to input features. Since the novel neighborhood construction introduces external words not originally part of the explicand, the derived feature attributions can be split into two parts: *intrinsic* and *extrinsic*

attributions. While intrinsic attributions reveal feature contributions to the target decision, influential extrinsics indicate important observations for inference in the target context, thereby facilitating the explanation for a decision [29], [30].

In addition to feature attributions, extracting factual and counterfactual instances from the more realistic neighboring texts forms instance-level explanations ξ^* , which complement word-level explanations ξ . Analyzing the intersections and distinctions between factials and counterfactuals helps to highlight the relevant components of a prediction. Contrary to greedily selecting instances closest to x , which often produces redundant samples, the construction of ξ^* incorporates diversity [31] alongside spatial closeness. Over an iterative selection process, ξ^* sequentially absorbs the best-fitting instance described by:

$$\hat{x} = \arg \max_{\hat{x} \in N(x)} \lambda(1 - d(\hat{x}, x)) + (1 - \lambda) \sum_{\hat{x}' \in \xi^*} \frac{d(\Delta\hat{x}, \Delta\hat{x}')}{|\xi^*|}$$

where $\Delta\hat{x} = \hat{x} - x$ denotes the manipulation direction from the explicand to a generated text. While the first term measures the spatial closeness, the second term quantifies the diversity of a candidate relative to the previously selected instances. Balanced by the parameter $\lambda \in [0, 1]$, the selection process avoids redundancy in ξ^* .

IV. EVALUATION

To evaluate the quality of explanations derived by adopting probability-based editing, this section tests XPROB on two real-world datasets with both qualitative (Section IV-B) and quantitative (Section IV-C) assessments. Subsequently, an experiment designed to verify explanation stability is conducted (Section IV-D). The final part of the experiments explores the impact of external resources on the quality of explanations, which validates concerns about leveraging opaque generators in the context of neighborhood construction (Section IV-E).

A. Experimental setup

Datasets: Two real-world datasets, Yelp reviews [32] and Amazon reviews polarity [33], are utilized for the evaluation of explanations. The *Yelp* dataset comprises restaurant reviews, labeled as either positive or negative. The *Amazon review polarity* dataset includes customer reviews of products with binarized labels. For both datasets, three disjoint subsets are created for classifier training: training, validation, and test sets with sizes of $20k/2k/4k$, respectively. Additionally, each dataset contains $200k$ unlabeled entries, which serve as a corpus reflecting text distribution for the training of generative models used by generator-based explainers. XPROB, which also utilizes an external corpus, downsamples $20k$ texts from the unlabeled set to form X_p for probability distribution estimation and prototype selection.

Text classifiers: In principle, model-agnostic explainers are applicable to classifiers regardless of their underlying architecture. For the sake of evaluation, we test the explainers on two black box models: a BERT model [34] and an eight-layer LSTM network [35], which represent popular architectures for

TABLE I: Details about datasets and black box models

| Dataset | Black box | | | Accuracy ^a | |
|---------|-----------|-------|------|-----------------------|--------|
| | Training | Valid | Test | BERT | LSTM |
| Amazon | 20K | 2K | 4K | 0.8543 | 0.7795 |
| Yelp | 20K | 2K | 4K | 0.9783 | 0.9548 |

^aPerformances are reported on the test sets.

natural language processing. These models are well-suited as black boxes due to their complexity and non-linearity. We fine-tuned a publicly available pretrained version of BERT³ on each dataset. The LSTM features an architecture starting with three dense layers that learn word representations, followed by an LSTM layer capturing the sequential information of texts. The output of the LSTM layer is then processed through another four additional dense layers, including an output layer for the classification tasks concatenated in the end. Table I presents the accuracy of the trained classifiers on both datasets.

Competitors: As this paper focuses on improving the neighborhood construction process and providing an alternative to generator-based solutions, we compare XPROB with one approach that adopts input perturbation and three generator-based explainers:

- **LIME** [5]: a local explanation method constructing neighborhoods by word dropping.
- **XSPELLS** [17]: a text-classifier-specified explainer generating neighbors through random latent space sampling and deriving explanations from a latent decision tree.
- **ABELE** [19]: a generator-based explanation method constructing neighborhoods with a heuristic approach.
- **XPROAX** [9]: an explainer implementing progressive neighborhood approximation through a generative model.

We note that ABELE [19] is designed to explain image classifiers, but it can be readily adapted to deal with textual data since neighborhood construction in the latent space is independent of the original input formats.

Generative model: Following the suggestion by [9], the generator-dependent competitors, namely XSPELLS, ABELE, and XPROAX, adopt a DAAE [36] as their generator to ensure the locality of the generated texts. The DAAE, a symmetric generative autoencoder, is designed to learn a locality-preserving latent space. The hyperparameters for the generator are configured according to the original paper [36]. The autoencoder has a bottleneck of size 128 with both its encoder and decoder implemented by a one-layer bidirectional LSTM with 1024 units.

Hyperparameters for XPROB: Across all test cases, the neighborhood population p and the number of initial prototypes k are set to 400 and 80 for XPROB⁴. Given the relatively shorter review texts, unigram ($n = 1$) is employed for representing local contexts, allowing more feasible edits during neighborhood construction.

³https://huggingface.co/docs/transformers/model_doc/bert

⁴The studies on hyperparameter impacts are not presented here due to the space limitation.

B. Qualitative evaluation

For qualitative evaluation, Table II lists sample explanations from different approaches for the selected decisions. In this table, feature attribution is visualized via saliency maps, where a blue (red) background indicates a contribution to positive (negative, respectively) sentiment. The intensity of the background color corresponds to the magnitude of feature attribution. The visualization is not applied to XSPELLS for a distinguishing purpose. Rather than estimating feature attributions, XSPELLS counts relative word frequencies (numbers in brackets) on the branch of the latent decision tree associated with the explicand. Beyond feature attributions, the explainers, except LIME, deliver additional explanatory information that further reveals model behaviors. This includes the most influential *extrinsic words* as well as the selected factual and counterfactual examples. For the extrinsic words, only the ones with distinctly high attribution scores are reported.

The first instance is the review “*excellent sta for beginning french students.*” from the Amazon dataset, which BERT correctly classifies as *positive*. According to the saliency maps, all competitors agree that “*excellent*” is the most influential word. Similar to the generator-based approaches, XPROB extends the insight into the decision-making process by providing the extrinsic word “*ridiculous*”, illustrating that the classifier learns to leverage sentimental words as evidence for its decisions. Another extrinsic word “*but*”, combined with the second counterfactual given by XPROB, reveals the model’s ability to comprehend contrastive relationships. The emphasized part in this counterfactual, following the term “*but*”, outweighs the positive sentiment and dominates the inference.

The second instance, “*the dessert is very bland.*”, from the Yelp dataset, is assigned a *negative* label by the LSTM model. Similar to the observation in the first example, XSPELLS delivers limited information with generic words and instances. LIME identifies the term “*bland*” as the most contributing feature, but the notably low importance score poorly matches the model’s confidence in its decision. Meanwhile, the last three approaches concur that the term “*bland*” is the primary factor affecting the decision.

Regarding instance-level explanations, the (counter-)factuals listed in the second example underscore the word-level explanations by showing how the presence of “*bland*” drives different model decisions. However, approaches like ABELE and XSPELLS, which rely on naïve latent space sampling, produce neighborhoods that lack either diversity or locality, which can be also observed in both examples. On the other hand, the instances determined by XPROB and XPROAX concentrate more on the target context, effectively complementing the word-level explanations. The synthetic texts enrich the understanding of the underlying decision-making process when comparing factuals with counterfactuals that share similar components. With the meaningful texts revealing explanatory information comparable to those from generator-based solutions, both examples imply that iterative probability-based editing offers a proper alternative to a black-box generator.

TABLE II: Example explanations by different methods

| Input 1 | excellent sta for beginning french students. Dataset: Amazon, BERT $f(\cdot)$: 0.99 | |
|---|---|--|
| LIME | Saliency: excellent sta for beginning french students. | |
| XSPELLS | Factuals: 1) excellent for the $\langle unk \rangle^b$ $\langle unk \rangle$! 2) an essential watch for the world coach students! 3) excellent summer for the field bible! 4) if not for this price , then the french is for you 5) excellent sta for the preface. Most frequent words in factuals: excellent (0.114), best (0.086), students (0.057) | Counterfactuals: 1) chicken for ground deep $\langle unk \rangle$ 2) excellent piece of the sta and mediocre yoga program as well 3) great beats see tv show cannot. 4) an empty direction for christian 5) worst shark move next book ever! Most frequent words in counterfactuals: ground (0.024), sta (0.024), chicken (0.024) |
| ABELE | Saliency: excellent sta for beginning french students. Factuals: 1) excellent sta for beginning french philosophy. 2) excellent sta for beginning french philosophy. 3) excellent sta for beginning french philosophy. 4) excellent sta for beginning french philosophy. 5) excellent sta for beginning french philosophy. | Extrinsic words: please not Counterfactuals: 1) please sta for the french courses. 2) please sta for the french courses. 3) please sta for the french courses. 4) please sta for the french courses. 5) please sta for the french courses. |
| XPROAX | Saliency: excellent sta for beginning french students. Factuals: 1) excellent sta for beginning french philosophy. 2) excellent sta for beginning to travel students. 3) excellent action for beginning of ed. 4) excellent sta for beginning to craft physics. 5) excellent sta for beginning to craft students. | Extrinsic words: disappointing worst Counterfactuals: 1) a disappointing for beginning to $\langle unk \rangle$ philosophy 2) excellent sta bad beginning and cover $\langle unk \rangle$ waste of 3) the sta to the bible on philosophy? 4) not woh the price nothing to deceiving. 5) bad end to the series... |
| XPROB | Saliency: excellent sta for beginning french students. Factuals: 1) sta for students 2) sta for beginning french students. 3) excellent a beginning french students. 4) excellent sta for a french students 5) excellent sta for beginning french | Extrinsic words: ridiculous but Counterfactuals: 1) a ridiculous sta to french students 2) excellent sta for a reference but bad for the beginning french students 3) beginning a ridiculous sta to a french students. 4) very average and for the beginning of the french students 5) excellent sta for good beginning... weak finish |
| ^b The generic token for words out of vocabulary. | | |
| Input 2 | the desserts were very bland. Dataset: Yelp, LSTM $b(\cdot)$: 0.99 | |
| LIME | Saliency: the desserts were very bland . | |
| XSPELLS | Factuals: 1) the carrot was a average. 2) the only real german food. 3) the only offer of the mid $\langle unk \rangle$ houses in the last area. 4) the desserts were very bland. 5) the hashbrowns were bland. Most frequent words in factuals: within (0.067), num (0.067), days (0.067) | Counterfactuals: 1) the baked potatoes was a very sweet and a small thing. 2) great hotel at this. 3) the mashed potato was good. 4) they had the certain menu. 5) the good donuts are the food. Most frequent words in counterfactuals: atrocious (0.083), dvd (0.083), used: (0.083) |
| ABELE | Saliency: the desserts were very bland . Factuals: 1) the desserts were very bland. 2) the desserts were very bland. 3) the desserts were very bland. 4) the desserts were very bland. 5) the desserts were very bland. | Extrinsic words: good Counterfactuals: 1) the desserts were very good. 2) the desserts were very good either. 3) the desserts were very good either. 4) the desserts were very good either. 5) the desserts were very good either. |
| XPROAX | Saliency: the desserts were very bland . Factuals: 1) the ingredients were very bland. 2) the desserts were very very bland. 3) the sausage were very good. 4) the desserts were very clean. 5) the sides were very bland. | Extrinsic words: good Counterfactuals: 1) the desserts were very good. 2) the deserts were also very tasty. 3) the beers were very good. 4) the desserts were pretty good. 5) the rooms were very good. |
| XPROB | Saliency: the desserts were very bland . Factuals: 1) the desserts were very bland. 2) desserts were the very bland. 3) the desserts were bland. 4) desserts were very bland 5) were the very bland: desserts | Extrinsic words: great delicious excellent Counterfactuals: 1) the desserts were very great . 2) the desserts are very delicious too bland. 3) the desserts were very good. 4) great fresh food, friendly service, bland and were the desserts to go 5) the were very excellent desserts. |

C. Quantitative evaluation

While the qualitative assessments provide valuable insights into the performances of the selected explainers, they are not scalable to conclude the performance of an explainer in general. Moreover, qualitative evaluation can be subjective and reliant on the interpretation of results. For greater objectivity, this section is dedicated to the quantitative evaluation of explanation quality. Specifically, the evaluation is structured around the three Cs of explainability [37], [38]:

- *Correctness*: Explanations should faithfully represent the behaviors of the target model $f(\cdot)$;
- *Completeness*: An explanation should encompass all input components that are influential to a model's decision;
- *Compactness*: Explanations should be concise, excluding irrelevant features, to present results efficiently.

To assess *correctness*, we report the fidelity [39] and the R^2 score of a surrogate model, both of which indicate the performance of the surrogate in imitating $f(\cdot)$. An explanation can be faithful only if the surrogate accurately reflects the behaviors of the target model. *Completeness* is evaluated using explanation-guided manipulation [40]. Given an explanation, the process masks out the influential words supporting the original decision while inserting the features opposing it. During the evaluation, influential features are empirically defined with an attribution threshold of 0.1; features whose absolute attribution exceeds this threshold are involved in the manipulation. Explanations that guide manipulations inducing larger confidence drops are favored in terms of completeness. For *compactness*, we measure an explainer's efficiency in presenting results using the AOPC (area over perturbation curve) [41], which is the cumulative sum of confidence drops during manipulation: $\text{AOPC}_x = \frac{1}{l} \sum_{i=1}^l (f(x) - f(x^{(i)}))$, where $x^{(i)}$ represents a variant of x after i manipulation steps, and l is the total number of manipulations. Unlike the evaluation for completeness, manipulations for compactness are applied sequentially following the descending order of relevant features according to their attribution scores. Compact explanations that prioritize attributions to relevant features will yield high AOPC scores, and vice versa.

The results of the quantitative evaluation are summarized in Table III, presenting the average performance on the classifier test set, along with the standard deviations. XSPELLS, utilizing a decision tree as a more capable surrogate model, outperforms other methods in terms of fidelity. The different surrogate choice also explains the absence of its R^2 score. However, the non-linear surrogate built in the latent space obstructs the extraction of explanatory information. As demonstrated by the qualitative examples, estimating feature contributions with word frequencies can overlook the actual relevant features because of the relatively small number of entries on one leaf. This limitation is reflected by XSPELLS's relatively modest performance in completeness and compactness, aligning with observations from the qualitative evaluation.

Among the remaining explainers that employ a linear regressor as their surrogate, XPROB achieves competitive

performance in terms of correctness. While pushing the figures towards a level comparable to those of the generator-based solutions, XPROB surpasses LIME, which adopts the simple input perturbation method, across various settings. The high fidelity implies the quality of the constructed neighborhoods, which stick to the underlying data manifold and emphasize only the relevant local decision boundary, thereby simplifying the imitation task for the surrogate.

In addition to correctness, XPROB excels in the evaluations for completeness and compactness. Without the support of powerful generators, the proposed approach secures the top performance in two of four cases and in general matches the performance of XPROAX, which considerably outperforms the remaining. These results align with the qualitative assessments, suggesting that XPROB effectively identifies the most contributing features and accurately determines their attribution scores. Compared to LIME, XPROB benefits from the neighboring extrinsic words during the manipulation test. These extrinsic words, which oppose the original decision, simplify the reverse of model decisions. However, compared with XSPELLS and ABELE, which also introduce extrinsic words, it becomes evident that the concrete choice of extrinsic words should be made thoughtfully and informed by insights into the local decision context of the explicand. Furthermore, while matching the performance of the best performing generator-based solution, XPROB largely reduces time costs as reported in the last column. The efficiency gain also originates from the substitution of the generative component with editing based on in-text context, highlighting its practical advantages.

D. Stability

In addition to effectiveness, *stability* is another desired property of explanation methods, requiring explainers to output similar explanations for similar explicands [4]. Instead of exhaustively searching corpora for similar texts, we evaluate stability with test cases created by text augmentation [42]. Table IV shows the pre-defined templates for test case creation, which reflect similar contexts commonly found in restaurant reviews. Substituting the placeholders in the templates, denoted by $\langle \text{ADJ} \rangle$ and $\langle \text{NOUN} \rangle$, with adjective-noun pairs drawn from Table V generates one test case containing five similar texts. A stable explainer should then attribute similar scores to an inserted word across the generated examples in one *test case* due to their contextual similarities.

The stability test adopts the *Yelp LSTM* model as the black box. The concrete word choices (listed in Table V) for test case generation are specified for the chosen classifier. These words are selected according to the confidence scores of the predictions when using them as single-word inputs for the black box. For each sentiment class, ten adjectives with the highest confidence scores for each class are selected, whereas the chosen nouns possess confidence scores close to 0.5, indicating neutrality for both classes. Enumerating possible adjective-noun pairs yields a total of 200 test cases.

For each test case, containing five generated texts sharing similar contexts, we measure the consistency of an explainer

TABLE III: Quantitative evaluation following the three Cs

| Dataset & Model | Explainer | Correctness | | Completeness | Compactness | Time cost (s) per entry |
|-----------------|-----------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|----------------------------|
| | | R^2 score | Fidelity | Confidence drop | AOPC | |
| Amazon & BERT | LIME | 0.694 ± 0.197 | 0.867 ± 0.123 | 0.346 ± 0.302 | 0.353 ± 0.246 | 0.160 |
| | XSPELLS | - | 1.000 ± 0.000 | 0.096 ± 0.225 | 0.082 ± 0.189 | 11.852 |
| | ABELE | 0.869 ± 0.063 | 0.942 ± 0.030 | 0.358 ± 0.398 | 0.316 ± 0.365 | 364.359 |
| | XPROAX | 0.878 ± 0.065 | 0.937 ± 0.035 | 0.583 ± 0.369 | 0.484 ± 0.329 | 8.972 |
| | XPROB | 0.828 ± 0.070 | 0.929 ± 0.026 | 0.584 ± 0.345 | 0.482 ± 0.311 | 1.019 |
| Amazon & LSTM | LIME | 0.880 ± 0.127 | 0.896 ± 0.110 | 0.343 ± 0.293 | 0.331 ± 0.223 | 0.023 |
| | XSPELLS | - | 1.000 ± 0.000 | 0.067 ± 0.151 | 0.101 ± 0.163 | 10.674 |
| | ABELE | 0.874 ± 0.059 | 0.936 ± 0.040 | 0.281 ± 0.293 | 0.273 ± 0.249 | 239.322 |
| | XPROAX | 0.912 ± 0.053 | 0.938 ± 0.035 | 0.567 ± 0.285 | 0.437 ± 0.234 | 4.582 |
| | XPROB | 0.906 ± 0.045 | 0.939 ± 0.025 | 0.582 ± 0.245 | 0.477 ± 0.220 | 0.804 |
| Yelp & BERT | LIME | 0.514 ± 0.280 | 0.882 ± 0.136 | 0.366 ± 0.467 | 0.502 ± 0.421 | 0.151 |
| | XSPELLS | - | 1.000 ± 0.000 | 0.231 ± 0.408 | 0.182 ± 0.333 | 11.852 |
| | ABELE | 0.872 ± 0.056 | 0.950 ± 0.227 | 0.407 ± 0.474 | 0.328 ± 0.409 | 369.802 |
| | XPROAX | 0.858 ± 0.064 | 0.970 ± 0.024 | 0.770 ± 0.401 | 0.629 ± 0.366 | 8.871 |
| | XPROB | 0.777 ± 0.071 | 0.943 ± 0.022 | 0.760 ± 0.405 | 0.609 ± 0.372 | 1.016 |
| Yelp & LSTM | LIME | 0.757 ± 0.222 | 0.954 ± 0.101 | 0.506 ± 0.424 | 0.594 ± 0.293 | 0.023 |
| | XSPELLS | - | 1.000 ± 0.000 | 0.288 ± 0.397 | 0.225 ± 0.299 | 10.829 |
| | ABELE | 0.867 ± 0.063 | 0.941 ± 0.039 | 0.448 ± 0.434 | 0.334 ± 0.346 | 233.105 |
| | XPROAX | 0.864 ± 0.071 | 0.958 ± 0.041 | 0.827 ± 0.299 | 0.623 ± 0.278 | 4.508 |
| | XPROB | 0.833 ± 0.054 | 0.945 ± 0.021 | 0.822 ± 0.296 | 0.658 ± 0.288 | 0.815 |

TABLE IV: Templates for creating similar texts

| | |
|------------------|--|
| Templates | 1. $\langle ADJ \rangle \langle NOUN \rangle$ |
| | 2. Very $\langle ADJ \rangle \langle NOUN \rangle$ |
| | 3. The $\langle NOUN \rangle$ is $\langle ADJ \rangle$ |
| | 4. A very $\langle ADJ \rangle \langle NOUN \rangle$ |
| | 5. This is a very $\langle ADJ \rangle \langle NOUN \rangle$ |

TABLE V: Choices of adjective and noun

| | |
|----------------------------|---|
| Negative adjectives | horrible terrible wrong awful disappointed poor bland worst bad cheap |
| Positive adjectives | delicious amazing excellent loved fantastic wonderful perfect fresh great best |
| Nouns | bread soup pizza food meal salad drink dessert fish steak |

by computing the deviation in attributions to each inserted word. A lower deviation across similar contexts signifies greater stability in explanation outcomes. To further quantify the overall stability of an explainer, we average the attribution deviations for adjectives and nouns respectively over all test cases, yielding the *averaged case deviation* $\bar{\sigma}$. For example, the averaged case deviation for adjectives is computed by: $\bar{\sigma}_{adj.} = \frac{1}{\#Cases} \sum_{w \in adj.} \sigma_w$. Analogously, the averaged case mean $\bar{\mu}$ is computed for each word category to assess whether an explainer uncovers the influence of adjectives and the neutrality of nouns in sentiment analysis.

Table VI presents the results of the stability evaluation, excluding XSPELLS due to its inconsistent frequency counts for intrinsic features. The figure in the first column represents the averaged case deviation of model outcomes. The small deviation indicates trivial prediction variation within each test case, thus strengthening the foundation of the stabil-

TABLE VI: Stability evaluation reported on Yelp LSTM

| $\bar{\sigma}_f(\cdot)$ | Explainer | Adjective | | Noun | |
|-------------------------|-----------|--------------------|-----------------------|----------------|-------------------|
| | | $\bar{\mu}_{adj.}$ | $\bar{\sigma}_{adj.}$ | $\bar{\mu}_n.$ | $\bar{\sigma}_n.$ |
| 0.005 | LIME | 0.441 | 0.109 | 0.043 | 0.062 |
| | ABELE | 0.139 | 0.062 | 0.064 | 0.054 |
| | XPROAX | 0.441 | 0.148 | 0.035 | 0.055 |
| | XPROB | 0.551 | 0.082 | 0.019 | 0.019 |

ity evaluation. The results, particularly the averaged attribution deviations in the fourth and sixth columns, showcase XPROB's excellence. It demonstrates low deviations for both word categories, indicating that a word receives consistent attribution scores across similar contexts. The consistency in explanation results is credited to its controlled generation process driven by probability-based editing. Moreover, while all competitors agree that the adjectives are more influential than the neutral nouns by showing $\bar{\mu}_{adj.} \gg \bar{\mu}_{noun}$, XPROB constantly assigns approximately zero attributions to the nouns, accurately reflecting their non-informative nature in sentiment analysis. Though ABLE also appears stable in attributions to the adjectives, quantified by $\bar{\sigma}_{adj.}$, we note that a nonnegligible factor in its low deviation is the considerably smaller scale of attribution scores as shown by $\bar{\mu}_{adj.}$.

E. Dependency on external resources

The main argument of the paper is that the utilization of opaque generators can induce mistrust in derived explanations. As the last part of the experiments, this section investigates the dependencies of generator-based approaches on these opaque components. Due to the space limitation, the dependency analysis is conducted for XPROAX, the best-performing generator-based explainer according to the quantitative evaluation. The analysis evaluates the explanation quality of the explainer under various generator configurations. Specifically, we test the performance of XPROAX with two generative models,

TABLE VII: Dependency of XPROAX on generative models

| Generator | Bottleneck | Amazon LSTM | |
|-----------|------------|-------------------------------------|-------------------------------------|
| | | Confidence drop | AOPC |
| DAAE | 32 | 0.464 \pm 0.341 | 0.373 \pm 0.281 |
| | 64 | 0.528 \pm 0.302 | 0.411 \pm 0.249 |
| | 96 | 0.557 \pm 0.280 | 0.435 \pm 0.236 |
| | 128 | 0.567 \pm 0.285 | 0.437 \pm 0.234 |
| VAE | 32 | 0.329 \pm 0.364 | 0.289 \pm 0.303 |
| | 64 | 0.268 \pm 0.357 | 0.235 \pm 0.302 |
| | 96 | 0.296 \pm 0.358 | 0.264 \pm 0.304 |
| | 128 | 0.332 \pm 0.352 | 0.299 \pm 0.294 |

TABLE VIII: A failed case by XPROAX with VAE

| | | | |
|--|--|--|--|
| Input: stay the hell away from this book | | Amazon, LSTM $b(\cdot)$: 0.86 | |
| stay the hell away from this book | | XPROB ($X_p = 5k$) | |
| Factuals: 1) stay the hell away for this book 2) from the hell 3) stay away the book ! | | Counterfactuals: 1) hell is a great book 2) this from the book away ! 3) stay away from this book | |
| stay the hell away from this book | | XPROAX (DAAE-128) | |
| Factuals: 1) stay the night it with this book! 2) stay the hell away for this book 3) stay the critics away in this book | | Counterfactuals: 1) the far side away from this book 2) enjoy the movie away from this book 3) why the hell do schools this book | |
| stay the hell away from this book | | XPROAX (VAE-128) | |
| Factuals: 1) where s roy with the rest? 2) where is jamie lee s a? 3) where s to the point? | | Counterfactuals: 1) where s roy with my mind 2) why is so many of these songs? 3) where s the new wave | |

DAAE and VAE [18], and alter the generator capacity by adjusting the bottleneck size of each autoencoder. To ensure a valid comparison, the network structures of the two generators are kept identical.

Table VII details confidence drops and AOPC scores of XPROAX when coupled with generators of different configurations on the Amazon LSTM model. When adopting the DAAE, the explanation quality, in terms of both completeness and compactness, positively correlates to the generator’s capacity constrained by the bottleneck size. In contrast, such a correlation is absent with the VAE. Despite having an identical network structure to the DAAE and differing only in training objectives, the utilization of the VAE results in a significant falloff in explainer performance regardless of the bottleneck size. Table VIII showcases the impact of the concrete generator choices on explanation quality through a concrete example. The attributions derived from the neighborhood constructed via the VAE fail to accurately illustrate the true contributing features, and the instance-level explanation significantly deviates from the target context. The observed discrepancies suggest that inadequate latent space geometry [43], lacking the locality-preserving property, is a potential cause of the declined performance. However, the inherent opacity of generators based on neural networks complicates the prevention and resolution of the issue, underscoring the concerns about adopting opaque components in explanatory frameworks.

As a comparison, the dependency of XPROB on its external resource X_p is presented in Table IX. Its evaluation follows the previous scheme and reports performance under various

TABLE IX: Dependency of XPROB on the prototype corpus

| Corpus size | Amazon LSTM | |
|-------------|-------------------------------------|-------------------------------------|
| | Confidence drop | AOPC |
| 2k | 0.478 \pm 0.287 | 0.385 \pm 0.241 |
| 5k | 0.543 \pm 0.270 | 0.442 \pm 0.235 |
| 10k | 0.557 \pm 0.255 | 0.462 \pm 0.227 |
| 20k | 0.582 \pm 0.245 | 0.477 \pm 0.220 |
| 40k | 0.584 \pm 0.245 | 0.479 \pm 0.217 |
| 80k | 0.593 \pm 0.239 | 0.489 \pm 0.211 |

settings of the external corpus, which serves for probability distribution estimation and prototype selection. As expected, a larger corpus better represents the underlying probability distribution, thus promoting the quality of text generated by probability-based editing. This results in a positive correlation between the explanation quality and the size of X_p . However, in contrast to the generator-based solution, XPROB exhibits a direct but less sensitive dependency on the volume of the corpus due to its more controllable text generation process. While the generator in this experiment requires 200k entries for training, the proposed approach still maintains competitive performance with a much smaller corpus of size 5k.

V. CONCLUSIONS

In this work, we propose XPROB, a model-agnostic attribution method that improves the neighborhood construction process with probability-based editing. While following the concept of likelihood maximization for generated texts, the editing approach distinguishes itself from opaque generators with transparent text manipulation on prototypes guided by local n -gram contexts. In addition to the desired transparency of the explanation process, the proposed explainer achieves competitive performance in both qualitative and quantitative evaluations, even compared to the best performing competitor supported by a capable generative autoencoder.

In future work, we plan to refine the recursive editing process of XPROB. Currently, the process greedily selects the optimal operation for each word-prototype pair, without considering the impacts on succeeding manipulations. The greedy strategy can be unsatisfactory during sequential manipulations. A potential solution to the problem is a regular restructuring of edited prototypes, thereby promoting the consistency of edits. Additionally, the setting of the explanation task can be generalized to broaden its applicability. While this paper focuses on binary classification tasks with short texts for simplicity, the proposed explainer can be conveniently adapted to multi-class scenarios. This adaptation only involves redefining the counterfactual as an instance with a low prediction confidence score for the class predicted for the explicand. As for handling longer inputs, probability-based editing can split a long text into shorter segments and manipulate each segment individually, as done for short texts.

REFERENCES

- [1] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. S. Iyengar, “A survey on deep learning: Algorithms, techniques, and applications,” *ACM Computing Surveys (CSUR)*, vol. 51, no. 5, pp. 1–36, 2018.

- [2] L. Bernstein, A. Sludds, R. Hamerly, V. Sze, J. Emer, and D. Englund, "Freely scalable and reconfigurable optical hardware for deep learning," *Scientific Reports*, vol. 11, no. 1, pp. 1–12, 2021.
- [3] B. Goodman and S. Flaxman, "European union regulations on algorithmic decision-making and a "right to explanation"," *AI Magazine*, vol. 38, no. 3, pp. 50–57, 2017.
- [4] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, p. 832, 2019.
- [5] M. T. Ribeiro, S. Singh, and C. Guestrin, "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*, 2016, pp. 1135–1144.
- [6] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in Neural Information Processing Systems*, vol. 30, pp. 4765–4774, 2017.
- [7] T. Laugel, X. Renard, M.-J. Lesot, C. Marsala, and M. Detryniecki, "Defining locality for surrogates in post-hoc interpretability," in *Workshop on Human Interpretability in Machine Learning (WHI)-International Conference on Machine Learning (ICML)*, 2018.
- [8] V. Petsiuk, A. Das, and K. Saenko, "RISE: Randomized input sampling for explanation of black-box models," in *British Machine Vision Conference*, 2018.
- [9] Y. Cai, A. Zimek, and E. Ntouts, "XPROAX-local explanations for text classification with progressive neighborhood approximation," in *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2021, pp. 1–10.
- [10] O. Lampridis, L. State, R. Guidotti, and S. Ruggieri, "Explaining short text classification with diverse synthetic exemplars and counter-exemplars," *Machine Learning*, vol. 112, no. 11, pp. 4289–4322, 2023.
- [11] E. Štrumbelj, I. Kononenko, and M. R. Šikonja, "Explaining instance classifications with interactions of subsets of feature values," *Data & Knowledge Engineering*, vol. 68, no. 10, pp. 886–904, 2009.
- [12] H. Chen, I. C. Covert, S. M. Lundberg, and S.-I. Lee, "Algorithms to estimate shapley value feature attributions," *Nature Machine Intelligence*, vol. 5, no. 6, pp. 590–601, 2023.
- [13] X. Jin, Z. Wei, J. Du, X. Xue, and X. Ren, "Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models," in *International Conference on Learning Representations*, 2019.
- [14] H. Chen, J. D. Janizek, S. Lundberg, and S.-I. Lee, "True to the model or true to the data?" in *Workshop on Human Interpretability in Machine Learning (WHI)-International Conference on Machine Learning (ICML)*, 2020.
- [15] S. Ghalebikesabi, L. Ter-Minassian, K. DiazOrdaz, and C. C. Holmes, "On locality of local explanation models," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18 395–18 407, 2021.
- [16] C. Frye, D. de Mijolla, T. Begley, L. Cowton, M. Stanley, and I. Feige, "Shapley explainability on the data manifold," in *International Conference on Learning Representations*, 2020.
- [17] O. Lampridis, R. Guidotti, and S. Ruggieri, "Explaining sentiment classification with synthetic exemplars and counter-exemplars," in *International Conference on Discovery Science*. Springer, 2020, pp. 357–373.
- [18] S. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 2016, pp. 10–21.
- [19] R. Guidotti, A. Monreale, S. Matwin, and D. Pedreschi, "Black box explanation by learning image exemplars in the latent feature space," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2019, pp. 189–205.
- [20] S. M. Shankaranarayana and D. Runje, "ALIME: Autoencoder based approach for local interpretability," in *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 2019, pp. 454–463.
- [21] A. Feder, N. Oved, U. Shalit, and R. Reichart, "CausaLM: Causal model explanation through counterfactual language models," *Computational Linguistics*, vol. 47, no. 2, pp. 333–386, 2021.
- [22] P. Rodriguez, M. Caccia, A. Lacoste, L. Zamparo, I. Laradji, L. Charlin, and D. Vazquez, "Beyond trivial counterfactual explanations with diverse valuable explanations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1056–1065.
- [23] L. Longo, M. Brcic, F. Cabitza, J. Choi, R. Confalonieri, J. Del Ser, R. Guidotti, Y. Hayashi, F. Herrera, A. Holzinger *et al.*, "Explainable artificial intelligence (XAI) 2.0: A manifesto of open challenges and interdisciplinary research directions," *Information Fusion*, p. 102301, 2024.
- [24] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.
- [25] D. Alvarez-Melis and T. Jaakkola, "A causal framework for explaining the predictions of black-box sequence-to-sequence models," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 412–421.
- [26] D. Jurafsky, *Speech & language processing*. Pearson Education India, 2000.
- [27] C. D. Manning, *Introduction to information retrieval*. Syngress Publishing, 2008.
- [28] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3982–3992.
- [29] J. Ma, J. Li, Z. Li, and J. Jiao, "Salient region detection by integrating intrinsic and extrinsic cues without prior information," *Journal of Engineering Science & Technology Review*, vol. 10, no. 3, 2017.
- [30] M. Wang, J. Konrad, P. Ishwar, K. Jing, and H. Rowley, "Image saliency: From intrinsic to extrinsic context," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 417–424.
- [31] Z. Gong, P. Zhong, and W. Hu, "Diversity in machine learning," *IEEE Access*, vol. 7, pp. 64 323–64 350, 2019.
- [32] T. Shen, T. Lei, R. Barzilay, and T. Jaakkola, "Style transfer from non-parallel text by cross-alignment," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6833–6844.
- [33] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: Understanding rating dimensions with review text," in *Proceedings of the 7th ACM Conference on Recommender systems*, 2013, pp. 165–172.
- [34] J. D. M.-W. C. Kenton and L. K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.
- [35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [36] T. Shen, J. Mueller, R. Barzilay, and T. Jaakkola, "Educating text autoencoders: Latent representation guidance via denoising," in *International Conference on Machine Learning*. PMLR, 2020, pp. 8719–8729.
- [37] W. Silva, K. Fernandes, M. J. Cardoso, and J. S. Cardoso, "Towards complementary explanations using deep neural networks," in *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*. Springer, 2018, pp. 133–140.
- [38] M. Nauta, J. Trienes, S. Pathak, E. Nguyen, M. Peters, Y. Schmitt, J. Schlöterer, M. van Keulen, and C. Seifert, "From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai," *ACM Computing Surveys*, vol. 55, no. 13s, pp. 1–42, 2023.
- [39] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Computing Surveys*, vol. 51, no. 5, pp. 1–42, 2018.
- [40] D. Nguyen, "Comparing automatic and human evaluation of local explanations for text classification," in *16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2018, pp. 1069–1078.
- [41] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller, "Evaluating the visualization of what a deep neural network has learned," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 11, pp. 2660–2673, 2016.
- [42] R. Rudinger, J. Naradowsky, B. Leonard, and B. Van Durme, "Gender bias in coreference resolution," in *Proceedings of NAACL-HLT*, 2018, pp. 8–14.
- [43] J. Zhao, Y. Kim, K. Zhang, A. Rush, and Y. LeCun, "Adversarially regularized autoencoders," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5902–5911.