Understanding Post-hoc Explainers: The Case of Anchors

Gianluigi Lopardo, Frédéric Precioso, Damien Garreau

Understanding Post-hoc Explainers: The Case of Anchors

Gianluigi Lopardo [1], Frédéric Precioso [2], Damien Garreau [1]

[1] *Université Côte d'Azur, Inria, CNRS, LJAD, France*
[2] *Université Côte d'Azur, Inria, CNRS, I3S, France*
{*glopardo, dgarreau, fprecioso*}*@unice.fr*

Résumé. Dans de nombreux scénarios, l'interprétabilité de modèles d'apprentissage automatique est une tâche hautement nécessaire mais difficile. Pour expliquer les prédictions individuelles de ces modèles, des approches locales et modèles-agnostiques ont été proposées. Cependant, le processus de génération des explications peut être, pour un utilisateur, aussi mystérieux que la prédiction à expliquer. De plus, les méthodes d'interprétabilité manquent souvent de garanties théoriques et leur comportement sur des modèles simples est souvent inconnu. S'il est difficile, voire impossible, de garantir qu'un explicateur se comporte comme prévu sur un modèle de pointe, nous pouvons au moins nous assurer que tout fonctionne sur des modèles simples et déjà interprétables. Dans cet article, nous présentons un'analyse théorique de Anchors (Ribeiro et al., 2018) : une méthode populaire d'interprétabilité basée sur des règles qui met en évidence un petit ensemble de mots pour expliquer la décision d'un classificateur de texte. Après avoir formalisé son algorithme et fourni des informations utiles, nous démontrons mathématiquement que Anchors produisent des résultats significatifs lorsqu'elles sont utilisées avec des classificateurs de texte linéaires en plus d'une vectorisation TF-IDF. Nous pensons que notre cadre d'analyse peut contribuer au développement de nouvelles méthodes d'explicabilité basées sur des fondements théoriques solides.

Mots-clés. Apprentissage statistique, Classification supervisée et non supervisée, Interprétabilité, Traitement du Langage Naturel.

Abstract. In many scenarios, the interpretability of machine learning models is a highly required but difficult task. To explain the individual predictions of such models, local model-agnostic approaches have been proposed. However, the process generating the explanations can be, for a user, as mysterious as the prediction to be explained. Furthermore, interpretability methods frequently lack theoretical guarantees, and their behavior on simple models is frequently unknown. While it is difficult, if not impossible, to ensure that an explainer behaves as expected on a cutting-edge model, we can at least ensure that everything works on simple, already interpretable models. In this paper, we present a theoretical analysis of Anchors (Ribeiro et al., 2018): a popular rule-based interpretability method that highlights a small set of words to explain a text classifier's decision. After formalizing its algorithm and providing useful insights, we demonstrate mathematically that Anchors produces meaningful results when used with linear text classifiers on top of a TF-IDF vectorization. We believe that our analysis framework can aid in the development of new explainability methods based on solid theoretical foundations.

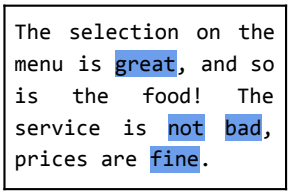
Keywords. Statistical learning, Supervised and unsupervised classification, Interpretability, Natural Language Processing.

1 Introduction

Complex machine learning models with billions of parameters, such BERT and GPT-3 (Devlin et al., 2019; Brown et al., 2020), have become increasingly popular in natural language processing. The interpretability of these models, however, continues to be a problem in sensitive or important scenarios when consumers and subject-matter experts demand explanations. Many solutions, including local model-agnostic approaches that explicate individual predictions for a particular instance, have been proposed for creating interpretable explanations to overcome this issue. These methods, however, may lack theoretical guarantees, and their behavior on simple, interpretable models is frequently unknown, potentially leading to misleading results.

In this work, we focus on Anchors (Ribeiro et al., 2018), an increasingly popular local model-agnostic, and we particularly deal with its implementation for text data. In this context, Anchors outputs a list of words that, if present, produce the same prediction with a high probability, and are presented to the user as such (see Figure 1). We examine whether Anchors, especially when the model being explained is already interpretable, can pinpoint the most crucial words for the prediction. In this paper, we present the framework for analyzing Anchors on linear text classifiers proposed in Lopardo et al. (2023). Our analysis lays the groundwork for future research in the theoretical foundations of interpretability and offers insightful information about the behavior of Anchors for text data.

Paper organization. The structure of the paper is as follows. We begin by introducing some related works on interpretability and its theoretical foundations in the paragraph after. Second, we formalize Anchors' text classification mechanism in Section 2 and explain its fundamental concepts. We then go over the definition of a more tractable, exhaustive version of the algorithm in Section 3, which is the main focus of our research. Next, in Section 4, we illustrate a theoretical and empirical analysis of Anchors' behavior on linear classifiers to better understand their efficacy for text data. Finally, we summarize the results of our study in Section 5 and reach our conclusions.



precision : 0.97
coverage : 0.12

Figure 1: Anchors explaining the positive prediction of a black-box model f on an example ξ from the Restaurant review dataset. The anchor $A = \{\textit{great, not, bad, fine}\}$ (in blue), having length $|A| = 4$ is selected. Intuitively, that a document contains these four words together ensures a positive prediction by f with high probability (precision = 0.97), while being not too uncommon (coverage = 0.12).

Related work. In recent years, several methods have been proposed for machine learning interpretability (Guidotti et al., 2018; Adadi and Berrada, 2018; Linardatos et al., 2021), and among them, rule-based methods have gained popularity. This is because users prefer rule-based explanations (Lim et al., 2009; Stumpf et al., 2007), and hierarchical decision lists (Wang and Rudin, 2015) can be useful for understanding the global behavior of a model. However, smaller and disjoint rules are easier to interpret, and Lakkaraju et al. (2016) introduces the concept of *coverage* to extract small and disjoint rules while compromising between accuracy and interpretability. Alternatively, Barbiero et al. (2022) proposes to learn simple logical rules along with the parameters of the model itself, so as not to sacrifice accuracy.

Several approaches have also focused on *local interpretability*, where, typically, a simpler model is used to approximate any black-box model around a specific instance to explain. For example, LORE (Guidotti et al., 2018) uses a decision tree as a local surrogate, while LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017) provide explanations using a linear model trained on perturbed samples of the instance to explain. Amoukou and Brunel (2022) proposed Minimal Sufficient Rules, similar to Anchors for tabular data, extended to regression models, which can directly deal with continuous features with no need for discretization. While LIME and SHAP assign a weight to each word of the example, Anchors extracts the minimal subset of words that is sufficient to have the same prediction as the example in high probability.

Despite the popularity of interpretability methods, few works have investigated their theoretical guarantees. For feature importance methods, Lundberg and Lee (2017) provides insights into the case of linear models for kernel SHAP, while Mardaoui and Garreau (2021) extended Garreau and Luxburg (2020) and investigated LIME for text data. In this paper, we exhibit an analysis of Anchors on linear text classifiers, first presented in Lopardo et al. (2023). Having theoretical guarantees makes it possible not only to act as a sanity check for a method, but also to compare different explainers with each other, which is otherwise only possible experimentally (Lopardo and Garreau, 2022).

2 Anchors for text data

This section describes Anchors for text data, as introduced by Ribeiro et al. (2018). We start by defining the setting and notation, followed by presenting the key concepts of precision and coverage. Finally, we explain the algorithm and discuss the sampling scheme.

Setting and notation. The problem at hand is to explain the decision of a binary classifier f that takes documents as input. We denote a generic document by z and the particular example being explained by Anchors as ξ. We define a global dictionary $\mathcal{D}$ of cardinality D and represent any document as a finite sequence of elements from $\mathcal{D}$. We also define a local dictionary $\mathcal{D}_\xi$ for a given document ξ as the set of distinct words in ξ.

We make two restrictive assumptions about the class of models they consider. First, we restrict the analysis to binary classification. Second, we assume that the classifier g relies on a vectorization of the documents, where $g = h \circ \varphi$, and φ is a deterministic mapping from

texts to $\mathbb{R}^D$, and $h : \mathbb{R}^D \to \mathbb{R}^p$ is a given measurable function.

We define an anchor as any non-empty subset of $[b]$, corresponding to a preserved set of words of ξ, and denote the set of all candidate anchors for ξ as $\mathcal{A}$. For any anchor $A \in \mathcal{A}$, they set $|A|$ as the length of the anchor, defined as the number of words contained in A. In practice, an anchor A for a document ξ is represented as a non-empty sublist of the words present in the document.

Precision and coverage. The precision of an anchor A is defined as the probability for a local perturbation of ξ to be classified as 1, and it can be written as $\text{Prec}(A) = \mathbb{P}_A\left(g(\varphi(x)) = 1\right)$, where x is a random perturbation of ξ still containing all the words included in A. The coverage measures the proportion of examples in a corpus that satisfy a given anchor, *i.e.* $\text{Cov}(A) = \mathbb{P}_{f(x)=1}\left(A\right)$, where the expectation is taken with respect to x, a random perturbation of ξ still containing all the words included in A. Anchors algorithm prioritizes the precision and coverage: in a nutshell, **Anchors will search for an anchor of maximum coverage with prescribed precision.** In the section after, we go into more detail.

Algorithm. In practice, computing the coverage can be expensive and may not be feasible if a corpus is unavailable during prediction. To address this issue, the default implementation minimizes the length of anchors instead of maximizing the coverage, as shorter anchors tend to have larger coverage. The optimization problem in this case is to find an anchor with minimal length $A \in \mathcal{A}$ that satisfy $\text{Prec}(A) \geq 1 - \varepsilon$, where ε is a tolerance threshold (set to 0.05 in practice). The precision of a specific anchor cannot be computed exactly because the expectation in the precision formula cannot be calculated in general. Instead, the precision is approximated using $\widehat{\text{Prec}}_n(A)$, an empirical approximation defined in Section 3. Since the cardinality of $\mathcal{A}$ is too large in practical scenarios, the default implementation applies the KL-LUCB algorithm to identify a subset of high-precision rules that serve as representatives of all candidate anchors to approximate the solution to the optimization problem. However, this paper focuses on an exhaustive version of Anchors described in Section 3, and does not consider the optimization procedure.

Sampling. The objective is to observe the behavior of the model f in a local neighborhood of ξ, while holding the set of words in A fixed. To achieve this, Anchors generates perturbed samples $x_1, \ldots, x_n$, for a given example ξ and a candidate anchor $A \in \mathcal{A}$. This process involves creating n identical copies of the example, drawing a number of copies to be perturbed for each word not in the anchor, and then replacing the selected words belonging to those copies with the token *UNK*. Note that this is done in practice in the official implementation, but it is different from the perturbation distribution used in Ribeiro et al. (2018), where selected words are replaced with others having the same part-of-speech tag with probability proportional to their similarity in an embedding space.

Replacing words with a predefined token can generate meaningless sentences that can deceive a classifier and produce unrealistic samples (Hase et al., 2021). Nevertheless, we consider the *UNK*-replacement in this work because it is the default choice used by Anchors'

users and replicates word removals exactly in the case of TF-IDF vectorization. Additionally, our experiments show that the results remain valid when words are replaced using *BERT*, which is an alternative solution propose din Anchors' package. Anchors' sampling process is similar to LIME for text data (Mardaoui and Garreau, 2021), except LIME removes all occurrences of a given word when it is selected for removal. The sampling process can be simplified as follows: for any sample x_i, each word $x_{i,k}$ such that $k \notin A$ is replaced independently with probability 1/2. Lopardo et al. (2023) shows that for any given anchor A, the random variable M_j, which is defined as the multiplicity of word w_j in the perturbed sample x, can be described by $M_j \sim a_j + \text{Bin}(m_j - a_j, 1/2)$, where a_j is the number of occurrences of w_j in A.

3 Exhaustive p-Anchors

In this section, we introduce exhaustive p-Anchors as the central object of our study. This is a formalized version of the original combinatorial optimization problem presented in Section 2, which can be solved for any evaluation function $p : \mathcal{A} \to \mathbb{R}$.

Description of the algorithm. The optimization problem of Anchors can be divided into two steps. Firstly, we select all anchors in $\mathcal{A}$ such that $\text{Prec}(A) \geq 1 - \varepsilon$ to obtain a subset of anchors $\mathcal{A}_1(\varepsilon)$. The full anchor $[b]$ has precision 1 and is always included in this set. Secondly, we select the anchors in $\mathcal{A}_1(\varepsilon)$ that have minimal length to obtain a subset of anchors $\mathcal{A}_2(\varepsilon)$. Finally, we select the anchor(s) with the highest precision from $\mathcal{A}_2(\varepsilon)$, which we call $\mathcal{A}_3(\varepsilon)$. If $\mathcal{A}_3(\varepsilon)$ contains more than one anchor, we randomly select one.

We have designed this algorithm to be flexible so that it can be used with different evaluation functions. For example, we can use the algorithm with $p = \widehat{\text{Prec}}_n$ or $p = \text{Prec}$ as a selection function, or any other function that approximates Prec well. When we use $p = \underline{\text{Prec}}$, we refer to this version of the algorithm as *exhaustive Anchors*. When we use $p = \widehat{\underline{\text{Prec}}}_n$, we refer to this version as *empirical Anchors*.

Empirical Anchors is very similar to Anchors, but instead of using an efficient approximate procedure, it considers all possible anchors. The main difference is that empirical Anchors selects anchors with the maximal precision in the third step, while this is not necessarily the case with the default implementation. We refer to Lopardo et al. (2023) for a more detailed description, where *empirical Anchors* and *exhaustive Anchors* are mathematically and empirically proved to be close.

4 Analysis on Linear Classifiers

In this Section we detail the framework used in Lopardo et al. (2023) to analyze Anchors on linear classifiers. First, we introduce the vectorizer that we are considering, and then delve into the analysis of Anchors' behavior on linear models, presenting our key findings. The

interested reader can find mathematical proofs and empirical validations of all our claims in the Appendix of Lopardo et al. (2023).

Vectorizers. Natural language processing classifiers rely heavily on a vector representation φ of documents, which is often obtained using the popular TF-IDF (Term Frequency-Inverse Document Frequency) transform (Luhn, 1957). This method assigns greater weight to words that appear frequently in a particular document z, but less frequently in the overall corpus $\mathcal{C}$. In this paper, we assume that models work with a non-normalized TF-IDF vectorizer, which is defined by a vector $\varphi(z)$ that is based on the inverse document frequency (IDF) of each word in the vocabulary. Once the TF-IDF vectorizer is fitted to a corpus, the vocabulary is fixed, and any word not present in the initial corpus is assigned an IDF term of zero.

Note that with this vectorizer, replacing any word with a fixed token *UNK* is equivalent to simply removing it, as the token is not present in the initial corpus. We also remark that when models are trained on a (non-normalized) TF-IDF vectorization, the exact location of the words in the document does not matter, and thus only the occurrences of each word in an anchor A are important when computing precision. We represent an anchor $A = (a_1, \ldots, a_d)$, where $a_j \leq m_j$ for all $j \in [D]$ and $a_j = 0$ for any $j > d$. Finally, note that the TF-IDF of w_j in the perturbed sample can be expressed as $(a_j + \mathrm{Bin}(m_j - a_j, 1/2))v_j$, which intuitively corresponds to the number of occurrences of w_j in the anchor plus a random number of occurrences depending on the sampling.

Linear classifiers. We now focus on linear classifiers in this section. For any document z, we define the linear classifier $f(z)$ as follows:

$$f(z) = \mathbf{1}_{\lambda^\top \varphi(z) + \lambda_0 > 0}\,, \tag{1}$$

where $\lambda \in \mathbb{R}^D$ is a vector of coefficients, and $\lambda_0 \in \mathbb{R}$ is an intercept. Equation (1) includes several examples, two of which are the perceptron (Rosenblatt, 1958), which predicts exactly as in Equation (1), and logistic models that predict as 1 if $\sigma(\lambda^\top \varphi(z) + \lambda_0) > 1/2$. Here, $\sigma : \mathbb{R} \to \sigma(t) = \frac{1}{1+\mathrm{e}^{-t}} \in [0, 1]$ is the logistic function. As $\sigma(y) > 1/2$ if, and only if, $y > 0$, logistic models can also be rewritten as in Equation (1). For a more complete overview of linear classifiers, refer to Chapter 4 in Hastie et al. (2009). We investigate the precision of a linear classifier using a Berry-Esseen-type statement (Berry, 1941; Esseen, 1942).

Proposition 1 (Precision of a linear classifier) *Let λ, λ_0 be the coefficients associated to the linear classifier defined by Eq. (1). Assume that for all $j \in [d]$, $\lambda_j v_j \neq 0$. Define, for all $A \in \mathcal{A}$,*

$$L\left(A\right) := \frac{-\lambda_0 - \frac{1}{2}\sum_{j=1}^{d} \lambda_j v_j (m_j + a_j)}{\sqrt{\frac{1}{4}\sum_{j=1}^{d} \lambda_j^2 v_j^2 (m_j - a_j)}}\,. \tag{2}$$

Let $\overline{\Phi} := 1 - \Phi$, where Φ denotes the cumulative distribution function of a $\mathcal{N}(0, 1)$. Then,

for any $A \in \mathcal{A}$ *such that* $|A| \leq b/2$,

$$\left|\text{Prec}(A) - \overline{\Phi}\left(L\left(A\right)\right)\right| \leq \tag{3}$$
$$C \cdot \left(\frac{\max_j \lambda_j^2 v_j^2}{\min_j \lambda_j^2 v_j^2}\right)^{3/2} \cdot \left(\frac{\max_j m_j}{\min_j m_j}\right)^{3/2} \cdot \frac{1}{\sqrt{d}},$$

where $C \approx 7.15$ *is a numerical constant.*

In practice, the precision of the anchor can be well approximated by $\overline{\Phi} \circ L$, where, $\overline{\Phi} := 1 - \Phi$, and Φ denotes the cumulative distribution function of a $\mathcal{N}(0, 1)$. We can use this approach to study exhaustive p-Anchors with $p = \overline{\Phi} \circ L$ instead of exhaustive Anchors. Proposition 1 is proven in Lopardo et al. (2023), where it is shown that in the case of normalized TF-IDF, a constant with the same rate appears.

A typical value for v_j and m_j in our setting lies between 1 and 10. Thus, the main assumption is the absence of zero components in λ. The fact that an explanation based on more than half the document is factually uninterpretable justifies the assumption regarding the length of anchors (less than half the length of the document). The constant would also be improved by assuming even smaller anchors.

We can now focus on exhaustive $\overline{\Phi} \circ L$-Anchors. Let us set $\gamma := \lambda_0 + \sum_j \lambda_j v_j m_j$. Note that, since we assume $f(\xi) = 1$, $\gamma > 0$. Let us also set

$$\mathcal{A}_+ := \{A \in \mathcal{A}, \text{ s.t. } a_j > 0 \Rightarrow \lambda_j > 0\}, \tag{4}$$

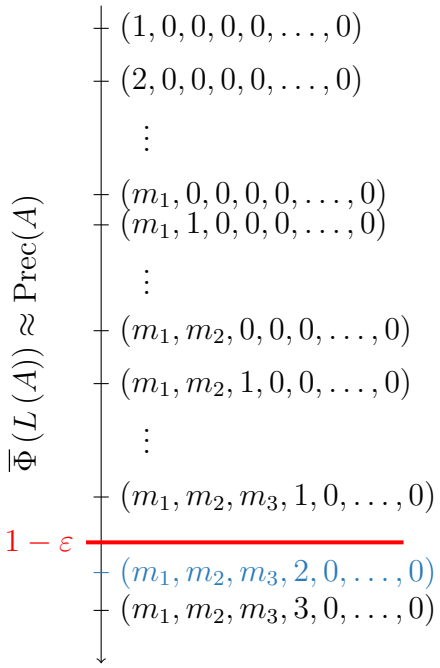
the set of anchors with support corresponding to words with a positive influence.

Proposition 2 (Approximate precision maximization) *Assume the words to be ordered such that* $\lambda_1 v_1 > \lambda_2 v_2 > \cdots > \lambda_d v_d$*, with at least one* λ_j *greater than zero. Assume that* $\lambda_0 > -\gamma/2$*. Then the Algorithm applied to the selection function* $p := \overline{\Phi} \circ L$ *will select an anchor* $A^p \in \mathcal{A}_+$ *such that there exists* $j_0 \in [d]$ *with the following property: for all* $j < j_0$*,* $a_j = m_j$*,* $a_{j_0} \leq m_j$*, and for all* $j \geq j_0$*,* $a_j = 0$*.*

In other words, Proposition 2 implies that Anchors keeps only words that have a positive influence on the prediction for a linear classifier: this is a reassuring property. Furthermore, it prioritizes the words with the highest $\lambda_j v_j$s, adding them to the anchor until the precision condition is met. See Figure 2 for an illustration.

We conducted the following experiment to empirically validate this Property. We first trained a logistic model on three review datasets, achieving accuracies between 85% and 88% on the test set. We then ran Anchors with the default setting 10 times on positively classified documents. For each document, we measure the Jaccard similarity between the anchor A and the first $|A|$ words ranked by $\lambda_j v_j$. Right panel on Figure 2 reports the average Jaccard index: results confirm Proposition 2.

It is important to note that the official implementation of Anchors is not designed to output the best anchor. In cases where the prediction is *easy*, such as when $g(\varphi(\xi)) \geq 0.75$ or $g(\varphi(\xi)) \geq 0.85$, there may be multiple anchors that satisfy the criteria, and the algorithm selects one at random. This explains the varying similarity between the full dataset and the harder subsets.



		Restaurants	Yelp	IMDB
full		0.69 ± 0.2	0.35 ± 0.2	0.32 ± 0.2
$\mathtt{pr} < 0.85$		0.78 ± 0.2	0.74 ± 0.2	0.45 ± 0.2
$\mathtt{pr} < 0.75$		0.82 ± 0.1	0.80 ± 0.2	0.59 ± 0.1

Figure 2: On the left, illustration of Proposition 2. On linear models, the algorithm includes words having the highest $\lambda_j v_j$s first. Finally, the minimal anchor satisfying the precision condition $\overline{\Phi}(L(A)) \approx \mathrm{Prec}(A) \geq 1-\varepsilon$ is selected, which is $A = (m_1, m_2, m_3, 2, 0, \ldots, 0, 0)$ in the example. On the right, validation of Proposition 2. Average Jaccard similarity between the anchor A and the first $|A|$ words ranked by $\lambda_j v_j$ for a logistic model on positive documents and low-confidently classified subset ($\mathtt{pr} = g(\varphi(\xi)) < 0.85$, or $\mathtt{pr} < 0.75$).

5 Conclusion

In this paper, we presented the first theoretical analysis of Anchors, focusing on its implementation for textual data and providing insights on the sampling procedure. Our study mainly focused on Anchors' behavior on linear models, and to this end, we introduced an approximate, tractable version of the algorithm that is similar to the default implementation. Our analysis demonstrated that Anchors provides meaningful results when applied to these models, which is supported by experiments with the official implementation.

Our work highlights the significance of theoretical analysis in the development of explainability methods. We believe that the insights presented in this paper will be valuable for researchers and practitioners in natural language processing in correctly comprehending the explanations provided by Anchors. Moreover, the analytical framework we presented can benefit the explainability community by facilitating the development of new methods founded on robust theoretical principles and analyzing current ones. As future work, we plan to extend this analysis to other classes of models and different types of data, including images and tabular data.

Acknowledgements. This work has been supported by the French government, through the NIM-ML project (ANR-21-CE23-0005-01), and by EU Horizon 2020 project AI4Media (contract no. 951911).

References

Adadi, A. and Berrada, M. (2018). Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE access*, 6:52138–52160.

Amoukou, S. I. and Brunel, N. J. B. (2022). Consistent Sufficient Explanations and Minimal Local Rules for explaining regression and classification models. In *Advances in Neural Information Processing Systems*.

Barbiero, P., Ciravegna, G., Giannini, F., Lió, P., Gori, M., and Melacci, S. (2022). Entropy-based logic explanations of neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6046–6054.

Berry, A. C. (1941). The accuracy of the gaussian approximation to the sum of independent variates. *Transactions of the American Mathematical Society*, 49(1):122–136.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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.

Esseen, C.-G. (1942). On the Liapunov limit error in the theory of probability. *Ark. Mat. Astr. Fys.*, 28:1–19.

Garreau, D. and Luxburg, U. (2020). Explaining the explainer: A first theoretical analysis of LIME. In *International Conference on Artificial Intelligence and Statistics*, pages 1287–1296. PMLR.

Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., and Giannotti, F. (2018). Local rule-based explanations of black box decision systems. *arXiv preprint arXiv:1805.10820*.

Hase, P., Xie, H., and Bansal, M. (2021). The out-of-distribution problem in explainability and search methods for feature importance explanations. *Advances in Neural Information Processing Systems*, 34:3650–3666.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Lakkaraju, H., Bach, S. H., and Leskovec, J. (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.

Lim, B. Y., Dey, A. K., and Avrahami, D. (2009). Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 2119–2128.

Linardatos, P., Papastefanopoulos, V., and Kotsiantis, S. (2021). Explainable AI: A Review of Machine Learning Interpretability Methods. *Entropy*, 23(1):18.

Lopardo, G. and Garreau, D. (2022). Comparing Feature Importance and Rule Extraction for Interpretability on Text Data. In *ICPR 2-nd Workshop on Explainable and Ethical AI - 26TH International Conference on Pattern Recognition (XAIE @ ICPR) 2022*.

Lopardo, G., Precioso, F., and Garreau, D. (2023). A Sea of Words: An In-Depth Analysis of Anchors for Text Data. In *International Conference on Artificial Intelligence and Statistics*.

Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development*, 1(4):309–317.

Lundberg, S. M. and Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, 30:4765–4774.

Mardaoui, D. and Garreau, D. (2021). An analysis of LIME for text data. In *International Conference on Artificial Intelligence and Statistics*, pages 3493–3501. PMLR.

Ribeiro, M. T., Singh, S., and Guestrin, C. (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.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

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

Stumpf, S., Rajaram, V., Li, L., Burnett, M., Dietterich, T., Sullivan, E., Drummond, R., and Herlocker, J. (2007). Toward harnessing user feedback for machine learning. In *Proceedings of the 12th international conference on Intelligent user interfaces*, pages 82–91.

Wang, F. and Rudin, C. (2015). Falling rule lists. In *Artificial intelligence and statistics*, pages 1013–1022. PMLR.