
A Sea of Words: An In-Depth Analysis of Anchors for Text Data

Gianluigi Lopardo

Université Côte d’Azur, Inria,
CNRS, LJAD, France

Frédéric Precioso

Université Côte d’Azur, Inria,
CNRS, I3S, France

Damien Garreau

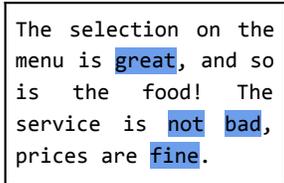
Université Côte d’Azur, Inria,
CNRS, LJAD, France

Abstract

Anchors (Ribeiro et al., 2018) is a post-hoc, rule-based interpretability method. For text data, it proposes to explain a decision by highlighting a small set of words (an anchor) such that the model to explain has similar outputs when they are present in a document. In this paper, we present the first theoretical analysis of Anchors, considering that the search for the best anchor is exhaustive. After formalizing the algorithm for text classification, we present explicit results on different classes of models when the vectorization step is TF-IDF, and words are replaced by a fixed out-of-dictionary token when removed. Our inquiry covers models such as elementary if-then rules and linear classifiers. We then leverage this analysis to gain insights on the behavior of Anchors for any differentiable classifiers. For neural networks, we empirically show that the words corresponding to the highest partial derivatives of the model with respect to the input, reweighted by the inverse document frequencies, are selected by Anchors.

1 INTRODUCTION

As with other areas of machine learning, the state-of-the-art in natural language processing consists of very complex models based on hundreds of millions or even billions of parameters (Devlin et al., 2019; Brown et al., 2020). This complexity is a huge limitation to the use of machine learning algorithms in critical or sensitive contexts: users and domain experts are reluctant to adopt decisions they cannot understand. In recent years, several methods have been proposed to meet the growing demand for *interpretability*. In particular, *local model-agnostic* techniques have



The selection on the menu is **great**, and so is the food! The service is **not bad**, prices are **fine**.

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 $A = \{great, not, bad, fine\}$ (in blue), having length $|A| = 4$ is selected. Intuitively, these four words together ensure a positive prediction by f with high probability (precision : 0.97), while being not too uncommon (coverage : 0.12).

emerged to explain the individual predictions of any classifier for a specific instance. The unique assumption is that the model can be queried as often as necessary. However, the process generating the explanations can be, for a user, as mysterious as the prediction to be explained. Moreover, interpretability methods often lack solid theoretical guarantees. In particular, their behavior on simple, already interpretable models is often unknown. Instead of helping, applying a poorly understood explainer to a complex model can lead to misleading results.

In this work, we focus on Anchors (Ribeiro et al., 2018), an increasingly popular, local model-agnostic method more and more included in explainability toolboxes such as Alibi¹, and more precisely its implementation for text data. For a given prediction, Anchors’ general idea is to provide a simple rule yielding the same prediction with high probability if it is satisfied. These rules can be formulated as the presence of a list of words in the document to be explained, and are presented as such to the user (see Figure 1). If the model to explain is intrinsically interpretable, in particular if we know with absolute certainty which words are important for the prediction, will Anchors highlight these words in the explanation?

First, in Section 2, we explain the basic concepts of Anchors and formalize its mechanism for text classification.

¹<https://github.com/SeldonIO/alibi>

We then delve into the definition of a more tractable, exhaustive version of the algorithm in Section 3, which constitutes the central object of our study. Next, in order to understand the efficacy of Anchors for text data, we perform a theoretical and empirical analysis of its behavior on explainable classifiers in Section 4. This allows us to gain new insights that can be extended to broader classes of models. In particular, in Section 5, we empirically show a surprising result on neural networks. This section provides valuable results into the real-world applications of Anchors that can be applied to explain document classifiers. Finally, in Section 6, we draw our conclusions and summarize the findings of our study. Some unexpected results from our research highlight the importance of theoretical analysis for explainers. We believe that the insights presented in this article will be useful for researchers and practitioners in the field of natural language processing to accurately interpret the explanations provided by Anchors. On the other hand, the framework we have designed for this analysis can be of great use to the explainability community, both in designing new methods with solid theoretical foundations and in analyzing existing ones.

Contributions. In this paper, we present the first theoretical analysis of Anchors for text data, based on the default implementation available on Github² (as of February 2023). The main restrictions of our analysis are the simplification of the combinatorial optimization procedure (therefore considering an *exhaustive* version of Anchors), the use of an out of dictionary token when removing words, and the assumption that a TF-IDF vectorization is used as a preprocessing step. Specifically,

- we dissect Anchors’ algorithm for text classification, showing that the sampling procedure can be described simply as an i.i.d. Bernoulli’s removal of words not belonging to the anchor (Proposition 1);
- we show that the exhaustive version is stable with respect to perturbation of the precision function, justifying our study of the exhaustive Anchors algorithm (Proposition 2 and 3);
- if the classifier ignores some words, they will not appear in the anchor selected by the exhaustive Anchors (Proposition 4);
- exhaustive Anchors for simple if-then rules provably outputs meaningful explanations, though words can be ignored from the explanation if their multiplicity is too high (Proposition 5);
- exhaustive Anchors picks the words associated to the most positive coefficients reweighted by the inverse document frequency for all linear classifiers (Proposition 6 and 7);

- we empirically show that exhaustive Anchors picks the words associated to the most positive partial derivatives scaled by the inverse document frequency for neural networks (Section 5).

All our theoretical claims are supported by mathematical proofs, available in Section A of the Appendix, and numerical experiments, detailed in Section C, whose code is available at https://github.com/gianluigilopardo/anchors_text_theory. Unless otherwise specified, experiments use the official implementation of Anchors with all default options.

Related work. Among several methods for machine learning interpretability proposed in recent years (Guidotti et al., 2018; Adadi and Berrada, 2018; Linardatos et al., 2021), rule-based methods are increasingly popular contenders. One reason is that users prefer rule-based explanations rather than alternatives (Lim et al., 2009; Stumpf et al., 2007). Hierarchical decision lists (Wang and Rudin, 2015) are useful for understanding the global behavior of a model, prioritizing the most interesting cases. Lakkaraju et al. (2016) compromises between accuracy and interpretability to extract small and disjoint rules, simpler to interpret, introducing the concept of *coverage*. 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.

Many other approaches focus on *local interpretability*, based on the idea that any black-box model can be approximated accurately by a simpler—easier to understand—model around a specific instance to explain. As an example, LORE (Guidotti et al., 2018) uses a decision tree: the explanation is the list of logical conditions satisfied by the instance within the tree. A central point of perturbation-based methods is the sampling scheme. Delaunay et al. (2020) modifies Anchors’ sampling for tabular data, implementing the Minimum Description Length Principle discretization (Fayyad and Irani, 1993) to learn the minimal number of intervals needed to separate instances from distinct classes. Amoukou and Brunel (2022) proposes Minimal Sufficient Rules, similar to anchors for tabular data, extended to regression models, that can directly deal with continuous features, with no need for a discretization.

There are few local, post-hoc explainability techniques for text data (Danilevsky et al., 2020). Among them, LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017) provide explanations using a linear model as local surrogate, trained on perturbed samples of the instance to explain. As we will see, 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, in *high probability*, the same prediction as the example. Delaunay et al. (2020) proposes to extend Anchors by also exploiting the absence of words.

²<https://github.com/marcotcr/anchor>

In this work, our main concern is to provide some theoretical guarantees for interpretability methods. For feature importance methods, Lundberg and Lee (2017) provides insights in the case of linear models for kernel SHAP, while Mardaoui and Garreau (2021) looks specifically into LIME for text data, extending Garreau and Luxburg (2020). These last two papers also consider simple if-then rules and linear models in their analysis. Another related work is Agarwal et al. (2022): graph neural networks explainers are compared in terms of faithfulness, stability, and fairness. As for rule-based methods, the question has not been addressed yet, to the best of our knowledge.

2 ANCHORS FOR TEXT DATA

In this section, we present the operating procedure of Anchors for text data, as introduced by Ribeiro et al. (2018). After specifying our setting and notation in Section 2.1, we present the key notions of *precision* and *coverage* in Section 2.2. The algorithm is described in Section 2.3. We give further details on the sampling scheme in Section 2.4.

2.1 Setting and notation

Throughout this paper, we consider the problem of explaining the decision of a classifier f taking documents as input. We will denote by z a generic document, and by ξ the particular example being explained by Anchors. Let us define $\mathcal{D} = \{w_1, \dots, w_D\}$ the *global dictionary* with cardinality $D = |\mathcal{D}|$. We see any document as a finite sequence of elements of \mathcal{D} . For a given document $\xi = \{\xi_1, \dots, \xi_b\}$ of b (not necessarily distinct) words, up to a re-ordering of \mathcal{D} , we can set $\mathcal{D}_\xi = \{w_1, \dots, w_d\} \in \mathcal{D}$ the *local dictionary*, containing the distinct words in ξ , with $d \leq b$. We denote by $m_j(z)$ the multiplicity of word w_j in z , i.e., $m_j(z) := |\{k \in [b], z_k = w_j\}|$. When the context is clear, we write m_j short for $m_j(\xi)$. Finally, for any integer k , we write $[k] := \{1, \dots, k\}$.

We make two restrictive assumptions on the class of models that we take into account. First, we restrict our analysis to *binary classification* and write $f(z) = \mathbf{1}_{g(z) \in R}$, where $g : \mathcal{T} \rightarrow \mathbb{R}^p$ is a given measurable function taking document as input, and R is a collection of (potentially unbounded) intervals of \mathbb{R}^p . Second, we assume that g relies on a *vectorization* of the documents. More precisely, we assume that $g = h \circ \varphi$, where φ is a deterministic mapping $\mathcal{T} \rightarrow \mathbb{R}^D$ (detailed in Section 4.1) and $h : \mathbb{R}^D \rightarrow \mathbb{R}^p$. Without loss of generality, **we will always assume that the example ξ is classified as positive**, i.e., $f(\xi) = 1$. In definitive, we consider models of the form

$$f(z) = \mathbf{1}_{h(\varphi(z)) \in R}. \quad (1)$$

We call an *anchor* any non-empty subset of $[b]$, corresponding to a preserved set of words of ξ . We set \mathcal{A} the set of

all candidate anchors for the example ξ . For any anchor $A \in \mathcal{A}$, we set $|A|$ the length of the anchor, defined as the number of (not necessarily distinct) words contained in the anchor. In practice, **an anchor A for a document ξ is represented as a non-empty sublist of the words present in the document**, and this is the output of Anchors (illustrated in Figure 1).

2.2 Precision and coverage

The *precision* of an anchor $A \in \mathcal{A}$ is defined by Ribeiro et al. (2018) as the probability for a local perturbation of ξ to be classified as 1. Since we assume $f(\xi) = 1$, the precision can be written as

$$\text{Prec}(A) = \mathbb{E}_A [f(x) = 1] = \mathbb{P}_A (g(\varphi(x)) = 1), \quad (2)$$

where the expectation is taken with respect to x , a random perturbation of ξ still containing all the words included in the anchor A . We detail the sampling of x further in Section 2.4. For the anchor containing all the words of ξ , the precision is exactly 1, while smaller anchors have, in general, smaller precision.

Of course, large anchors with size comparable to b are not very interesting from the point of view of interpretability (the text in Figure 1 would be completely highlighted). To quantify this idea, one can use the notion of *coverage*, defined in our case as the proportion of documents in the corpus (i.e., the dataset of documents on which the vectorizer is fitted) that contain the anchor. For instance, the coverage of the anchor in Figure 1 is 0.12, meaning that 12% of the reviews contain it. The notions of precision and coverage are paramount to the Anchors algorithm: in a nutshell, **Anchors will look for an anchor of maximal coverage with prescribed precision**. We detail this in the next section.

2.3 The algorithm

In practice, the coverage can be costly to compute, and in many cases a corpus is not available when the prediction is explained. Since anchors with smaller length tend to have larger coverage, a natural solution, used in the default implementation, is to minimize the length instead of maximizing the coverage, leading to:

$$\text{Minimize } |A|, \text{ such that } \text{Prec}(A) \geq 1 - \varepsilon, \quad (3)$$

where $\varepsilon > 0$ is a pre-determined tolerance threshold (set to 0.05 in practice). The lower ε is, the harder it is to find an anchor satisfying Eq. (3).

Of course, the exact precision of a specific anchor $A \in \mathcal{A}$ is unknown, since we cannot compute the expectation appearing in Eq. (2) in general. The strategy used by Ribeiro et al. (2018) is to approximate $\text{Prec}(A)$ by $\widehat{\text{Prec}}_n(A)$, an empirical approximation, defined in Section 3. Let us note

that the optimization problem in Eq. (3) is generally intractable, whatever the selection function may be. The cardinality of \mathcal{A} is simply too large in all practical scenarios. As a consequence, the default implementation applies the KL-LUCB (Kaufmann and Kalyanakrishnan, 2013) algorithm to identify a subset of rules with high precision: at the next step, this subset is used as representative of all candidate anchors, finding an approximate solution to Eq. (3). In this paper, we do not consider this optimization procedure and consider an exhaustive version of Anchors, described in Section 3.

2.4 The sampling

We now detail the sampling procedure performed to compute the precision of an anchor (see Eq. (2)). The idea is to look at the behavior of the model f in a local neighborhood of ξ , while fixing the set of words in A . In the official implementation, this amounts to setting `use_unk_distribution=True` (default choice). Formally, for a given example ξ and for a candidate anchor $A \in \mathcal{A}$, Anchors generates perturbed samples x_1, \dots, x_n in the following way:

1. *copies generation*: create n identical copies x_1, \dots, x_n of the example to explain ξ ;
2. *random selection*: for each word with index $k \in [b]$ not belonging to the anchor, draw $B_k \sim \text{Bin}(n, 1/2)$ a number of copies to be perturbed (words in the candidate anchor are the blue columns of Table 2);
3. *word replacement*: for each word not in the anchor, draw independently uniformly at random a set S_k of cardinality B_k of copies to be perturbed. Replace the words belonging to copies whose indices are in S_k by the token “UNK.”

Note that the perturbation distribution described in this section is different from what is described in Ribeiro et al. (2018), *i.e.*, replacing selected words with others having the same *part-of-speech* tag “with probability proportional to their similarity in an embedding space.” In fact, replacing words with a predefined token generates meaningless sentences which can fool a classifier, as it produces unrealistic samples (Hase et al., 2021). However, this option is not implemented in the official release, while it is possible to replace the selected words in step 3 using BERT (Devlin et al., 2019). In this work, we nevertheless consider the UNK-replacement because (i) we believe the default choices to be the most used by Anchors’ users, thus the ones most needing interpretation and theoretical guarantees, and (ii) as we detail in Section 4.1, in the case of TF-IDF vectorization, the UNK-replacement exactly replicates word removals. Nevertheless, experiments show that our results still hold when the BERT-replacement is applied (see Section C.8 of the Appendix).

	ξ_1	ξ_2	ξ_3	ξ_4	ξ_5	ξ_6	ξ_7	ξ_8	ξ_9
ξ	the	quick	brown	fox	jumps	over	the	lazy	dog
x_1	UNK	UNK	brown	fox	jumps	over	the	lazy	dog
x_2	the	quick	brown	UNK	jumps	UNK	the	lazy	dog
\vdots									
x_n	the	quick	brown	UNK	jumps	over	the	lazy	UNK

Figure 2: Anchors’ sampling is performed in three main steps: copies generation, random selection, word replacing. Here, for instance, for the fourth word (“fox”), $B_4 = 2$ and $S_4 = \{2, n\}$, so the second and the n -th copies are considered for replacement.

We remark that Anchors’ sampling procedure is similar to that of LIME for text data (Ribeiro et al., 2016), with the crucial difference that LIME removes *all occurrences* of a given word when it is selected for removal. We refer to Mardaoui and Garreau (2021) for more details. We show in Appendix A.1 that the sampling procedure can be described more simply:

Proposition 1 (Equivalent sampling). *The sampling process described above is equivalent to replacing, for any sample x_i , each word $x_{i,k}$ such that $k \notin A$ independently with probability $1/2$.*

Intuitively, parsing each line of Table 2, Anchors flips an imaginary coin for each word not belonging to the anchor, replaces it in the perturbed example if the coin hits heads and keeps it if it hits tails. Proposition 1 is the first step in our analysis, giving us a simple description of M_j , the random variable defined as the multiplicity of word w_j in the perturbed sample x . Namely, for any given anchor A , $M_j \sim a_j + \text{Bin}(m_j - a_j, 1/2)$, where a_j is the number of occurrences of w_j in A .

3 EXHAUSTIVE p -ANCHORS

In this section, we present the central object of our study, exhaustive p -Anchors. In a nutshell, it is a formalized version of the original combinatorial optimization problem of Eq. (3) for any evaluation function $p : \mathcal{A} \rightarrow \mathbb{R}$. We describe the procedure in Section 3.1, and thereafter provide a key stability property motivating further investigations in Section 3.2.

3.1 Description of the algorithm

The optimization problem of Eq. (3) can be decomposed in two steps: first, all anchors in \mathcal{A} such that $\text{Prec}(A) \geq 1 - \varepsilon$ are selected. We call this first subset of anchors $\mathcal{A}_1(\varepsilon)$. Note that $\mathcal{A}_1(\varepsilon) \neq \emptyset$ since the full anchor $[b]$ has precision 1. Then, among these anchors, the ones with minimal length are kept, giving raise to $\mathcal{A}_2(\varepsilon)$. At this point, it is not clear from Eq. (3) which anchors should be selected, and we settle for the ones with the highest precision. Equality cases can happen at this step (for instance, there can be

Algorithm 1 An overview of exhaustive p -Anchors.

input set of candidate anchors \mathcal{A} , selection function $p : \mathcal{A} \rightarrow \mathbb{R}$, tolerance threshold ε

select $\mathcal{A}_1^p(\varepsilon) = \{A \in \mathcal{A} \text{ s.t. } p(A) \geq 1 - \varepsilon\}$

select $\mathcal{A}_2^p(\varepsilon) = \arg \min_{A' \in \mathcal{A}_1^p(\varepsilon)} |A'|$

select $\mathcal{A}_3^p(\varepsilon) = \arg \max_{A' \in \mathcal{A}_2^p(\varepsilon)} p(A')$

select $A^p(\varepsilon) \in \mathcal{A}_3^p(\varepsilon)$ uniformly at random

return $A^p(\varepsilon)$

several anchors with precision 1): we call $\mathcal{A}_3(\varepsilon)$ the corresponding set of anchors. If $\mathcal{A}_3(\varepsilon)$ is not reduced to a single element, we draw uniformly at random the selected anchor. Algorithm 1 formally describes this procedure for a generic evaluation function $p : \mathcal{A} \rightarrow \mathbb{R}$, which we illustrate in Figure 3. When using p , we write $\mathcal{A}_k^p(\varepsilon)$ the sets constructed and $A^p(\varepsilon)$ the selected anchor.

The goal here is to have a flexible framework: we can use Algorithm 1 with $p = \widehat{\text{Prec}}_n$ or $p = \text{Prec}$ as a selection function, or any other function which is a good approximation of Prec . When $p = \text{Prec}$, we call this version of the algorithm *exhaustive Anchors*, whereas when $p = \widehat{\text{Prec}}_n$ we call this version *empirical Anchors*.

Empirical Anchors is very similar to Anchors; the main difference is that the former is looking at all possible anchors, while the latter uses an efficient approximate procedure, which we do not consider here. A second difference is that empirical Anchors selects anchors with maximal precision in the third step. This is not necessarily the case with the default implementation, since an approximate procedure is used. We notice, nevertheless, that the chosen anchors tend to have high precision, and we demonstrate in Section C.2 of the Appendix that empirical Anchors and the default implementation give very similar output in practice.

3.2 Stability with respect to the evaluation function

We show that applying Algorithm 1 to functions taking similar values on \mathcal{A} leads to similar results.

Proposition 2 (Stability of exhaustive p -Anchors). *Let $\varepsilon > 0$ be a tolerance threshold, $p : \mathcal{A} \rightarrow \mathbb{R}$ be an evaluation function, and set $A^* := A^p(\varepsilon)$ the output of exhaustive p -Anchor. Assume that (i) $p(A^*) \geq 1 - \varepsilon/4$, and (ii) $p(A) \leq 1 - 3\varepsilon/4$ for any $A \in \mathcal{A}_2^p(\varepsilon) \setminus \{A^*\}$. Let $q : \mathcal{A} \rightarrow \mathbb{R}$ be another evaluation function such that*

$$\delta := \sup_{A \in \mathcal{A}} |p(A) - q(A)| < \frac{\varepsilon}{4}. \quad (4)$$

Then $A^q(\varepsilon - \delta) = A^*$.

In other words, if A^* is a solution with high value for the chosen p function, and we have a good approximation q of p , then running Algorithm 1 on q instead of p will yield

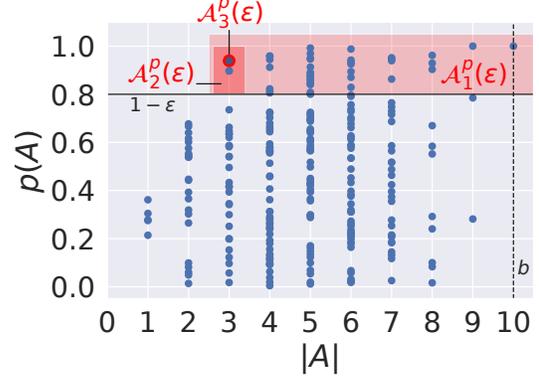


Figure 3: An illustration of Algorithm 1 with evaluation function $p = \text{Prec}$. Each blue dot is an anchor, with x coordinate its length and y coordinate its value for p . Here, $\varepsilon = 0.2$ and the maximal length of an anchor is $b = 10$ (the length of ξ). In the end, the anchor A such that $|A| = 3$ and $p(A) = 0.9$ is selected (red circle).

approximately the same result. **This is the key motivation for studying Prec instead of $\widehat{\text{Prec}}$, and later considering further approximations to Prec :** one can study directly Prec , or an approximation thereof, and get insights on the output of the original algorithm. We prove Proposition 2 in Section A.2 of the Appendix.

Note that, since we perturb the values of the p function, an anchor with smaller length than A^* could cross the $1 - \varepsilon$ barrier and become a solution for exhaustive q -Anchors if we kept the same tolerance threshold ε . This is for instance the case for the anchor with length two and highest value of p in Figure 3. Therefore, the tolerance threshold has to decrease: Proposition 2 cannot be improved to show that $A^q(\varepsilon) = A^*$. However, having large value of p for A^p (assumption (i)) is not strictly necessary, what is important is that the gap between A^p and the anchor with second-largest value of p function in $\mathcal{A}_2^p(\varepsilon)$ cannot be filled by q (assumption (ii)). Otherwise, an anchor with the same length could get a larger value for q than A^p and be selected in the final step.

As a first application of Proposition 2, let us come back to the *empirical precision*,

$$\widehat{\text{Prec}}_n(A) := \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{f(x_i)=1},$$

where n is the number of perturbed samples with A fixed, as described in Section 2.4. Then the empirical precision satisfies the following:

Proposition 3 ($\widehat{\text{Prec}}_n(A)$ uniformly approximates Prec). *Recall that we denote by b the number of words in ξ . Let $\delta > 0$. With probability higher than $1 - 2^{b+1}e^{-2n\delta^2}$,*

$$\forall A \in \mathcal{A}, \quad \left| \widehat{\text{Prec}}_n(A) - \text{Prec}(A) \right| \leq \delta. \quad (5)$$

In particular, Proposition 3 guarantees that $\widehat{\text{Prec}}_n$ and Prec satisfy Eq. (4) with high probability, as soon as $n \gg b/\varepsilon^2$. This is the main motivation for studying *exhaustive Anchors* in the next section: Proposition 2 shows that *exhaustive Anchors* and *empirical Anchors* will output the same result with high probability. We prove Proposition 3 in Section A.3 of the Appendix.

4 ANALYSIS ON EXPLAINABLE CLASSIFIERS

Before presenting our main results, we describe the vectorizer we are considering in Section 4.1. We then go into the analysis by studying Anchors’ behavior when applied to simple rule-based classifiers in Section 4.2 and to linear models in Section 4.3. All our claims are supported by mathematical proofs, available in Section A of the Appendix, and validated by reproducible experiments.

4.1 Vectorizers and immediate consequences

Natural language processing classifiers are mostly based on a vector representation φ of the document (Young et al., 2018). The TF-IDF (Term Frequency-Inverse Document Frequency) transform Luhn (1957) is popularly used to obtain such vectorization. The principle is to assign more weight to words that appear frequently in the document z , and not so frequently in the corpus \mathcal{C} . We perform our analysis by assuming that models work with the (non-normalized) TF-IDF vectorizer:

Definition 1 (TF-IDF). *Let N be the size of the initial corpus \mathcal{C} , i.e., the number of documents in the dataset. Let N_j be the number of documents containing the word $w_j \in \mathcal{D}$. The TF-IDF of z is the vector $\varphi(z) \in \mathbb{R}^D$ defined as*

$$\forall j \in [D], \quad \varphi(z)_j := m_j(z)v_j,$$

where $v_j := \log \frac{N+1}{N_j+1} + 1$ is the inverse document frequency (IDF) of w_j in \mathcal{C} .

Note that once the TF-IDF vectorizer is fitted on a corpus \mathcal{C} , the vocabulary is fixed once and forever afterward. Meaning that, if a word is not part of the initial corpus \mathcal{C} , its IDF term is zero. As seen in Section 2.4, Anchors perturbs documents by replacing words with a fixed token “UNK”. We make the (realistic) assumption that the word “UNK” is not present in the initial corpus. Thus, **replacing any word with this token is equivalent to simply removing it, from the point of view of TF-IDF.**

In this paper, we will always consider models trained on a (non-normalized) TF-IDF vectorization φ as in Definition 1. However, we show in Section A.9 of the Appendix that the same results hold for the ℓ_2 -normalized TF-IDF

vectorization, defined as

$$\forall j \in [D], \quad \phi(z)_j := \frac{m_j(z)v_j}{\sqrt{\sum_{j=1}^D m_j(z)^2 v_j^2}}.$$

We note that this is the default normalization in the `scikit-learn` implementation of TF-IDF.

Since our models are of the form Eq. (1), whenever the vectorizer is applied, **the exact location of the words in the document does not matter**. Therefore, when computing precision, only the occurrences of word w_j in anchor A matter. For this reason, we write an anchor as $A = (a_1, \dots, a_d, \dots, a_D)$. Since $a_j \leq m_j$ for all $j \in [D]$, we see that $a_j = 0$ for any $j > d$. Thus, we can write $A = (a_1, \dots, a_d)$ without ambiguity.

With this notation in hand, according to the discussion following Proposition 1, the TF-IDF of w_j in the perturbed sample x will satisfy

$$\varphi(x)_j = M_j v_j \sim (a_j + \text{Bin}(m_j - a_j, 1/2))v_j, \quad (6)$$

where M_j is the random multiplicity of the word w_j . Intuitively, M_j corresponds to a_j occurrences of w_j which cannot be removed, plus a random number of occurrences depending on the sampling. This has several important consequences in our analysis, the first being:

Proposition 4 (Dummy features). *Let f be defined as in Eq. (1) and assume that g does not depend on coordinate j . Let $\varepsilon > 0$ be a tolerance threshold. Then, for any anchor $A \in \mathcal{A}_3^{\text{Prec}}(\varepsilon)$, $a_j = 0$.*

Proposition 4 is a natural property: if the model does not depend on word w_j , then it should not appear in the explanation. It is often investigated in the interpretability literature, originally introduced by Friedman (2004). Here we use the vocabulary of Sundararajan et al. (2017), which introduced the notion as an axiom for feature importance.

Note that Proposition 4 is not satisfied by the empirical version (with $p = \widehat{\text{Prec}}_n$), neither by the default implementation of Anchors (see Section C.3 in the Appendix). We prove Proposition 4 in Section A.4 of the Appendix.

4.2 Simple if-then rules

We now focus on classifiers relying on the presence or absence of given words. In this setting, the function g introduced in Eq. (1) will take a simple form. Indeed, since we are working with the TF-IDF vectorizer, the *presence* (resp. *absence*) of word w_j in x simply corresponds to the condition $\varphi(x)_j > 0$ (resp. $\varphi(x)_j = 0$).

Thus g is the projection on the relevant coordinates, and using Eq. (6) we will be able to compute the precision of any given anchor. We show here the case of a model classifying documents according to the presence of a set of words. Different cases are presented in Section C.5 of the Appendix.

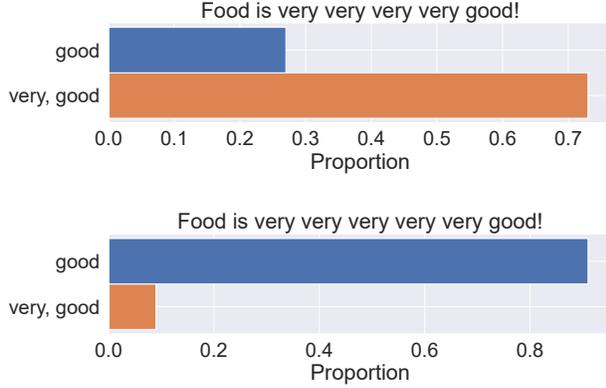


Figure 4: Making a word disappear from the explanation by adding one occurrence. The model predicts 1 if “very” and “good” are present. The count corresponds to the appearance of the word(s) in the selected anchor on 100 runs of Anchors. When the multiplicity of “very” in the document crosses the breakpoint value $B = 4$ (by default, $\varepsilon = 0.05$), it disappears from the selected anchor with high probability (on the top panel, $m_{\text{very}} = 4$, on the bottom panel, $m_{\text{very}} = 5$).

Proposition 5 (Presence of a set of words). *Assume that $m_1 > \dots > m_k$. Let $J = \{1, \dots, k\}$ be a subset of $[d]$, and assume that the model is defined as*

$$f(z) = \mathbf{1}_{w_j \in z, \forall j \in J} = \prod_{j \in J} \mathbf{1}_{w_j \in z} = \prod_{j \in J} \mathbf{1}_{\varphi(z)_j > 0}.$$

Let us define the quantities $B := \lfloor \log_2 1/\varepsilon \rfloor$ and $c_0 := \inf\{c \in [k], \prod_{\ell=1}^{k-c+1} (1 - 2^{-\ell}) \geq 1 - \varepsilon\}$. If there exists $j \in J$ such that $m_j > B$, then the anchor A_{c_0} such that $a_j = 1$ for all $j \in \llbracket k - c_0 + 1, k \rrbracket$ and $a_j = 0$ otherwise will be selected by exhaustive Anchors. On the contrary, if $m_j < B$ for all $j \in [d]$, the anchor A_J such that $a_j = 1$ for all $j \in [k]$ and $a_j = 0$ otherwise will be selected.

Note that A_J contains all the words with index in J , i.e., each word in the support of the classifier f . Instead, A_{c_0} contains the c_0 words with the lowest multiplicity among those indexed in J . We prove Proposition 5 in Section A.5 of the Appendix.

As a concrete example, let us consider a model classifying reviews as positive if both the words “very” and “good” are present. The review “Food is very good!” will be classified as a positive one and Anchors extracts {“very”, “good”} as an anchor, which makes a lot of sense. However, if the multiplicity of the word “very” (resp. “good”) exceeds the breakpoint value $B = 4$ (by default, $\varepsilon = 0.05$), Proposition 5 predicts that Anchors will only extract {“good”} (resp. {“very”}). In this example, **we are effectively seeing a word disappear from an explanation simply because its multiplicity crosses an arbitrary threshold.** This is verified in practice, with the default implementa-

tion of Anchors (see Figure 4), with some variability coming from the sampling and the approximate optimization scheme of Anchors.

4.3 Linear classifiers

We now shift our focus to *linear classifiers*. Namely, in this section, for any document z , we have

$$f(z) = \mathbf{1}_{\lambda^\top \varphi(z) + \lambda_0 > 0}, \quad (7)$$

where $\lambda \in \mathbb{R}^D$ is a vector of coefficients, and $\lambda_0 \in \mathbb{R}$ is an intercept. Coming back to Eq. (1), $g(y) = \lambda^\top y + \lambda_0$, and $R = (0, +\infty)$. Eq. (7) encompasses several important examples, two of which we investigate empirically:

- **the perceptron** (Rosenblatt, 1958), predicting exactly as in Eq. (7);
- **logistic models**, predicting as 1 if $\sigma(\lambda^\top \varphi(z) + \lambda_0) > 1/2$, where $\sigma : \mathbb{R} \rightarrow \sigma(t) = \frac{1}{1+e^{-t}} \in [0, 1]$ is the *logistic function*. Since $\sigma(y) > 1/2$ if, and only if, $y > 0$, they can also be rewritten as in Eq. (7).

Of course, this list is not exhaustive, we refer to Chapter 4 in Hastie et al. (2009) for a more complete overview. In this setting, starting from Eq. (6), we show that the precision satisfies a Berry-Esseen-type statement (Berry, 1941; Esseen, 1942):

Proposition 6 (Precision of a linear classifier). *Let λ, λ_0 be the coefficients associated to the linear classifier defined by Eq. (7). Assume that for all $j \in [d]$, $\lambda_j v_j \neq 0$.*

Define, for all $A \in \mathcal{A}$,

$$L(A) := \frac{-\lambda_0 - \frac{1}{2} \sum_{j=1}^d \lambda_j v_j (m_j + a_j)}{\sqrt{\frac{1}{4} \sum_{j=1}^d \lambda_j^2 v_j^2 (m_j - a_j)}}. \quad (8)$$

Let $\bar{\Phi} := 1 - \Phi$, where Φ denotes the cumulative distribution function of a $\mathcal{N}(0, 1)$. Then, for any $A \in \mathcal{A}$ such that $|A| \leq b/2$,

$$|\text{Prec}(A) - \bar{\Phi}(L(A))| \leq \quad (9)$$

$$C \cdot \left(\frac{\max_j \lambda_j^2 v_j^2}{\min_j \lambda_j^2 v_j^2} \right)^{3/2} \cdot \left(\frac{\max_j m_j}{\min_j m_j} \right)^{3/2} \cdot \frac{1}{\sqrt{d}},$$

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

In other words, when d is large, **the precision of an anchor for a linear classifier can be well approximated by $\bar{\Phi} \circ L$** and we can use a (local) version of Proposition 2 to study exhaustive p -Anchors with $p = \bar{\Phi} \circ L$ instead of exhaustive Anchors. Nevertheless, we observe a very good fit between the two terms even for small values of d (see Section C.4 in the Appendix). Proposition 6 is proven in Section A.6

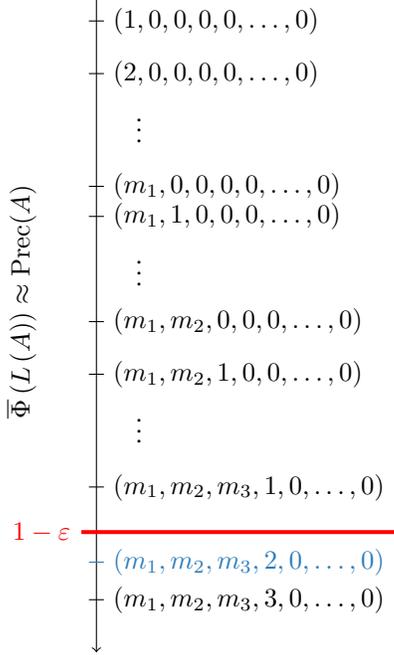


Figure 5: Illustration of Proposition 7. On linear models, the algorithm includes words having the highest $\lambda_j v_j$ s first. Finally, the minimal anchor satisfying the precision condition $\bar{\Phi}(L(A)) \approx \text{Prec}(A) \geq 1 - \varepsilon$ is selected, which is $A = (m_1, m_2, m_3, 2, 0, \dots, 0)$ in the example.

of the Appendix. In Section A.9 of the Appendix we show 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 (see Section C.1 in the Appendix). Thus, the main assumption is the absence of zero components in λ . Note also that the result is only true for anchors having a length less than half the document length. This is realistic, since an explanation based on more than half of the document does not occur in practice and is not really interpretable.

In view of Proposition 2, we can now focus on exhaustive $\bar{\Phi} \circ L$ -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

$$\mathcal{A}_+ := \{A \in \mathcal{A}, \text{ s.t. } a_j > 0 \Rightarrow \lambda_j > 0\}, \quad (10)$$

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

Proposition 7 (Approximate precision maximization). *Assume that $\lambda_1 v_1 > \lambda_2 v_2 > \dots > \lambda_d v_d$, with at least one λ_j greater than zero. Assume that $\lambda_0 > -\gamma/2$. Then Algorithm 1 applied to the selection function $p := \bar{\Phi} \circ L$ will select an anchor $A^p \in \mathcal{A}_+$ such that there exists $j_0 \in [d]$ with the following property: for all $j < j_0$, $a_j = m_j$, $a_{j_0} \leq m_{j_0}$, and for all $j \geq j_0$, $a_j = 0$.*

In plain words, Proposition 7 implies that **for a linear clas-**

Table 1: Validation of Proposition 7. 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 ($\text{pr} = g(\varphi(\xi)) < 0.85$, or $\text{pr} < 0.75$).

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

sifier, Anchors keeps only words with a positive influence on the prediction. Moreover, it **selects the words having the highest $\lambda_j v_j$ s first**, adding them to the anchor until the precision condition is met. This is a reassuring property of Anchors. We prove Proposition 7 in Section A.7 of the Appendix. To demonstrate this phenomenon, we conducted the following experiment. 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 $\lambda_j v_j$. In Table 1 we report the average Jaccard index: results confirm Proposition 7.

Note that the official Anchors’ implementation does not apply step 3 of Algorithm 1: when the prediction is *easy* (for instance, $g(\varphi(\xi)) \geq 0.75$ or $g(\varphi(\xi)) \geq 0.85$), realistically \mathcal{A}_3 contains more than one anchors, and the algorithm will randomly select among them. This explains the different similarity between the full dataset and the *hard* subset. We also remark that the individual multiplicities do not come into play in the ranking of the w_j s, unlike the discussion following Proposition 5. This behavior is consistent with all models that we tested (see Section C.5 of the Appendix).

5 ANCHORS ON NEURAL NETWORKS

In this section, we show empirical results for neural networks linking the explanations provided by Anchors with the partial derivatives of the model with respect to the input. Intuitively, while looking at a specific prediction, Anchors generates local perturbations of the example to explain. The behavior of a neural network on such a local neighborhood of the example, at order one, is approximately linear. This implies that Proposition 6 roughly remains true, taking as linear coefficients

$$\forall j \in [d], \quad \lambda_j := \frac{\partial g(\varphi(x))}{\partial \varphi(x)_j},$$

where $\frac{\partial g(\varphi(x))}{\partial \varphi(x)_j}$ is the partial derivative of the model g with respect to the word w_j . In practice, the implication is that

Table 2: Average Jaccard similarity between the extracted anchor A and the first $|A|$ words ranked by $\lambda_j v_j$ for a 3 (top table), 10 (middle), and 20-layers (bottom) feed-forward neural network for positively classified documents and low-confidently ($\text{pr} = g(\varphi(\xi)) < 0.85$, or $\text{pr} < 0.75$) classified subsets.

		Restaurants	Yelp	IMDB
3-layers	full	0.68 ± 0.2	0.50 ± 0.3	0.45 ± 0.3
	$\text{pr} < 0.85$	0.68 ± 0.2	0.88 ± 0.1	0.52 ± 0.3
	$\text{pr} < 0.75$	0.74 ± 0.2	0.82 ± 0.1	0.68 ± 0.2
10-layers	full	0.76 ± 0.2	0.56 ± 0.2	0.55 ± 0.2
	$\text{pr} < 0.85$	0.80 ± 0.2	0.78 ± 0.1	0.79 ± 0.2
	$\text{pr} < 0.75$	0.83 ± 0.1	0.69 ± 0.2	0.81 ± 0.2
20-layers	full	0.73 ± 0.1	0.60 ± 0.3	0.63 ± 0.2
	$\text{pr} < 0.85$	–	0.81 ± 0.2	0.69 ± 0.2
	$\text{pr} < 0.75$	–	–	0.74 ± 0.1

anchors selects the words corresponding to the highest partial derivatives of the model with respect to the input, reweighted by the inverse document frequencies, until the precision condition is met.

To validate this conjecture, we trained three feed-forward neural networks on three datasets, and measured for each document of the test set the Jaccard similarity between the anchor A and the first $|A|$ words ranked for all $j \in [d]$, by $\lambda_j v_j = \frac{\partial g(\varphi(x))}{\partial \varphi(x)_j} v_j$. This is similar to the experiments from the previous Section. Details on the training are reported in Section C.7 of the Appendix, we achieved accuracy around 90% in each case. We ran Anchors with default setting 10 times on positively classified documents to account for the randomness in Anchors’ optimization.

Table 2 shows the results for the three networks. There is a significant overlap between the anchors selected by Anchors and the subset suggested by our analysis, which becomes a near match for examples that are hard to predict. As for the previous section, we notice that when the prediction is *easy*, *i.e.*, the classifier is very confident about the prediction, the anchor is selected at random among many candidates. This is due to the fact that one needs to strongly perturb (that is, remove a lot of words from) a document confidently classified to change its prediction. Because these models perform better than linear models, confidence is frequently extremely high, making the process much more random, which motivates lower similarity values. More details about the experiments are available in Section C.7 of the Appendix.

Compared with the Anchors’ algorithm for text classification, obtaining explanations for a prediction in this way would be a faster and more efficient procedure (since the

randomness due to the optimization scheme is avoided) for obtaining explanations on neural networks and, we conjecture, any other differentiable classifier. Clearly, this would not be a model-agnostic approach, since it requires knowledge of the model gradient and the inverse document frequencies for each example to explain.

We note that this is a somewhat surprising result: without our theoretical analysis and empirical evidence, we would intuitively expect to obtain explanations for this class of models from the gradient reweighted by the input (*i.e.*, the entire vectorization).

We conjecture that, for any differentiable classifier, it is possible to predict the behavior of Anchors by extending the results for linear classifiers, considering a first-order approximation of the model.

6 Conclusion

In this paper, we presented the first theoretical analysis of Anchors. Specifically, we formalized the implementation for textual data, in particular giving insights on the sampling procedure. We then studied Anchors’ behavior on simple if-then rules and linear models. To this end, we introduced an approximate, tractable version of the algorithm, which is close to the default implementation. Our analysis showed that Anchors provides meaningful results when applied to these models, which is supported by experiments with the official implementation. Finally, we exploited our theoretical claim about explainable classifiers to obtain empirical results for neural networks, yielding a surprising result that links the classifier gradient to the importance of words for a prediction. When having access to the model, this result can be used as a faster and more efficient method of obtaining explanations.

This work uncovered some surprising results that emphasize the importance of theoretical analysis in the development of explainability methods. We believe that the insights presented in this article may be valuable for researchers and practitioners in natural language processing who seek to correctly interpret Anchors’ explanations. Furthermore, the analysis framework we developed can aid the explainability community in designing new methods based on sound theoretical foundations and in scrutinizing existing ones. As future work, we plan to extend this analysis to other classes of models, such as CART trees, and to more advanced text vectorizers. We also plan to study Anchors’ behavior on 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.
- Agarwal, C., Zitnik, M., and Lakkaraju, H. (2022). Probing GNN explainers: A rigorous theoretical and empirical analysis of GNN explanation methods. In *International Conference on Artificial Intelligence and Statistics*, pages 8969–8996. PMLR.
- 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.
- Boucheron, S., Lugosi, G., and Massart, P. (2013). *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press.
- 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.
- Danilevsky, M., Qian, K., Aharonov, R., Katsis, Y., Kawas, B., and Sen, P. (2020). A Survey of the State of Explainable AI for Natural Language Processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 447–459.
- Delaunay, J., Galárraga, L., and Largouët, C. (2020). Improving anchor-based explanations. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 3269–3272.
- 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.
- Diaconis, P. and Zabell, S. (1991). Closed form summation for classical distributions: variations on a theme of de Moivre. *Statistical Science*, pages 284–302.
- Esseen, C.-G. (1942). On the Liapunov limit error in the theory of probability. *Ark. Mat. Astr. Fys.*, 28:1–19.
- Fayyad, U. and Irani, K. (1993). Multi-interval discretization of continuous-valued attributes for classification learning.
- Friedman, E. J. (2004). Paths and consistency in additive cost sharing. *International Journal of Game Theory*, 32(4):501–518.
- 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.
- Kaufmann, E. and Kalyanakrishnan, S. (2013). Information complexity in bandit subset selection. In *Conference on Learning Theory*, pages 228–251. PMLR.
- Knoblauch, A. (2008). Closed-form expressions for the moments of the binomial probability distribution. *SIAM Journal on Applied Mathematics*, 69(1):197–204.
- 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.
- 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.
- MacWilliams, F. J. and Sloane, N. J. A. (1977). *The theory of error correcting codes*, volume 16. Elsevier.

- 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.
- Meixner, J. (1934). Orthogonale Polynomsysteme mit einer besonderen Gestalt der erzeugenden Funktion. *Journal of the London Mathematical Society*, 1(1):6–13.
- 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.
- Shevtsova, I. G. (2010). An improvement of convergence rate estimates in the Lyapunov theorem. *Doklady Mathematics*, 82(3):862–864.
- 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.
- Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR.
- Wang, F. and Rudin, C. (2015). Falling rule lists. In *Artificial intelligence and statistics*, pages 1013–1022. PMLR.
- Young, T., Hazarika, D., Poria, S., and Cambria, E. (2018). Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3):55–75.

Appendix for the paper “A Sea of Words: An In-Depth Analysis of Anchors for Text Data”

Organization of the Appendix. We start by providing all proofs of results presented in the paper in Section A. In particular, in Section A.9 we show that all our findings are valid for a normalized TF-IDF vectorization. Section B collects all required technical results. Finally, for further empirical validation of our findings, the interested reader will find additional experiments in Section C.

A PROOFS

In this section, we collect all the proofs omitted from the main paper. In Section A.1 we prove Proposition 1. Sections A.2 and A.3 refer to Section 3. Sections A.4 to A.7 contain the proofs for Section 4. In Section A.8 we provide an additional result to those presented in Section 4.2. In Section A.9 we prove that what was stated for TF-IDF vectorization remains valid for a normalized TF-IDF vectorization.

In all proofs where there is only one tolerance threshold ε and where the selection function p is clear from context, we write \mathcal{A}_k instead of $\mathcal{A}_k^p(\varepsilon)$ for $k \in [3]$.

A.1 Proof of Proposition 1: Equivalent sampling

Let $i \in [n]$ be fixed and let $R \subseteq [b]$ be the (random) set of replaced indices for this specific perturbed example. Let us first compute the probability that a given word with index $k \in [b] \setminus A$ is removed:

$$\begin{aligned}
 \mathbb{P}(k \in R) &= \mathbb{P}(i \in R_k) && \text{(definition of } R_k) \\
 &= \sum_{\ell=0}^n \mathbb{P}(i \in R_k | B_k = \ell) \cdot \mathbb{P}(B_k = \ell) && \text{(law of total probability)} \\
 &= \sum_{\ell=0}^n \binom{n-1}{\ell-1} \binom{n}{\ell}^{-1} \cdot \binom{n}{\ell} \frac{1}{2^n} = \frac{2^{n-1}}{2^n} && \text{(uniform distribution among all subsets)} \\
 \mathbb{P}(k \in R) &= \frac{1}{2}.
 \end{aligned}$$

Let us now show that the removals are independent. First, we notice that the independence column-wise is verified by definition: for each word with index $k \in [b]$, the number B_k of copies to perturb is drawn independently by construction. Next, for a given column $k \in [b]$, let us show that the removal from examples to examples are independent. Let $i, j \in [n]$, with $i \neq j$. We write

$$\begin{aligned}
 \mathbb{P}(i, j \in R_k) &= \sum_{\ell=0}^n \mathbb{P}(i, j \in R_k | B_k = \ell) \mathbb{P}(B_k = \ell) && \text{(law of total probability)} \\
 &= \sum_{\ell=0}^n \binom{n-2}{\ell-2} \binom{n}{\ell}^{-1} \binom{n}{\ell} \frac{1}{2^n} = \frac{2^{n-2}}{2^n} && \text{(uniform distribution among all subsets)} \\
 \mathbb{P}(i, j \in R_k) &= \frac{1}{4}.
 \end{aligned}$$

According to the first part of the proof, this is exactly $\mathbb{P}(i \in R_k) \cdot \mathbb{P}(j \in R_k)$, and we can conclude. \square

A.2 Proof of Proposition 2: Stability of exhaustive p -Anchors

Recall that we set $A^* = A^p(\varepsilon)$. First, let us show that $\mathcal{A}_1^q(\varepsilon - \delta) \subseteq \mathcal{A}_1^p(\varepsilon)$. Let $A \in \mathcal{A}_1^q(\varepsilon - \delta)$. Using Eq. (4), we have

$$p(A) \geq q(A) - \delta \geq 1 - (\varepsilon - \delta) - \delta = 1 - \varepsilon.$$

Therefore, $A \in \mathcal{A}_1^p(\varepsilon)$. We directly deduce that $\mathcal{A}_2^q(\varepsilon - \delta) \subseteq \mathcal{A}_2^p(\varepsilon)$. We now show that $\mathcal{A}_2^q(\varepsilon - \delta)$ is non-empty, and more precisely that it contains A^* . Indeed,

$$q(A^*) \geq p(A^*) - \delta \geq 1 - \varepsilon/4 - \delta \geq 1 - (\varepsilon - \delta),$$

since $\delta < \varepsilon/4 < 3\varepsilon/8$. At this point, it suffices to show that A^* has (strict) maximal q among $\mathcal{A}_2^q(\varepsilon - \delta)$. Let us pick any $A \in \mathcal{A}_2^q(\varepsilon - \delta)$ such that $A \neq A^*$. Then,

$$\begin{aligned} q(A^*) - q(A) &= q(A^*) - p(A^*) + p(A^*) - p(A) + p(A) - q(A) \\ &\geq -\delta + 1 - \varepsilon/4 - (1 - 3\varepsilon/4) - \delta \geq \varepsilon/4 > 0, \end{aligned}$$

since $\delta < \varepsilon/4$. Finally, there is no uniform random draw (last step of Algorithm 1), since $\mathcal{A}_3^p(\varepsilon) = \mathcal{A}_3^q(\varepsilon - \delta) = \{A^*\}$. \square

A.3 Proof of Proposition 3: $\widehat{\text{Prec}}_n(A)$ uniformly approximates Prec

Let $A \in \mathcal{A}$ be any anchor and let x_1, \dots, x_n be perturbed examples associated to this anchor. For all $i \in [n]$, the random variables $Y_i := \mathbf{1}_{f(x_i)=1} \in \{0, 1\}$ are independent and bounded by construction. Since $\text{Prec}(A) = \mathbb{E} \left[\widehat{\text{Prec}}_n(A) \right]$, we can apply Hoeffding's inequality to the Y_i s (Boucheron et al., 2013, Theorem 2.8). We obtain

$$\mathbb{P} \left(\left| \widehat{\text{Prec}}_n(A) - \text{Prec}(A) \right| > \delta \right) = \mathbb{P} \left(n \left| \widehat{\text{Prec}}_n(A) - \text{Prec}(A) \right| > n\delta \right) \leq 2e^{-2n\delta^2}. \quad (11)$$

There are less than 2^b anchors (since we are not considering the empty anchor as a valid anchor), and we can conclude *via* a union bound argument. \square

A.4 Proof of Proposition 4: Dummy features

Let $A \in \mathcal{A}_3^{\text{Prec}}(\varepsilon)$. If $a_j = 0$, there is nothing to prove. Thus let us assume that $a_j > 0$ and come to a contradiction. Let us set A^j the anchor identical to A but with coordinate j to zero. The precision of A is given by

$$\text{Prec}(A) = \mathbb{P}_A (g(\varphi(x)) \in R) = \mathbb{P}_A (g(\dots, \varphi(x)_j, \dots) \in R).$$

According to the discussion preceding Proposition 4, $\varphi(x)_j = M_j v_j$, where $M_j \sim a_j + B_j$, with $B_j \sim \text{Bin}(m_j - a_j, 1/2)$ (this is Eq. (6)). Since g does not depend on coordinate j and the sampling is independent, $g(\dots, v_j(a_j + B_j), \dots)$ is equal in distribution to $g(\dots, v_j \text{Bin}(m_j, 1/2), \dots)$. In particular, $\text{Prec}(A^j) = \text{Prec}(A)$. Since $|A^j| < |A|$, we can conclude. \square

A.5 Proof of Proposition 5: Presence of a set of words

Let $p_i := 1 - \frac{1}{2^{m_i}}$. Since the model only depends on the coordinates belonging to J , according to Proposition 4, we can restrict ourselves to anchors such that $a_j \neq 0$ if $j \in J$. Let us start by computing the precision of any candidate anchor

$A \in \mathcal{A}$. We write

$$\begin{aligned}
 \text{Prec}(A) &= \mathbb{E}_A [\mathbf{1}_{f(x)=f(\xi)}] \\
 &= \mathbb{P}_A (f(x) = 1) && \text{(since } f(\xi) = 1) \\
 &= \mathbb{P}_A (w_j \in x, \forall j \in J) \\
 &= \prod_{j \in J} \mathbb{P}_A (w_j \in x) && \text{(by independence)} \\
 &= \prod_{j \in J} \mathbb{P}_A (v_j M_j > 0) \\
 &= \prod_{j \in J} \mathbb{P}_A (a_j + B_j > 0) \\
 \text{Prec}(A) &= \prod_{j \in J} p_j^{\mathbf{1}_{a_j=0}}.
 \end{aligned}$$

Let us now apply Algorithm 1 step by step in each of the cases outlined in the statement of the result.

Case (I): $\max_{j \in J} m_j \leq B$. If for all $j \in J$, $a_j > 0$, then, according to the previous discussion,

$$\text{Prec}(A) = \prod_{j \in J} p_j^{\mathbf{1}_{a_j=0}} = 1.$$

Therefore, the anchor $(1, \dots, 1)$ belongs to \mathcal{A}_1 . If, instead, there exists $j \in J$ such that $a_j = 0$, then $\sum_{j \in J} \mathbf{1}_{a_j=0} \geq 1$ and

$$\begin{aligned}
 \text{Prec}(A) &= \prod_{j \in J} p_j^{\mathbf{1}_{a_j=0}} \\
 &\leq \prod_{j \in J} \left(1 - \frac{1}{2^B}\right)^{\mathbf{1}_{a_j=0}} && \text{(since } m_j \leq B \text{ for all } j \in J) \\
 &= \left(1 - \frac{1}{2^B}\right)^{\sum_{j \in J} \mathbf{1}_{a_j=0}} && \text{(since } \sum_{j \in J} \mathbf{1}_{a_j=0} \geq 1) \\
 &\leq 1 - \frac{1}{2^B} \\
 \text{Prec}(A) &< 1 - \varepsilon. && \text{(since } 1 - \frac{1}{2^B} \leq 1 - \varepsilon \text{ by definition of } B)
 \end{aligned}$$

Thus \mathcal{A}_1 consists of anchors having at least one occurrence of each word of J , and these anchors only. In \mathcal{A}_2 , Algorithm 1 will select the anchor $A_J = \{w_j, j \in J\}$, such that $a_j = 1$ for all $j \in J$ and $a_j = 0$ if $j \notin J$, which is the shortest anchor satisfying the precision condition. Since there are no equality cases, \mathcal{A}_3 is a singleton and we can conclude.

Case (II): $\max_{j \in J} m_j > B$. We first make two claims:

Claim 1. We can restrict our analysis to anchors $A \in \mathcal{A}$ such that $a_j \in \{0, 1\}$ for $j \in J$.

Indeed, for any $j \in J$, the model f is checking the presence of the word w_j in a document δ , i.e., that $\varphi(\delta)_j > 0$, disregarding of its multiplicity. As said before, the anchor A_J (such that $a_j = 1$ for all $j \in J$ and $a_j = 0$ if $j \notin J$) has precision 1. Any other anchor $A = (a_1, \dots, a_d)$ such that $a_i \geq 2$ has the same precision, but higher length.

Now let us consider two indices j and j' in J such that $j > j'$ (implying, $p_j < p_{j'}$) and an anchor A such that $a_j = a_{j'} = 0$. We set A^j (resp. $A^{j'}$) the anchor identical to A except coordinate j (resp. j') put to 1. Let $I_A := \{j \in [d], a_j > 0\}$.

Claim 2. If $j > j'$, $\text{Prec}(A^j) > \text{Prec}(A^{j'})$.

a_1	a_2	\dots	a_{k-c_0}	a_{k-c_0+1}	a_{k-c_0+2}	\dots	a_{k-1}	a_k	$\text{Prec}(A) = \prod_{j \in J} p_j^{1_{a_j=0}} = \prod_{\ell=1}^k p_\ell^{1_{a_\ell=0}}$
0	0	\dots	0	0	0	\dots	0	0	$\prod_{j \in J} p_j < 1 - \varepsilon$
0	0	\dots	0	0	0	\dots	0	1	$\prod_{\ell=1}^{k-1} p_\ell < 1 - \varepsilon$
0	0	\dots	0	0	0	\dots	1	1	$\prod_{\ell=1}^{k-2} p_\ell < 1 - \varepsilon$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	1	1	$\text{Prec}(A) < 1 - \varepsilon$
0	0	\dots	0	0	1	\dots	1	1	$\prod_{\ell=1}^{k-c_0+1} p_\ell < 1 - \varepsilon$
0	0	\dots	0	1	1	\dots	1	1	$\prod_{\ell=1}^{k-c_0} p_\ell \geq 1 - \varepsilon$
0	0	\dots	1	1	1	\dots	1	1	$\prod_{\ell=1}^{k-c_0-1} p_\ell > 1 - \varepsilon$
\vdots	\vdots	1	1	1	1	1	1	1	$\text{Prec}(A) > 1 - \varepsilon$
1	1	1	1	1	1	1	1	1	$1 > 1 - \varepsilon$

Figure 6: Illustration of Proposition 5. Anchors extraction for a model classifying words according to the presence or absence of words w_j , $j \in J = \{1, \dots, k\}$, ranked such that $m_1 > m_2 > \dots > m_k$. The anchor such that $a_\ell = 1$ for $\ell \in \llbracket k - c_0 + 1, k \rrbracket$ and $a_\ell = 0$ otherwise, is the minimal anchor satisfying the precision condition.

Indeed,

$$\begin{aligned}
 \text{Prec}(A^j) &= \prod_{\ell \in J, a_\ell=0} p_\ell \\
 &= \prod_{\ell \in J \setminus (I_A \cup \{j\})} p_\ell \\
 &= p_{j'} \cdot \prod_{\ell \in J \setminus (I_A \cup \{j\} \cup \{j'\})} p_\ell \\
 &> p_j \cdot \prod_{\ell \in J \setminus (I_A \cup \{j\} \cup \{j'\})} p_\ell \quad (\text{since } p_j < p_{j'}) \\
 \text{Prec}(A^j) &= \text{Prec}(A^{j'}).
 \end{aligned}$$

As a consequence, for any anchor of fixed length, we can get higher precision by moving indices to the right. Therefore, the anchor A_{c_0} will be selected by Algorithm 1, see Figure 6 for an illustration. \square

A.6 Proof of Proposition 6: Precision of a linear classifier

Let us set

$$Z_d := \lambda_0 + \sum_{j=1}^d \lambda_j v_j M_j,$$

where $M_j = a_j + B_j$ are the random multiplicities, that is, a_j is the number of anchored words for j and, as before, $B_j \sim \text{Bin}(m_j - a_j, 1/2)$. In our notation, the problem of evaluating the precision of an anchor A is that of evaluating

accurately $\mathbb{P}(Z_d > 0)$. From Section B.1, we see that, for all $j \in [d]$,

$$\mathbb{E}[M_j] = \frac{1}{2}(m_j + a_j), \quad \text{and} \quad \text{Var}(M_j) = \frac{1}{4}(m_j - a_j).$$

We deduce that

$$\mathbb{E}[Z_d] = \frac{1}{2} \sum_{j=1}^d \lambda_j v_j (m_j + a_j) \quad \text{and} \quad \text{Var}(Z_d) = \frac{1}{4} \sum_{j=1}^d \lambda_j^2 v_j^2 (m_j - a_j). \quad (12)$$

By the Berry-Esseen theorem for non-identically distributed version (Shevtsova, 2010), uniformly in $t \in \mathbb{R}$, it holds that

$$\left| \mathbb{P}(Z_d \leq t) - \Phi\left(\frac{t - \mathbb{E}[Z_d]}{\sqrt{\text{Var}(Z_d)}}\right) \right| \leq C \cdot \frac{\sum_{j=1}^d \mathbb{E}\left[|\lambda_j v_j M_j - \mathbb{E}[\lambda_j v_j M_j]|^3\right]}{\text{Var}(Z_d)^{3/2}}, \quad (13)$$

where $C \approx 7.15$ is a numerical constant. Setting $t = -\lambda_0$ in the previous display, we recognize $1 -$ the definition of the precision. Using Eq. (12), we have obtained the left-hand side of Eq. (9). The numerator is upper bounded by first writing

$$\begin{aligned} \mathbb{E}\left[|\lambda_j v_j M_j - \mathbb{E}[\lambda_j v_j M_j]|^3\right] &\leq \max_j |\lambda_j v_j|^3 \cdot \mathbb{E}\left[|B_j - \mathbb{E}[B_j]|^3\right] && \text{(definition of } M_j) \\ &\leq \max_j |\lambda_j v_j|^3 \cdot \frac{1}{\sqrt{8\pi}} \left(\frac{m_j - a_j}{2}\right)^{3/2} && \text{(Eq. (21))} \\ \mathbb{E}\left[|\lambda_j v_j M_j - \mathbb{E}[\lambda_j v_j M_j]|^3\right] &\leq \frac{1}{\sqrt{\pi}} \cdot \max_j |\lambda_j v_j|^3 \cdot (\max_j m_j)^{3/2}. \end{aligned}$$

We deduce that

$$\sum_{j=1}^d \mathbb{E}\left[|\lambda_j v_j M_j - \mathbb{E}[\lambda_j v_j M_j]|^3\right] \leq \frac{1}{\sqrt{\pi}} \cdot \left(\max_j \lambda_j^2 v_j^2\right)^{3/2} \cdot (\max_j m_j)^{3/2} \cdot d. \quad (14)$$

Regarding the denominator, we have

$$\begin{aligned} \text{Var}(Z_d) &= \frac{1}{4} \sum_{j=1}^d \lambda_j^2 v_j^2 (m_j - a_j) && \text{(Eq. (12))} \\ &\geq \frac{1}{4} \cdot \left(\min_j \lambda_j^2 v_j^2\right) \cdot (b - |A|) && \text{(definition of } b \text{ and } |A|) \\ \text{Var}(Z_d) &\geq \frac{1}{8} \cdot \left(\min_j \lambda_j^2 v_j^2\right) \cdot \min_j m_j \cdot d. \end{aligned}$$

Thus

$$\text{Var}(Z_d)^{3/2} \geq \frac{1}{16\sqrt{2}} \cdot \left(\min_j \lambda_j^2 v_j^2\right)^{3/2} \cdot \left(\min_j m_j\right)^{3/2} \cdot d^{3/2}. \quad (15)$$

In particular, this is a positive quantity. Coming back to Eq. (13), we see that

$$\left| \mathbb{P}(Z_d \leq t) - \Phi\left(\frac{t - \mathbb{E}[Z_d]}{\sqrt{\text{Var}(Z_d)}}\right) \right| \leq \frac{16C\sqrt{2}}{\sqrt{\pi}} \cdot \left(\frac{\max_j \lambda_j^2 v_j^2}{\min_j \lambda_j^2 v_j^2}\right)^{3/2} \cdot \left(\frac{\max_j m_j}{\min_j m_j}\right)^{3/2} \cdot \frac{1}{\sqrt{d}},$$

as announced. \square

A.7 Proof of Proposition 7: Approximate precision maximization

We start by proving two lemmas. The first shows that we can restrict ourselves to \mathcal{A}_+ when considering the minimization of L .

Lemma 1 (Restriction to positive anchors). *Let $A \in \mathcal{A}$ be such that $a_j > 0$ whereas $\lambda_j < 0$. Then*

$$L(\dots, 0, \dots) < L(\dots, a_j, \dots).$$

Proof. Keeping in mind that $\lambda_j > 0$, we notice that $-\lambda_j v_j (m_j + a_j) > -\lambda_j v_j m_j$, and that $\lambda_j^2 v_j^2 (m_j - a_j) < \lambda_j^2 v_j^2 m_j$. In other terms, removing the word w_j from the anchor A both decreases the numerator and increases the numerator of L . \square

The second shows that the minimization of L on \mathcal{A}_+ is straightforward, modulo a technical assumption on the size of the intercept.

Lemma 2 (Minimization of L). *Assume that $\lambda_0 > -\gamma/2$. Assume further that the indices of the local dictionary are ordered such that the $\lambda_j v_j$ s are strictly decreasing. Then, for any $k < \ell$ such that $\lambda_k, \lambda_\ell > 0$,*

$$L(a_1, \dots, a_k + 1, \dots, a_\ell, \dots, a_d) < L(a_1, \dots, a_k, \dots, a_\ell + 1, \dots, a_d). \quad (16)$$

Proof. First, since $A \in \mathcal{A}_+$ and $\lambda_0 \geq -\gamma/2$, we deduce that

$$\lambda_0 + \frac{1}{2} \sum_j \lambda_j v_j m_j + \frac{1}{2} \sum_j \lambda_j v_j a_j \geq 0. \quad (17)$$

Now, we will prove both inequalities by a function study. For any $j \in [d]$, let us set $\alpha_j := \lambda_j v_j$. We also set $\Omega_1 := \sum_j \alpha_j (m_j - a_j)$, and $\Omega_2 := \sum_j \alpha_j^2 (m_j - a_j)$. Further, for any $a, b \in \mathbb{R}$, let us define the mapping

$$f_{a,b}(t) := \frac{a+t}{\sqrt{b-t^2}}.$$

With this notation in hand, Eq. (16) becomes

$$\frac{\gamma - \frac{1}{2}(\Omega_1 - \alpha_k)}{\sqrt{\Omega_2 - \alpha_k^2}} > \frac{\gamma - \frac{1}{2}(\Omega_1 - \alpha_\ell)}{\sqrt{\Omega_2 - \alpha_\ell^2}}$$

which is simply

$$f_{2\gamma - \Omega_1, \Omega_2}(\alpha_k) > f_{2\gamma - \Omega_1, \Omega_2}(\alpha_\ell).$$

Observe that, by our assumptions, $2\gamma - \Omega_1 > 0$, and $\Omega_2 > 0$. It is straightforward to show that $f'_{a,b}(t) = \frac{at+b}{(b-t^2)^{3/2}}$, and therefore $f_{2\gamma - \Omega_1, \Omega_2}(\alpha_k)$ is a *strictly increasing* mapping on $[0, \sqrt{b}]$. Since $\sqrt{\Omega_2} \geq \alpha_k > \alpha_\ell > 0$, we can conclude. \square

Proof of Proposition 7. In this proof, we set $p = \bar{\Phi} \circ L$. Let A^F be the anchor containing all words of ξ . In our notation,

$$A^F = (m_1, \dots, m_d).$$

We first notice that \mathcal{A}_1^p is non-empty since $\bar{\Phi}(L(A^F)) = 1$. By construction, \mathcal{A}_2^p , consisting of anchors of \mathcal{A}_1^p of minimal length, is non-empty as well. Lemma 1 ensures that \mathcal{A}_3^p , the anchors of \mathcal{A}_2^p with the highest p value, is a non-empty subset of \mathcal{A}_+ . Indeed, one can remove the anchors corresponding to the indices $j \in [d]$ such that $\lambda_j v_j < 0$, and increase the value of p . Since we assumed that at least one λ_j is positive, it is always possible to do this removal. Finally, let ℓ be the common length of the anchors belonging to \mathcal{A}_3^p . Since we satisfy the assumptions of Lemma 2, we see that the p value of any anchor of length ℓ is *strictly increasing* if we swap indices towards the lower indices. We deduce the result. \square

A.8 Additional result for Section 4.2: Simple if-then rules

We present here a further result on a simple if-then classifier based on the presence of disjoint subsets of words.

Proposition 8 (Small decision tree). *Let the (binary) classifier f be defined as follows:*

$$f(z) = \mathbf{1}_{(w_1 \in z \text{ and } w_2 \in z) \text{ or } w_3 \in z}$$

Then, for any $\varepsilon > 0$, the anchor $A = (0, 0, 1)$, will be selected by exhaustive Anchors.

For example, consider the sentiment analysis task, and f the model returning 1 (a positive prediction) if words “not” and “bad” or the word “good” are present in the document. Proposition 8 implies that only the word “good” will be selected as an anchor. This is a satisfying property and corresponds to the intuition that we have from Anchors: in this class of examples, **the smallest rule is provably selected by Anchors**. We prove Proposition 8 in Section A.8 of the Appendix.

Of course, the scope of Proposition 8 is limited. It is possible to obtain similar results for other simple sets of rules, though challenging to present these results with a sufficient amount of generality.

Proof of Proposition 8 Let us start by computing $\text{Prec}(A)$ for any candidate anchor $A \in \mathcal{A}$. Since the model only depends on the first three coordinates, according to Proposition 4, we can restrict ourselves to $A = (a_1, a_2, a_3)$. In this proof, we set $p_i := 1 - \frac{1}{2^{m_i}}$ the probability of keeping the word w_i while sampling and $B_k \sim \text{Bin}(m_j - a_j, 1/2)$. The precision of a candidate anchor $A \in \mathcal{A}$ (Eq. (2)) associated to f is

$$\begin{aligned} \text{Prec}(A) &= \mathbb{E}_A [\mathbf{1}_{f(x)=f(\xi)}] \\ &= \mathbb{P}_A (f(x) = 1) \\ &= \mathbb{P}_A (\varphi(x)_1 > 0) \cdot \mathbb{P}_A (\varphi(x)_2 > 0) \cdot (1 - \mathbb{P}_A (\varphi(x)_3 > 0)) + \mathbb{P}_A (\varphi(x)_3 > 0) \quad (\text{by independence}) \\ &= \mathbb{P}_A (M_1 > 0) \cdot \mathbb{P}_A (M_2 > 0) \cdot (1 - \mathbb{P}_A (M_3 > 0)) + \mathbb{P}_A (M_3 > 0) \quad (\text{since } v_j > 0 \text{ for all } j \in [d]) \\ &= \mathbb{P}_A (a_1 + B_1 > 0) \cdot \mathbb{P}_A (a_2 + B_2 > 0) \cdot (1 - \mathbb{P}_A (a_3 + B_3 > 0)) + \mathbb{P}_A (a_3 + B_3 > 0) \\ &= \begin{cases} 1, & \text{if } a_3 > 0, \\ 1, & \text{if } a_1, a_2 > 0 \text{ and } a_3 = 0, \\ p_2(1 - p_3) + p_3 = 1 - \frac{1}{2^{m_2+m_3}}, & \text{if } a_1 > 0 \text{ and } a_2, a_3 = 0, \\ p_1(1 - p_3) + p_3 = 1 - \frac{1}{2^{m_1+m_3}}, & \text{if } a_2 > 0 \text{ and } a_1, a_3 = 0, \\ p_1 p_2 (1 - p_3) + p_3 = 1 - \frac{2^{m_1} + 2^{m_2} - 1}{2^{m_1+m_2+m_3}}, & \text{if } a_1 = a_2 = a_3 = 0. \end{cases} \end{aligned}$$

Now let us follow Algorithm 1 step by step. According to the previous discussion, any anchor such that a_1 and a_2 are positive and/or $a_3 > 0$ has precision 1, and thus belongs to \mathcal{A}_1 . In particular, the anchor $(0, 0, 1)$ belongs to \mathcal{A}_1 . We note that it also has minimal length 1, and therefore belongs to \mathcal{A}_2 . Finally, any other anchor with the same length will have a smaller precision, since $p_2(1 - p_3) + p_3 < 1$, $p_1(1 - p_3) + p_3 < 1$, and $p_1 p_2 (1 - p_3) + p_3 < 1$. In conclusion, \mathcal{A}_3 is reduced to a singleton and the anchor $A = (0, 0, 1)$ will be selected by exhaustive Anchors. \square

A.9 Normalized TF-IDF

In this section we show that our theoretical results demonstrated considering a TF-IDF vectorization as defined in Definition 1 still hold for the ℓ_2 -normalized TF-IDF vectorization, defined as

$$\forall j \in [D], \quad \phi(z)_j := \frac{m_j(z)v_j}{\sqrt{\sum_{j=1}^D m_j(z)^2 v_j^2}},$$

that is, the default normalization in the `scikit-learn` implementation of TF-IDF. The main result of this section is the following:

Proposition 9 (Normalized-TF-IDF, Berry-Esseen). *Assume that $0 < v_{\min} \leq v_j \leq v_{\max}$ and $m \leq m_j \leq M$ for all $j \in [d]$. Assume further that A is not the empty anchor, that $|A| \leq b/2$ and that $a_j < m_j$ for all j . Finally, assume that $\|\lambda\| = 1$ as $d \rightarrow +\infty$. For all $t \in \mathbb{R}$, define*

$$P(t) := \Phi \left(\frac{t \sqrt{\sum_j \{(m_j + a_j)^2 + m_j - a_j\} v_j^2 - \sum_j \lambda_j (m_j + a_j) v_j}}{\sqrt{\sum_j \lambda_j^2 (m_j - a_j) v_j^2}} \right).$$

Then

$$\left| \mathbb{P}(\lambda^\top \phi(x) \leq t) - P(t) \right| \leq \frac{2\sqrt{M}v_{\min}^{-3/2}}{d^{3/4-\varepsilon}} + 2\exp \left(-\frac{d^{1+2\varepsilon}v_{\min}^6}{4v_{\max}^4 M^4} \right). \quad (18)$$

As a direct consequence of Proposition 9, we know that a good approximation of $\text{Prec}(A)$ in the normalized TF-IDF case is $\bar{\Phi}(L(A))$, with

$$L(A) = \frac{-\lambda_0 \sqrt{\sum_j \{(m_j + a_j)^2 + m_j - a_j\} v_j^2 - \sum_j \lambda_j (m_j + a_j) v_j}}{\sqrt{\sum_j \lambda_j^2 (m_j - a_j) v_j^2}}. \quad (19)$$

This is reminiscent of Eq. (9) in the non-normalized case. When $\lambda_0 = 0$, the analysis of the maximization problem is a subcase of the non-normalized case, and we recover the same result. Although $\lambda_0 = 0$ can be a reasonable assumption (assuming centered data and no intercept), we conjecture that the result is true for a larger range of λ_0 , similarly to the unnormalized case.

Let us now prove Proposition 9. Let us set

$$Z_d := \frac{\sum_{j=1}^d \lambda_j v_j M_j}{\sqrt{\sum_{j=1}^d v_j^2 M_j^2}} \quad \text{and} \quad \tilde{Z}_d := \frac{\sum_{j=1}^d \lambda_j v_j M_j}{\sqrt{\frac{1}{4} \sum_{j=1}^d \{(m_j + a_j)^2 + m_j - a_j\}}}.$$

Intuitively, when d is large enough, both these quantities are close with high probability, and \tilde{Z}_d has the same structure as the linear form studied in the normalized case, up to a constant. Thus, the analysis boils down to the previous case, modulo the following:

Proposition 10 (Z_d and \tilde{Z}_d are close with high probability). *Let $\varepsilon \in (0, 1/2)$. Assume that $0 < v_{\min} \leq v_j \leq v_{\max}$ and $m \leq m_j \leq M$ for all $j \in [d]$. Assume further that A is not the empty anchor. Finally, assume that $\|\lambda\| = 1$ as $d \rightarrow +\infty$. Then*

$$\mathbb{P}\left(\left|Z_d - \tilde{Z}_d\right| > \frac{c}{d^{1/2-\varepsilon}}\right) \leq 2\exp\left(-\frac{c^2 d^{1+2\varepsilon} v_{\min}^6}{4v_{\max}^4 M^4}\right),$$

for any small positive constant c .

Proof of Proposition 10. In this proof, we write $D := \sum_{j=1}^d v_j^2 M_j^2$. We begin by computing the expectation of D . We know that $M_j = a_j + B_j$, where $B_j \sim \text{Bin}(m_j - a_j, 1/2)$. Therefore,

$$\begin{aligned} \mathbb{E}[M_j^2] &= \mathbb{E}[a_j^2 + 2a_j B_j + B_j^2] \\ &= a_j^2 + a_j(m_j - a_j) + \frac{1}{4}(m_j - a_j)^2 + \frac{1}{4}(m_j - a_j) \\ \mathbb{E}[M_j^2] &= \frac{1}{4}(m_j + a_j)^2 + \frac{1}{4}(m_j - a_j), \end{aligned}$$

where we used Lemma 6 to compute $\mathbb{E}[B_j^2]$. By linearity, we deduce that

$$\mathbb{E}[D] = \frac{1}{4} \sum_{j=1}^d \{(m_j + a_j)^2 + m_j - a_j\}.$$

Note that, with this notation in hand,

$$Z_d = \frac{\sum_{j=1}^d \lambda_j M_j v_j}{\sqrt{D}} \quad \text{and} \quad \tilde{Z}_d = \frac{\sum_{j=1}^d \lambda_j M_j v_j}{\sqrt{\mathbb{E}[D]}}.$$

We need to prove the following, which shows that D is concentrated around its expectation:

Lemma 3 (Concentration of D). *Assume that $0 < v_{\min} \leq v_j \leq v_{\max}$ and that $m \leq m_j \leq M$ for all $j \in [d]$. Then, for all $t > 0$,*

$$\mathbb{P}(|D - \mathbb{E}[D]| > t) \leq 2\exp\left(\frac{-2t^2}{dv_{\max}^4 M^4}\right).$$

Proof. This is a straightforward application of Hoeffding's inequality once we notice that the random variables $v_j^2 M_j^2$ are bounded and independent, and that $\sum_j m_j^4 v_j^4 \leq dv_{\max}^4 M^4$ under our assumptions. \square

Note that Lemma 3 is tight, since Hoeffding's inequality is tight for Bernoulli random variables, a case which is possible under our assumption. Lemma 3 allows controlling the small deviations of D , a fact that we will maybe not use in the following, but can nonetheless be useful to split a complicated event. Next, we control the size of D .

Lemma 4 (D is small with high probability). *Assume that $0 < v_{\min} \leq v_j \leq v_{\max}$ for all $j \in [d]$. Then*

$$\mathbb{P}\left(D < \frac{1}{4} dv_{\min}^2\right) \leq 2\exp\left(\frac{-dv_{\min}^4}{8v_{\max}^4 M^4}\right).$$

Proof. We write

$$\begin{aligned} \mathbb{P}\left(D < \frac{1}{4}dv_{\min}^2\right) &= \mathbb{P}\left(D - \mathbb{E}[D] < \frac{1}{4}dv_{\min}^2 - \mathbb{E}[D]\right) \\ &\leq \mathbb{P}\left(|D - \mathbb{E}[D]| < \frac{1}{2}dv_{\min}^2 - \frac{1}{4}dv_{\min}^2\right), \end{aligned}$$

since $\frac{1}{4}\{(m_j + a_j)^2 + m_j - a_j\} \geq \frac{1}{4} \cdot (1^2 + 1)$ for all $j \in [d]$. We conclude by applying Lemma 3 with $t = \frac{1}{4}dv_{\min}^2$. \square

Now we can control the key quantity:

Lemma 5 (Control of the key quantity). *Assume that $0 < v_{\min} \leq v_j \leq v_{\max}$ and $m \leq m_j \leq M$ for all $j \in [d]$. Assume further that A is not the empty anchor. Then, for any $t > 0$,*

$$\mathbb{P}\left(\left|1 - \frac{\sqrt{D}}{\sqrt{\mathbb{E}[D]}}\right| > t\right) \leq 2\exp\left(\frac{-d^2t^2v_{\min}^6}{4v_{\max}^4M^4}\right).$$

In particular, by taking t of the order $\frac{1}{d^{1/2-\varepsilon}}$ for some $\varepsilon > 0$, we see that $\mathbb{P}\left(\left|1 - \sqrt{D/\mathbb{E}[D]}\right| > t\right) = o(1)$.

Proof. Multiplying by $\mathbb{E}[D]$, we see that we want to control

$$\mathbb{P}\left(\left|\sqrt{D} - \sqrt{\mathbb{E}[D]}\right| > t\sqrt{\mathbb{E}[D]}\right).$$

Since $\mathbb{E}[D] \leq \frac{1}{2}dv_{\min}^2$, we can simply control

$$\mathbb{P}\left(\left|\sqrt{D} - \sqrt{\mathbb{E}[D]}\right| > \frac{1}{2}d^2tv_{\min}^2\right).$$

Additionally, since A is not the empty anchor, $D \geq v_{\min}^2$ almost surely, which is positive. Since the mapping $x \mapsto \sqrt{x}$ is $1/2\sqrt{C}$ -Lipschitz on $[C, +\infty)$, we see that

$$\left|\sqrt{D} - \sqrt{\mathbb{E}[D]}\right| \leq \frac{1}{2v_{\min}} |D - \mathbb{E}[D]|,$$

which allows us to focus on

$$\mathbb{P}\left(|D - \mathbb{E}[D]| > dtv_{\min}^3\right).$$

We control this last display using Lemma 3, and we obtain

$$\mathbb{P}\left(\left|1 - \frac{\sqrt{D}}{\sqrt{\mathbb{E}[D]}}\right| > t\right) \leq 2\exp\left(\frac{-d^2t^2v_{\min}^6}{4v_{\max}^4M^4}\right),$$

as promised. \square

Finally, coming back to the original problem, we write

$$\mathbb{P}\left(|Z_d - \tilde{Z}_d| > s\right) = \mathbb{P}\left(|Z_d| \cdot \left|1 - \frac{\sqrt{D}}{\sqrt{\mathbb{E}[D]}}\right| > s\right) \leq \mathbb{P}\left(\left|1 - \frac{\sqrt{D}}{\sqrt{\mathbb{E}[D]}}\right| > s\right). \quad (20)$$

By Cauchy-Schwarz inequality, we have

$$\left|\sum_j \lambda_j M_j v_j\right| \leq \|\lambda\| \cdot \sqrt{D},$$

and we deduce that $|Z_d| \leq \|\lambda\| = 1$ under our assumptions. Coming back to Eq. (20), we can therefore take $s = \frac{1}{d^{1/2-\varepsilon}}$ and use Lemma 5 to conclude. \square

We can now conclude this section with the proof of our main result.

Proof of Proposition 9. Let us set $N := \sum_j \lambda_j M_j v_j$. With this notation, $\mathbb{P}(\lambda^\top \phi(x) \leq t) = \mathbb{P}(Z_d \leq t)$, and

$$\lambda^\top \phi(x) = \frac{N}{\sqrt{D}} = Z_d \quad \text{and} \quad \tilde{Z}_d = \frac{N}{\sqrt{\mathbb{E}[D]}}.$$

Let $s > 0$. Using Lemma 9, we have

$$\mathbb{P}(\tilde{Z}_d \leq t - s) - \mathbb{P}(|Z_d - \tilde{Z}_d| > s) \leq \mathbb{P}(Z_d \leq t) \leq \mathbb{P}(\tilde{Z}_d \leq t + s) + \mathbb{P}(|Z_d - \tilde{Z}_d| > s).$$

Let us set $s = \frac{1}{d^{1/2-\varepsilon}}$ for some small $\varepsilon > 0$. By Proposition 10, we know that $\mathbb{P}(|Z_d - \tilde{Z}_d| > s) \leq 2\exp\left(-\frac{d^{1+2\varepsilon} v_{\min}^6}{4v_{\max}^4 M^4}\right)$.

Let us now turn towards the remaining terms, depending on \tilde{Z} . We write, for any $u \in \mathbb{R}$,

$$\begin{aligned} \mathbb{P}(\tilde{Z}_d \leq u) &= \mathbb{P}\left(\frac{N}{\sqrt{\mathbb{E}[D]}} \leq u\right) \\ &= \mathbb{P}\left(\frac{N - \mathbb{E}[N]}{\sqrt{\text{Var}(N)}} \leq \frac{u\sqrt{\mathbb{E}[D]} - \mathbb{E}[N]}{\sqrt{\text{Var}(N)}}\right) \\ \mathbb{P}(\tilde{Z}_d \leq u) &= \Phi\left(\frac{u\sqrt{\mathbb{E}[D]} - \mathbb{E}[N]}{\sqrt{\text{Var}(N)}}\right) + o(1), \end{aligned}$$

uniformly in u , where we used Proposition 6 in the last derivation.

Since Φ is 1-Lipschitz, we see that

$$\left| \mathbb{P}(\tilde{Z}_d \leq t + s) - \Phi\left(\frac{t\sqrt{\mathbb{E}[D]} - \mathbb{E}[N]}{\sqrt{\text{Var}(N)}}\right) \right| \leq s \frac{\sqrt{\mathbb{E}[D]}}{\sqrt{\text{Var}(N)}} + 2\exp\left(-\frac{d^{1+2\varepsilon} v_{\min}^6}{4v_{\max}^4 M^4}\right).$$

Moreover, under our assumptions, $\sqrt{\mathbb{E}[D]}/\sqrt{\text{Var}(N)} = \mathcal{O}(1)$ and $s = \frac{1}{d^{1/2-\varepsilon}}$. Therefore,

$$\mathbb{P}(\tilde{Z}_d \leq t + s) = \Phi\left(\frac{t\sqrt{\mathbb{E}[D]} - \mathbb{E}[N]}{\sqrt{\text{Var}(N)}}\right) + \frac{1}{d^{1/2-\varepsilon}} \frac{\sqrt{\mathbb{E}[D]}}{\sqrt{\text{Var}(N)}} + 2\exp\left(-\frac{d^{1+2\varepsilon} v_{\min}^6}{4v_{\max}^4 M^4}\right).$$

Additionally, one can show that $\text{Var}(M_j^2) = \nu(m_j - a_j)$, with

$$\nu(x) := \frac{x}{8}(2(2a+x)^2 + x - 1).$$

It is straightforward to show that ν is non-decreasing. When $m_j > a_j$, we see that

$$\nu(m_j - a_j) \geq \nu(1) = a_j^2 + a_j + \frac{1}{4} \geq \frac{1}{4}.$$

Therefore $\sum_j v_j^4 \text{Var}(M_j^2) \geq \frac{d}{4} v_{\min}^4$. Under our assumptions, $\|\lambda\| = 1$, and by applying Cauchy-Schwarz inequality $\mathbb{E}[D] \leq \sqrt{d} v_{\max} M$, we deduce

$$\frac{\mathbb{E}[D]}{\text{Var}(N)} \leq \frac{4v_{\max} M}{\sqrt{d} v_{\min}^4}.$$

Thus, we find that

$$\begin{aligned} \mathbb{P}(\tilde{Z}_d \leq t + s) &\leq \Phi\left(\frac{t\sqrt{\mathbb{E}[D]} - \mathbb{E}[N]}{\sqrt{\text{Var}(N)}}\right) + \frac{1}{d^{1/2-\varepsilon}} \frac{2\sqrt{M}}{d^{1/4} v_{\min}^{3/2}} + 2\exp\left(-\frac{d^{1+2\varepsilon} v_{\min}^6}{4v_{\max}^4 M^4}\right) \\ &= \Phi\left(\frac{t\sqrt{\mathbb{E}[D]} - \mathbb{E}[N]}{\sqrt{\text{Var}(N)}}\right) + \frac{2\sqrt{M}}{d^{3/4-\varepsilon} v_{\min}^{3/2}} + 2\exp\left(-\frac{d^{1+2\varepsilon} v_{\min}^6}{4v_{\max}^4 M^4}\right). \end{aligned}$$

The same reasoning applies to $t - s$, and we can conclude by recognizing $\Phi\left(\frac{t\sqrt{\mathbb{E}[D]} - \mathbb{E}[N]}{\sqrt{\text{Var}(N)}}\right)$ as $P(t)$.

□

B TECHNICAL RESULTS

We present here some technical results that were used in our analysis regarding binomial random variables (Section B.1) and two additional lemmas (Section B.2).

B.1 Binomial wonderland

In this section, we collect some facts about binomial random variables. We focus on the case $p = 1/2$ because of the sampling scheme of Anchors, with a few exceptions. We start with straightforward moment computations, which are stated here for completeness' sake.

Lemma 6 (Moments of the binomial distribution). *Let $m \geq 1$ be an integer and $B \sim \text{Bin}(m, 1/2)$. Then*

$$\mathbb{E}[B] = \frac{m}{2}, \quad \mathbb{E}[B^2] = \frac{m^2}{4} + \frac{m}{4}, \quad \mathbb{E}[B^3] = \frac{m^3}{8} + \frac{3m^2}{8}, \quad \text{and} \quad \mathbb{E}[B^4] = \frac{m^4}{16} + \frac{3m^3}{8} + \frac{3m^2}{16} - \frac{m}{8}.$$

In particular, $\text{Var}(B) = m/4$.

Proof. We use the formula

$$\mathbb{E}[B^p] = \sum_{k=1}^p S_{k,p} m^k \frac{1}{2^k},$$

where $m^{\underline{k}} = m(m-1)\cdots(m-k+1)$ and $S_{k,p}$ are the Stirling numbers of the second kind (see Knoblauch (2008) for instance). \square

Next, we turn to the computation of the third absolute moment of the binomial, which intervenes in the proof of Proposition 6.

Lemma 7 (Third absolute moment of the binomial). *Let $m \geq 1$ be an even integer. Then*

$$\mathbb{E}[|B - m/2|^3] = \frac{m^2}{2^{m+2}} \binom{m}{m/2}.$$

From Lemma 7, we deduce that

$$\forall m \geq 1, \quad \mathbb{E}[|B - m/2|^3] \leq \frac{1}{\sqrt{8\pi}} m^{3/2}, \quad (21)$$

where we used the well-known bound $\binom{m}{m/2} \leq \frac{\sqrt{22^m}}{\sqrt{\pi m}}$. Eq. (21) is better than a Jensen-type bound, which can be obtained by noticing that

$$\left(\mathbb{E}[|B - m/2|^3]\right)^{4/3} \leq \mathbb{E}[(B - \mathbb{E}[B])^4] = \frac{m}{4} \left(1 + \frac{3m-6}{4}\right),$$

where we used Lemma 6 in the last step. This last expression is less than $3m^2/16$, and this approach yields $\mathbb{E}[|B - m/2|^3] \leq (3/16)^{3/4} m^{3/2}$. Since $1/\sqrt{8\pi} \approx 0.2$ whereas $(3/16)^{3/4} \approx 0.28$, we prefer the use of Eq. (21) when bounding the third absolute moment of the binomial.

Proof. We follow Diaconis and Zabell (1991). First, we notice that the polynomial $(X - m/2)^3$ can be written

$$(X - m/2)^3 = \frac{-3}{4} P_3^m(X) + \frac{-3m+2}{8} P_1^m(X), \quad (22)$$

where P_k^m denotes the Kravchuk polynomial of order k (MacWilliams and Sloane, 1977). Using Lemma 1 of Diaconis

and Zabell (1991), we see that

$$\begin{aligned}
 \frac{1}{2^m} \sum_{k=0}^{m/2} \binom{m}{k} (k - m/2)^3 &= \frac{-3}{4} \frac{1}{2^m} \sum_{k=0}^{m/2} \binom{m}{k} P_3^m(k) + \frac{-3m+2}{8} \frac{1}{2^m} \sum_{k=0}^{m/2} \binom{m}{k} P_3^m(k) \\
 &= \frac{-3}{4} \frac{m}{6} \frac{1}{2^m} \binom{m}{m/2} P_2^{m-1}(m/2) \\
 &\quad + \frac{-3m+2}{8} \frac{m}{2} \frac{1}{2^m} \binom{m}{m/2} P_0^{m-1}(m/2) \\
 \frac{1}{2^m} \sum_{k=0}^{m/2} \binom{m}{k} (k - m/2)^3 &= \frac{-m^2}{2^{m+3}} \binom{m}{m/2}.
 \end{aligned}$$

Observing that the third absolute moment is twice the absolute value of the last display yields the desired result. \square

Remark 1. *It is unfortunately not possible to obtain a simple closed-form for a parameter of the binomial p not equal to $1/2$ using this method. Indeed, using the more general expression of the Kravchuk polynomials (sometimes called the Meixner polynomials (Meixner, 1934)) the decomposition obtained in Eq. (22) becomes $(X - mp)^3 = \sum_{q=0}^3 \lambda_q P_q^m(X)$, with*

$$\begin{cases}
 \lambda_0 &= mp(1-p)(1-2p) \\
 \lambda_1 &= \frac{-3m+2}{4}(1-p) + \frac{3m-6}{4}(1-p)(1-2p)^2 \\
 \lambda_2 &= 6(1-2p)(1-p)^2 \\
 \lambda_3 &= -6(1-p)^3.
 \end{cases}$$

In particular, λ_0 is nonzero whenever $p \neq 1/2$. Therefore, the partial sums of the binomial coefficients make their appearance, for which there is no simple closed-form.

B.2 Other probability results

Lemma 8 (Probability splitting). *Let X and Y be two random variables, $t \in \mathbb{R}$ and $\varepsilon > 0$. Then*

$$\mathbb{P}(Y \leq t) \leq \mathbb{P}(X \leq t + \varepsilon) + \mathbb{P}(|X - Y| > \varepsilon).$$

Proof. This result is classical, we report the proof for completeness' sake.

$$\begin{aligned}
 \mathbb{P}(Y \leq t) &= \mathbb{P}(Y \leq t, X \leq t + \varepsilon) + \mathbb{P}(Y \leq t, X > t + \varepsilon) \\
 &\leq \mathbb{P}(X \leq t + \varepsilon) + \mathbb{P}(Y - X \leq t - X, t - X < -\varepsilon) \\
 &\leq \mathbb{P}(X \leq t + \varepsilon) + \mathbb{P}(Y - X < -\varepsilon) \\
 &\leq \mathbb{P}(X \leq t + \varepsilon) + \mathbb{P}(Y - X < -\varepsilon) + \mathbb{P}(Y - X > \varepsilon) \\
 \mathbb{P}(Y \leq t) &\leq \mathbb{P}(X \leq t + \varepsilon) + \mathbb{P}(|X - Y| > \varepsilon).
 \end{aligned}$$

\square

As a direct consequence, we have the following:

Lemma 9 (Convergence in probability implies convergence in distribution). *Let X and Y be two random variables, $t \in \mathbb{R}$ and $s > 0$. Then*

$$\mathbb{P}(X \leq t - s) - \mathbb{P}(|X - Y| > s) \leq \mathbb{P}(Y \leq t) \leq \mathbb{P}(X \leq t + s) + \mathbb{P}(|X - Y| > s).$$

Proof. Applying Lemma 8 to Y and X instead of X and Y , and $t - s$ instead of t yields

$$\mathbb{P}(X \leq t - s) \leq \mathbb{P}(Y \leq t) + \mathbb{P}(|X - Y| > s). \tag{23}$$

Combined with the original statement, we obtain the result. \square

C ADDITIONAL EXPERIMENTAL RESULTS

In this section, we collect additional experimental results omitted from the main paper due to space limitations. Specifically, in Section C.1 we report statistics about the TF-IDF vectorization, in Section C.2 we empirically show that Anchors and exhaustive Anchors produce similar explanations, In Section C.3 we provide a counterexample proving that the default implementation of Anchors does not satisfy Property 4 (Dummy Property). Sections C.4 and C.6 provide empirical validation of Propositions 6 and 9, respectively. Finally, additional experimental results for Sections 4 and 5 are in Sections C.5 and C.7. The code used for the experiments is available at https://github.com/gianluigilopardo/anchors_text_theory.

Setting. All the experiments reported in this Section and in the paper are implemented in Python and executed on CPUs. Three dataset are used: Restaurant Reviews (available at <https://www.kaggle.com/hj5992/restaurantreviews>), Yelp Reviews (available at <https://www.kaggle.com/omkarsabnis/yelp-reviews-dataset>), and IMDB Reviews (available at <https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews>). Unless otherwise specified, all the experiments work with the official implementation of Anchors (available and licensed at <https://github.com/marcotcr/anchor>) and default parameters. The vectorizer is always TF-IDF from https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html with the option `norm=None`. When experiments require it (Sections C.2, C.3, C.5, C.7), we use 75% of the dataset for training and 25% for testing. All machine learning models used in the experiments were trained with the default parameters of <https://scikit-learn.org/>. Finally, we remark that we always consider documents with positive predictions, *i.e.*, such that $f(z) = 1$.

C.1 Typical values of m_j and v_j

Figure 7 and Figure 8 show statistics about the TF-IDF transforms of the two considered datasets. In Figure 7 the average document length b is 11: each document is a short review, generally containing one or two short sentences, while in Figure 8 the average length b is 133: documents are quite longer. This significant difference in documents size is also visible in the multiplicities. In Figure 7, the typical value for the term frequency m_j is 1 and it is rarely higher than 3, while in Figure 8 the average is closer to 2 and multiplicities greater than 10 are present. In contrast, the average, median, and maximum value for the inverse document frequency v_j are around 7 for both datasets: indeed, considering their size is around $N = 1000$ and that the typical value for N_j is 1, we get $v_j = \log \frac{N+1}{N_j+1} + 1 \approx 7$.

C.2 Comparison between Anchors and exhaustive Anchors

We compute the similarity through the Jaccard index, defined as

$$J(A^P, A^d) := \frac{|A^P \cap A^d|}{|A^P \cup A^d|} = \frac{|A^P \cap A^d|}{|A^P| + |A^d| - |A^P \cap A^d|},$$

where A^P is the anchor obtained by running empirical Anchors (exhaustive version with empirical precision as an evaluation function) and A^d with default implementation. Table 3 shows the average Jaccard index for the two datasets considered and five different models. Overall, the output of the two methods is quite similar.

Table 3: Jaccard similarity between exhaustive Anchors and default implementation.

	Indicator	DTree	Logistic	Perceptron	RandomForest
Restaurants	1.00	1.00	0.90	0.87	0.93
Yelp	1.00	1.00	0.71	0.68	0.75

As shown in Figure 8, the Yelp dataset has longer documents, making Anchors more unstable (namely outputting quite different anchors for the same model / document configuration). This explains why the similarity is lower in that case. In addition, Anchors requires a computational capacity that grows exponentially with the length of the document (and the

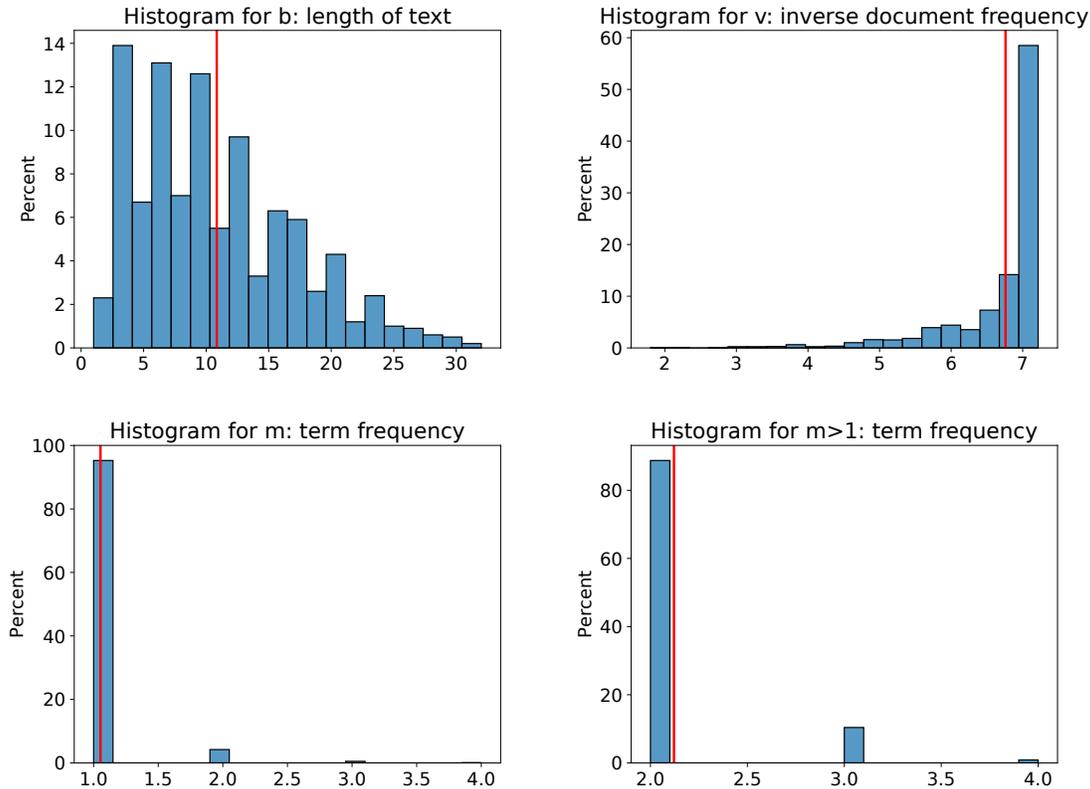


Figure 7: Histograms for length of the document b (upper left), the inverse document frequency v_j (upper right), the term frequency m (lower left), the term frequency when $m > 1$ (lower right) for the Restaurant Reviews dataset from <https://www.kaggle.com/hj5992/restaurantreviews>. Average value is reported in red.

length of the optimal anchor). This makes it particularly onerous to apply empirical Anchors to large documents. Indeed, the experiment of Table 3 requires about half an hour on Restaurants reviews, while more than 24 hours are needed on Yelp reviews.

C.3 Dummy property

We report a counterexample showing that the default implementation of Anchors does not satisfy Proposition 4. In Figure 9, the word *indie* appears in 4 anchors, even though the model does not depend on it. While the frequency of occurrence is not high, it is still non-zero. This is slightly problematic in our opinion: since the model does not depend on the word *indie*, its appearance in the explanation is misleading for the user. We conjecture that this behavior is entirely due to the optimization procedure used in the default implementation of Anchors, since the exhaustive version is guaranteed not to have this behavior by Proposition 4. We want to emphasize that there is nothing special with the example presented here and other counterexamples can be readily created.

C.4 Empirical validation of Proposition 6: Precision of a linear classifier

Figure 10 shows an empirical validation for Proposition 6 for different document size and for anchors of different sizes. The fit between the empirical distribution and $\bar{\Phi} \circ L$ is much better as predicted by Proposition 6, even for small values of d . This motivates our further study of the approximate precision instead of the precision. From the results in Figure 10, we can see why the anchors need to be small with respect to the document size: if they are too large, the approximation of the precision is not justified. We remark, again, that this assumption is entirely reasonable, since an anchor using more than half the document to explain a prediction is not interpretable. In addition, Anchors rarely returns such anchors.

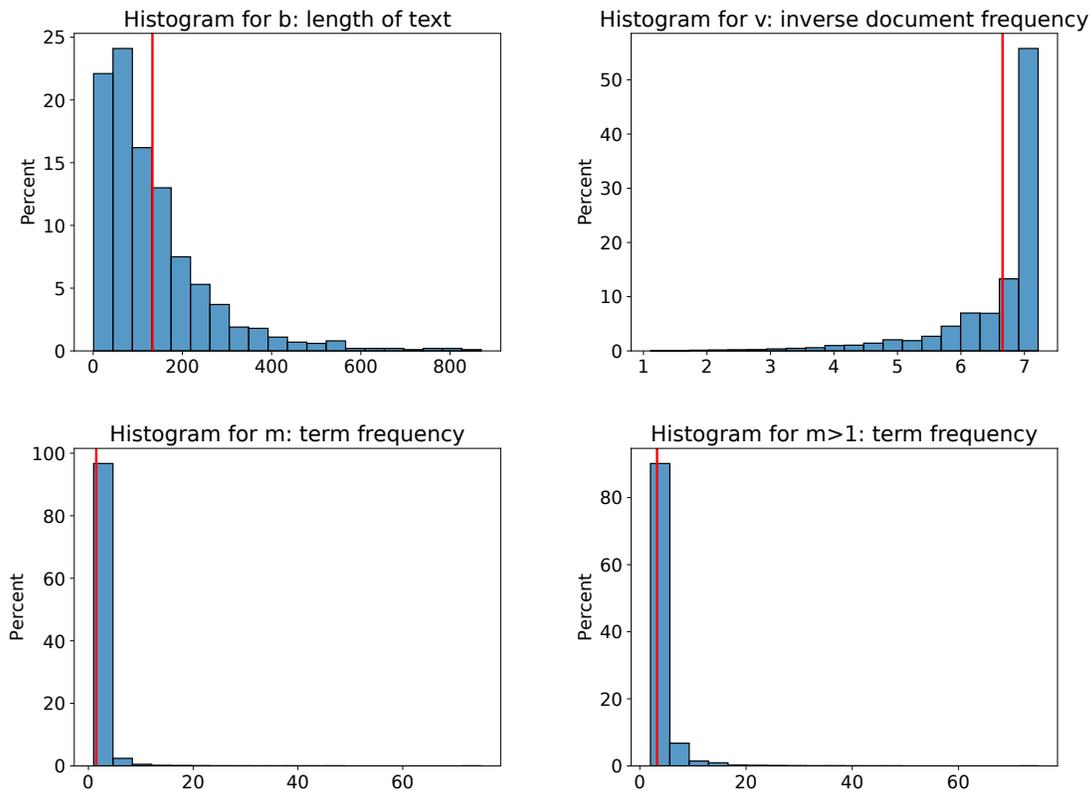


Figure 8: Histograms for length of the document b (upper left), the inverse document frequency v_j (upper right), the term frequency m (lower left), the term frequency when $m > 1$ (lower right) for a subset of the Yelp Reviews dataset from <https://www.kaggle.com/omkarsabnis/yelp-reviews-dataset>. Note that the maximum value of multiplicity m is 75 in this case, while the average (in red) is 1.5.

C.5 Additional experiments for Section 4: Analysis on explainable classifiers

In this Section we report additional experiments for our Analysis on explainable classifiers. First, we validate our results on simple if-then-rules: Figure 12 and Figure 13 illustrate Proposition 8 and Proposition 5, respectively.

Second, we validate Proposition 7 as in Figure 10, *i.e.*, after training a logistic model (Figure 14) and a perceptron model (Figure 15), we apply a shift S to the intercept λ_0 , as follows

$$f(z) = \mathbf{1}_{\lambda^\top \varphi(z) + (\lambda_0 - S) > 0}. \quad (24)$$

As S increases, the prediction becomes harder, and longer anchors are needed to reach the precision threshold. When a new word is included, we show that, as predicted by Proposition 7, the first word with higher $\lambda_j v_j$ is picked.

Error bars In our experiments there are two sources of variability, coming from different runs and documents, as we ran 10 times Anchors on each positively classified document. Figure 11 shows the standard deviation for 10 runs on Restaurant reviews (model is a 10-layers neural network): for half the documents, it is actually zero.

To further demonstrate this phenomenon, we also conducted the following experiment. We first trained a logistic model on three review datasets, achieving accuracies between 80% and 90%. 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 $\lambda_j v_j$. In Table 1 we report the average Jaccard index: results validate Proposition 7.

C.6 Empirical validation of Proposition 9: Normalized-TF-IDF, Berry-Esseen

Figure 16 shows an empirical validation for Proposition 9 for different document size and for anchors of different sizes.

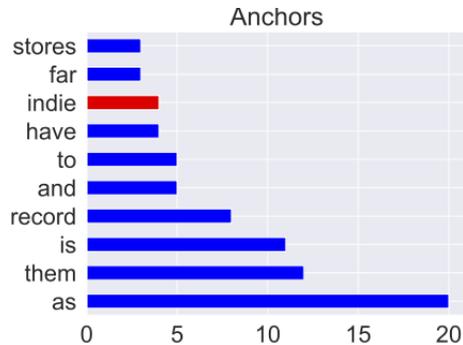


Figure 9: 10 most frequent anchors on 100 runs of default Anchors on a logistic model with zero coefficient for *indie* and arbitrary coefficients for the other words. *indie* is a dummy feature, but still appears in 4 anchors.

C.7 Additional experiments for Section 5: Anchors on Neural Networks

We show in Table 2 additional experiments that validate our conjecture expressed in Section 5. To this end, we trained, for each dataset (Restaurants, Yelp, and IMDB), three feed-forward neural networks, with 3, 10, and 20 layers, achieving accuracies around 90%. The code used for model training is available at https://github.com/gianluigilopardo/anchors_text_theory. We then ran Anchors with default settings 10 times on positively classified documents. For each document, we get the gradient of the model with respect to the input: for all $j \in [d]$, $\lambda_j := \frac{\partial g(\varphi(x))}{\partial \varphi(x)_j}$. We then measure the average Jaccard similarity between the anchor A and the first $|A|$ word ranked by $\lambda_j v_j$.

C.8 BERT replacement

As discussed in Section 4.1, we study the UNK-replacement option even if when replacing words with a fixed token can produce unrealistic samples and lead to out-of-distribution issue. Nevertheless, we performed the same experiments of Section 5 using the BERT-replacement option when a 3-layers neural network is applied on a sample of 50 Restaurants reviews. Somewhat surprisingly, our message still stands: we reach a Jaccard Similarity of 0.83 ± 0.2 , similarly to the UNK setting. What is more, we notice that such option is 10 times slower and produces longer anchors.

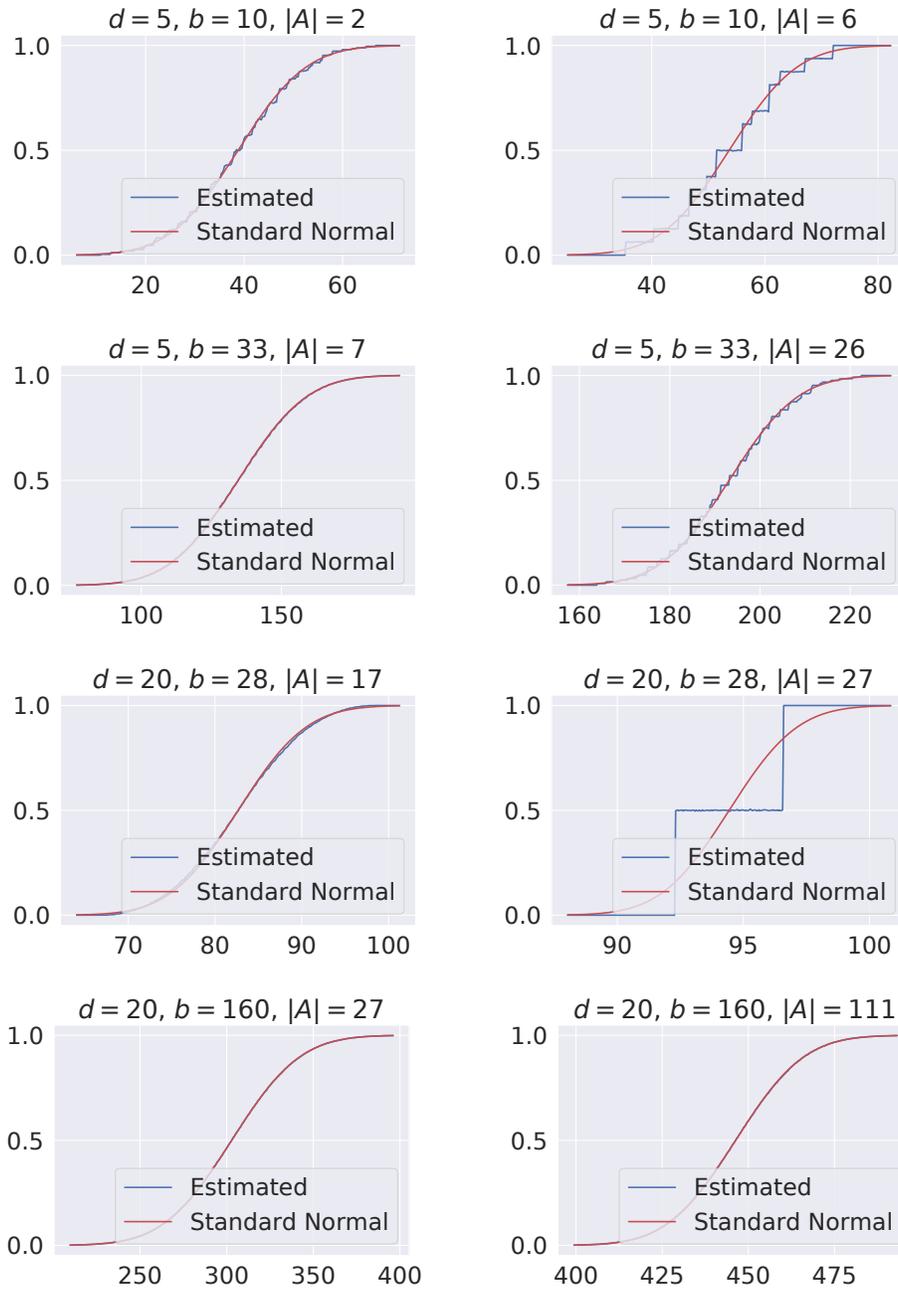


Figure 10: Illustration of Proposition 6. The multiplicities are arbitrary numbers between 1 and 10. The $\lambda_j v_j$ where drawn according to a Gaussian. Monte-Carlo simulation of the probability in blue 10^5 simulations). In red, the cumulative distribution function of the $\mathcal{N}(0, 1)$. Note that Proposition 6 assumes $|A| \leq b/2$: the approximation may be inaccurate when this assumption is not satisfied (right column).

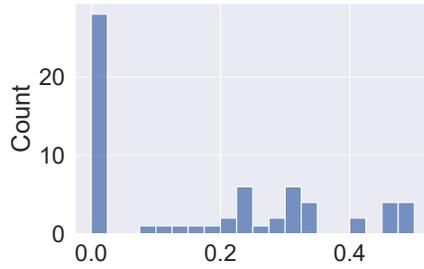


Figure 11: Jaccard index standard deviations for 10 runs on Restaurant reviews (model is a 10-layers neural network), same experiment as Table 2 in the paper. Overall standard deviation is 0.39, while the average one is 0.17.

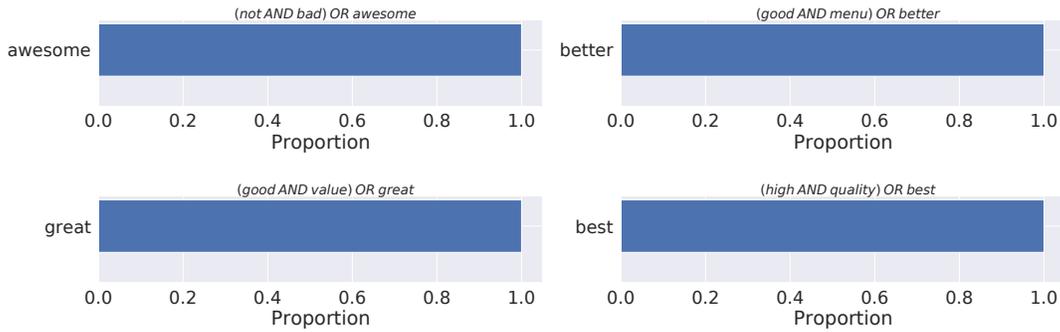


Figure 12: Illustration of Proposition 8. Count on 100 runs of Anchors on four different reviews. Classification rules consist of two conditions, shown above each figure. Note that, for each case, **both conditions are satisfied** by the example. The shorter is always selected, as predicted by Proposition 8.

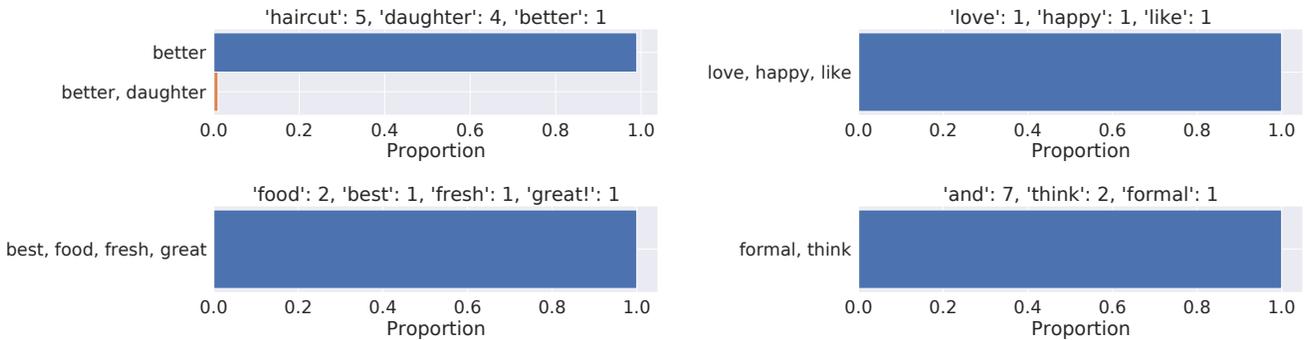


Figure 13: Illustration of Proposition 5. Count on 100 runs of Anchors on four different reviews. The classifier looks for the joint presence of the words reported above each figure (for instance, the model in the upper left panel predicts 1 if z contains “haircut,” “daughter,” and “better”). The multiplicities of each word in the document is reported. As predicted by Proposition 5, words with multiplicity higher than a given threshold disappear from the explanation.

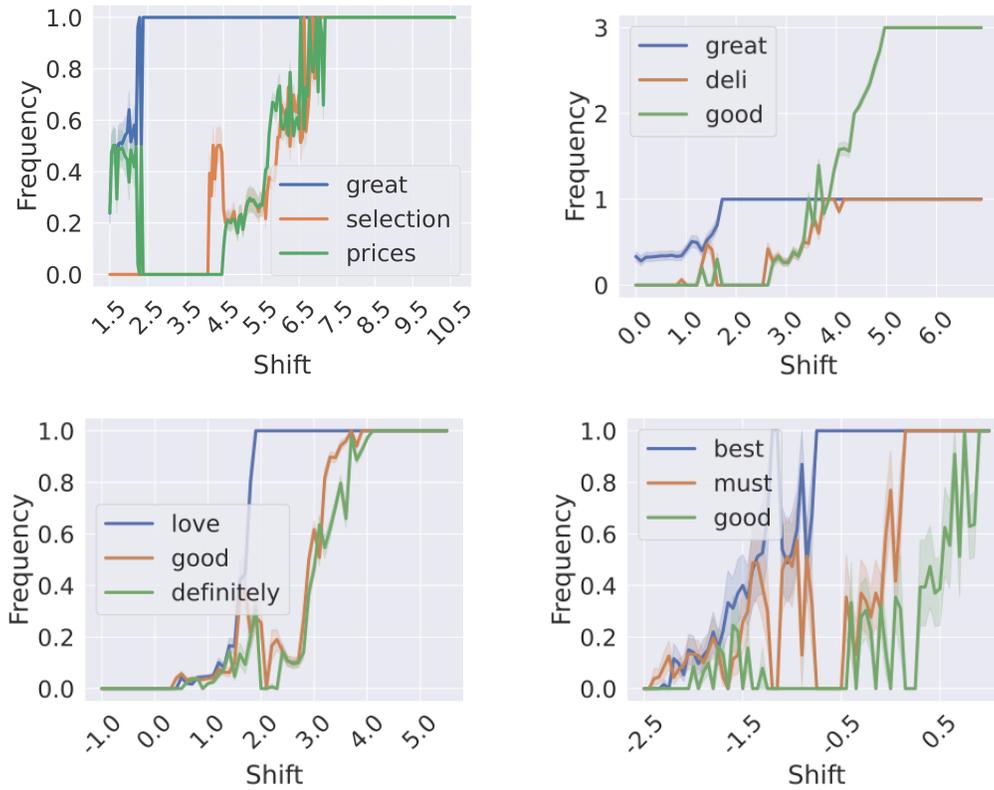


Figure 14: Illustration of Proposition 7. Frequency on 100 runs of Anchors on four different documents with four different logistic models when a shift S is applied. Legend shows the first three words ordered by $\lambda_j v_j$. For example, in the lower-right figure, the shift S increases from -2.5 to 1 . As predicted by Proposition 7, first the words with higher $\lambda_j v_j$ are selected. Note that the overlap of some curves is due to similar coefficients for the corresponding words.

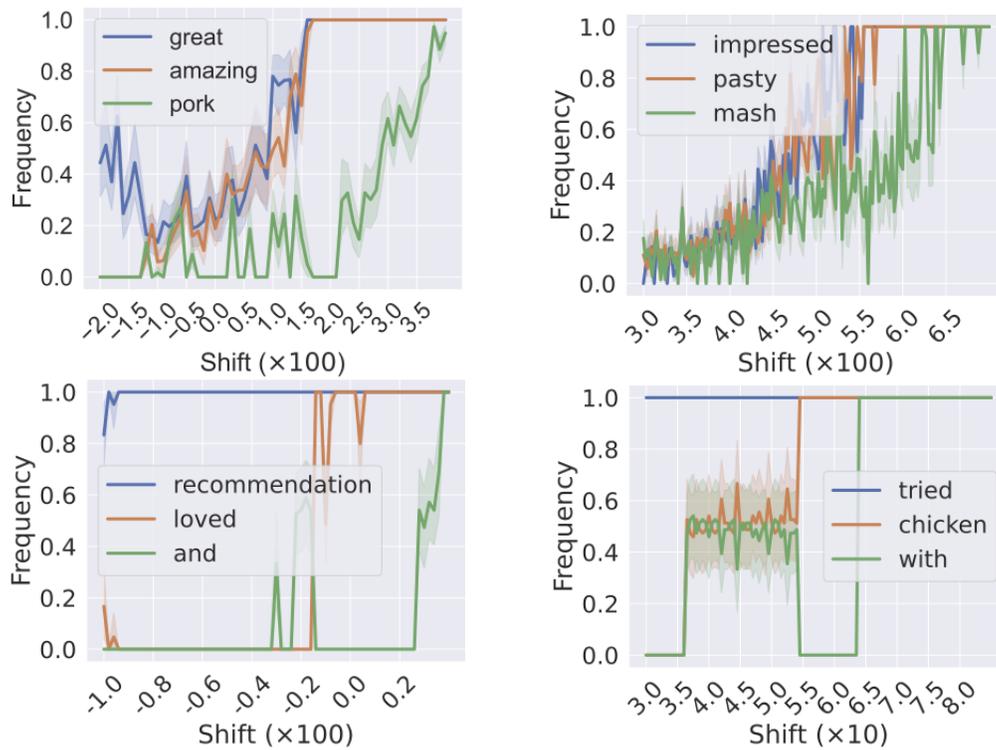


Figure 15: Illustration of Proposition 7. Frequency on 100 runs of Anchors on four different documents with four different perceptron models when a shift S is applied. Legend shows the first three words ordered by $\lambda_j v_j$. Anchors includes new words in order of $\lambda_j v_j$, as predicted by Proposition 7. The overlap of some curves is due to similar coefficients for the corresponding words. For example, in the top-left figure, “great” has a coefficient $\lambda_j v_j$ equal to 207, close to the coefficient for “amazing”, 192.

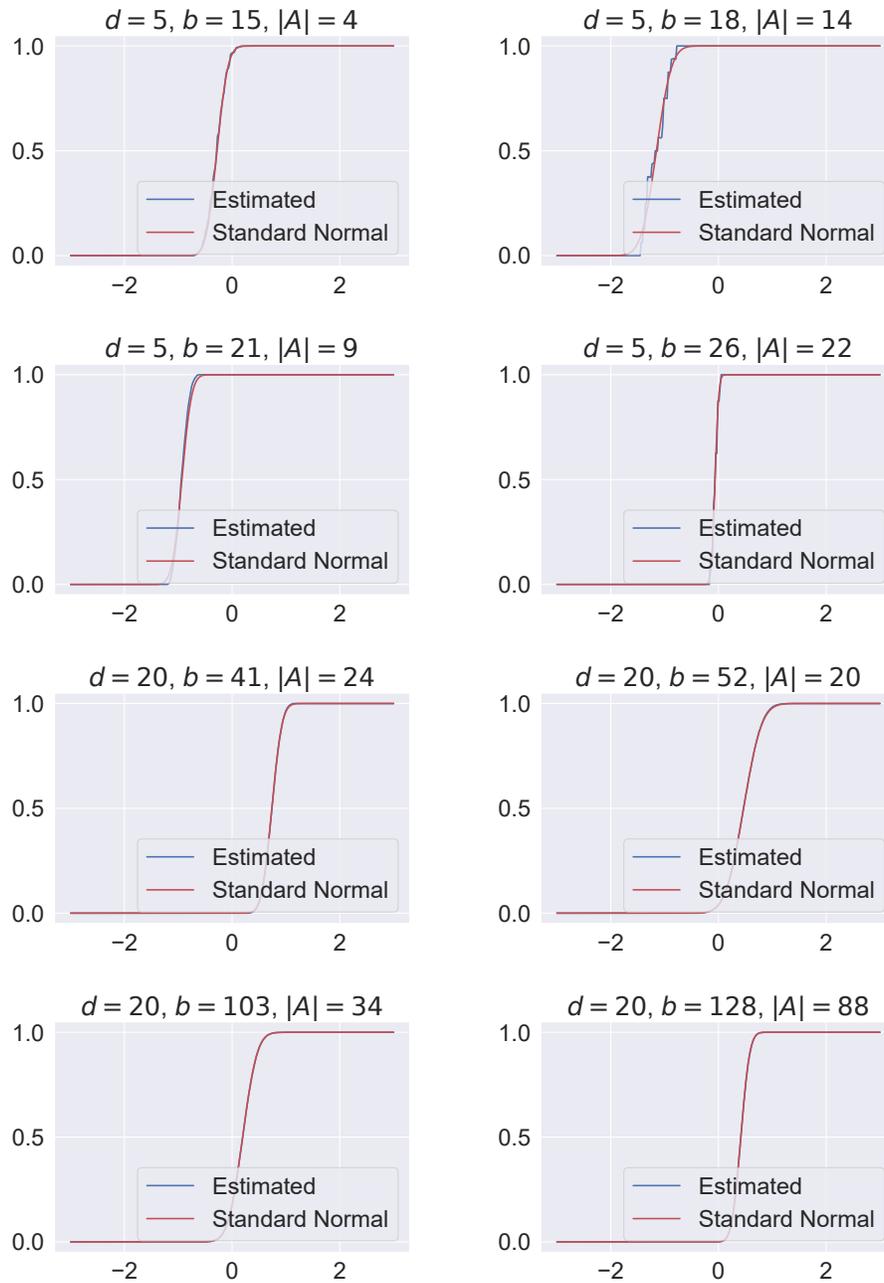


Figure 16: Illustration of Proposition 9. The multiplicities are arbitrary numbers between 1 and 10. The $\lambda_j v_j$ were drawn according to a Gaussian. Monte-Carlo simulation of the probability in blue (10^5 simulations). In red, the cumulative distribution function of the $\mathcal{N}(0, 1)$. Note that Proposition 9 assumes $|A| \leq b/2$: the approximation may be inaccurate when this assumption is not satisfied (right column).