Machine Learning (2023) 112:4289–4322
https://doi.org/10.1007/s10994-022-06150-7

Explaining short text classification with diverse synthetic exemplars and counter-exemplars

Orestis Lampridis[1] · **Laura State**[2,3] · **Riccardo Guidotti**[2,4] · **Salvatore Ruggieri**[2,4]

Received: 1 March 2021 / Revised: 22 September 2021 / Accepted: 21 February 2022 /
Published online: 11 May 2022

Abstract
We present XSPELLS, a model-agnostic local approach for explaining the decisions of black box models in classification of short texts. The explanations provided consist of a set of exemplar sentences and a set of counter-exemplar sentences. The former are examples classified by the black box with the same label as the text to explain. The latter are examples classified with a different label (a form of counter-factuals). Both are close in meaning to the text to explain, and both are meaningful sentences – albeit they are synthetically generated. XSPELLS generates neighbors of the text to explain in a latent space using Variational Autoencoders for encoding text and decoding latent instances. A decision tree is learned from randomly generated neighbors, and used to drive the selection of the exemplars and counter-exemplars. Moreover, diversity of counter-exemplars is modeled as an optimization problem, solved by a greedy algorithm with theoretical guarantee. We report experiments on three datasets showing that XSPELLS outperforms the well-known LIME method in terms of quality of explanations, fidelity, diversity, and usefulness, and that is comparable to it in terms of stability.

Keywords Explainable AI · Short text Classification · Synthetic exemplars · Counter-factuals · Model-agnostic explanation

Editors: Annalisa Appice, Grigorios Tsoumakas.

✉ Riccardo Guidotti
riccardo.guidotti@unipi.it

Orestis Lampridis
lorestis@csd.auth.gr

Laura State
laura.state@di.unipi.it

Salvatore Ruggieri
salvatore.ruggieri@unipi.it

[1] Aristotle University of Thessaloniki, 541 24 Thessaloniki, Greece

[2] University of Pisa, Largo B. Pontecorvo, 3, 56127 Pisa, Italy

[3] Scuola Normale Superiore, Piazza dei Cavalieri 7, 56126 Pisa, Italy

[4] ISTI-CNR, Via G. Moruzzi, 1, 56127 Pisa, Italy

1 Introduction

Text classification is the task of assigning to strings of text a label from a predefined set. Automating text classification is motivated by the massive amount of texts available in the current digital age, and by the complexity of determining the correct label for new texts. Automated text classification is made possible by increasingly accurate algorithms (Kowsari et al. 2019), such as supervised models from machine learning (Sebastiani 2002) and deep learning (Minaee et al. 2021), or semantic methods Altinel and Ganiz (2018) that overcome the simple representation of texts as bag of words. For example, sentiment classification Liu and Zhang (2012) of subjective texts about a product (or, a brand, a politician, a regulation, etc.) into positive, negative, or neutral opinions is supported by several techniques Hemmatian and Sohrabi (2019). Other applications of text classification Li et al. (2020); Zhou et al. (2020) include: filling texts into appropriate sections in websites or folders (document organization); assigning one or more subjects to texts (topic labeling); selecting relevant texts from a stream (text filtering); detecting unsolicited emails (spam detection); determining the stance – favor, against, neither – of the author of a text towards a target (stance detection); etc.

The classification of short texts (Song et al. 2014), which abound in micro-blogging sites such as Twitter and in online reviews, is especially challenging, due to their sparsity, non-uniformity, and noisiness. Deep Neural Networks (DNNs) Korde and Mahender (2012), Zhang et al. (2015) and Random Forests (RFs) da Silva et al. (2014) and Xu et al. (2012), have been shown to be effective in terms of predictive accuracy and robustness to noise. However, the decision logic learned by a DNN or by a RF to classify a given text remains obscure to human inspection. These inscrutable "black box" models may reproduce and amplify biases learned from data, such as prejudice (Bolukbasi et al. 2016), or they may introduce new forms of bias (Olteanu et al. 2019), e.g., due to spurious correlations. When opinions concern specific individuals, this may also result in social discrimination against protected-by-law groups identified by their sensitive traits (e.g. gender identity, sexual orientation, ethnic background) (Ntoutsi 2020).

Explainability of decisions made by black box models is nowadays a mandatory requirement (Doshi-Velez and Kim 2017; Freitas 2013) for the social acceptance of Artificial Intelligence (AI) applications Danks (2019). Developers need to understand a model's decisions for debugging and optimization. For example, to characterize conditions under which the model can be trustfully applied, and conditions for which the model should abstain to make decisions. People subject to black box decisions may inquire to be provided with "meaningful information of the logic involved" (articles 13–15 of European Union General Data Protection Regulation, sometimes referred to as *right to legibility* (Malgieri and Comandé 2017) or *right to explanation* (Selbst and Powles 2017)). For example, if a comment in a social network has been removed because it has been classified as *hate speech*, the author has the right to know *why* the machine learning system has assigned such a label to her comment.

In this paper, we investigate the problem of explaining the decisions of a black box model (simply, a black box) when classifying a given short text in input, e.g., as in sentiment classification. We design and experiment with a model-agnostic local approach named XSPELLS (explaining sentiment prediction generating exemplars in the latent space). XSPELLS's explanations for the prediction $y = b(x)$ assigned by a black box b to a short text x consists of set of *exemplar* texts E, a set of *counter-exemplar* texts C, and the most frequent words in each of those sets $W = W_E \cup W_C$. Exemplars are sentences classified by

the black box with the same label as x and close in meaning to x. They are intended to provide the user with hints about the kind of texts in the neighborhood of x that the black box classifies in the same way as x. Counter-exemplars are sentences that the black box classifies differently from y, but like exemplars, are also close in meaning to x. They are intended to provide the user with hints about the kind of texts in the neighborhood of x that the black box classifies differently from x. The usefulness of *counter-factual reasoning* has been widely recognized in the literature on Explainable AI Byrne (2019), particularly as a tool for causal understanding of the behavior of black boxes. Diversity of the provided counter-exemplars is modeled as an optimization problem, which is solved by resorting to a greedy algorithm with a theoretical guarantee. By contrasting exemplars and counter-exemplars, the user can gain an understanding of the factors affecting the classification of x. To help such an understanding, XSPELLS provides also the most frequent words appearing in the exemplar texts E and in the counter-exemplar texts C.

The main novelty of our approach lies in the fact that the exemplars and counter-exemplars produced by XSPELLS are *meaningful* texts, albeit synthetically generated. We map the input text x from a (sparse) high-dimensional vector space into a low-dimensional latent space vector z by means of Variational Autoencoders Kingma and Welling (2014), which are effective in encoding and decoding diverse and well-formed short texts (Bowman et al. 2016). Then we study the behavior of the black box b in the neighborhood of z, or, more precisely, the behavior of b on texts decoded back from the latent space. Finally, we exploit a decision tree built from latent space neighborhood instances to drive the selection of exemplars and counter-exemplars. Experiments on three standard datasets and two black box classifiers show that XSPELLS overtakes the baseline method LIME Ribeiro et al. (2016) by providing understandable, faithful, useful, and stable explanations.

This paper extends the conference version Lampridis et al. (2020) in several aspects. First, we formulate the problem of diverse counter-exemplar selection and provide a solution based on a greedy algorithm. Second, in addition to training a VAE on a subset of available data, we consider using a pre-trained VAE from the state of the art. Third, a deeper experimental qualitative/quantitative analysis is conducted. The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 formalizes the problem and recalls key notions for the proposed method, which is described in Sect. 4. Section 5 presents the experimental results. Finally, Sect. 6 summarizes our contribution, its limitations, and future work.

2 Related work

Research on interpretability and explainability in AI has bloomed over the last few years (Guidotti et al. 2019b; Miller 2019), with many implementations of proposed methods (Bodria et al. 2021; Linardatos et al. 2021). Intrinsically explainable AI models can be directly interpretable by humans, or the explanation of their decisions arise as part of their prediction process (self-explainability). Examples in the area of short text classification include linear classifiers exploiting word taxonomies (Skrlj et al. 2021) or lexicons (Clos and Wiratunga 2017). The best performing text classifiers, however, rely on black box models, which are inaccessible, inscrutable, or simply too complex for humans to understand. Hence, they require post-hoc explanations of their decisions. Explanation methods can be categorized as: (i) *Model-specific* or *model-agnostic*, depending on whether the approach requires access to the internals of the black box; (ii) *Local* or *global*, depending

on whether the approach explains the prediction for a specific instance or the overall logic of the black box.

XSPELLS falls into the category of local, model-agnostic methods. Well known tools in this category that are able to work also on textual data include LIME, ANCHOR and shap. LIME Ribeiro et al. (2016) randomly generates synthetic instances in the neighborhood of the instance to explain. An interpretable linear model is trained from such instances. Feature weights of the linear model are used for explaining the feature importance over the instance to explain. In the case of texts, a feature is associated to each of the top frequent words in a dataset. LIME has two main weaknesses. First, the number of top features/words to be considered is assumed to be provided as an input by the user. Second, the neighborhood texts are generated by randomly removing words, possibly generating meaningless texts. ANCHOR Ribeiro et al. (2018) follows the main ideas of LIME but it returns decision rules (called anchors) as explanations. In the case of texts, such rules state which words, once fixed, do not alter the decision of the black box when randomly replacing all other words by similar words (in an embedding space) with the same part-of-speech (POS) tag. ANCHOR adopts a bandit algorithm that constructs anchors with predefined minimum precision. Its weaknesses include the need for user-defined precision threshold parameters, and, as for LIME, the generation of possibly meaningless instances. shap Lundberg and Lee (2017) relates game theory with local explanations and overcomes some of the limitations of LIME and ANCHOR. Also shap audits the black box with possibly meaningless synthetic sentences. The method XSPELLS proposed in this paper recovers from this drawback by generating the sentences for the neighborhood in a latent space by taking advantage of variational autoencoders.

With regard to model-specific local approaches, lionets, deeplift and neurox are designed to explain deep neural networks and they are able to work on textual data. deeplift Shrikumar et al. (2017) decomposes the prediction of neural networks on a specific input by back-propagating the contributions of all neurons in the network to the input features. Then it compares the activation of each neuron to its "reference activation" and it assigns contribution scores according to the difference. neurox Dalvi et al. (2019) facilitates the analysis of individual neurons in DNNs. In particular, it identifies specific dimensions in the vector representations learned by a neural network model that are responsible for specific properties. Afterwards, it allows for the ranking of neurons and dimensions based on their overall saliency. Finally, lionets Mollas et al. (2019) looks at the penultimate layer of a DNN, which models texts in an alternative representation, randomly permutes the weights of nodes in that layer to generate new vectors, classifies them, observes the classification outcome and returns the explanation using a linear regressor like LIME. Differently from these model-specific methods, XSPELLS is not tied to a specific architecture and can be used to explain any black box classifier.

Other approaches such as IntGrad Sundararajan et al. (2017), LRP Bach et al. (2015), DeepLift Shrikumar et al. (2017) and L2X Chen et al. (2018) are designed to explain image classifiers. However, they can be adapted to work on text classifiers by returning a sort of saliency map as explanation, also named sentence highlighting, that highlights the most important words responsible for the classification. For instance, Arras et al. (2017) adapts the model-specifc approach of layer-wise relevance propagation (LRP Bach et al. 2015) to extract word-wise relevance score. Similarly, Li et al. (2016) proposes an attention-based sentence-highlighting explanation system that extracts a saliency score for every word by using the weights layer of a DNN black box. Attention-based explanations aim also at connecting the importance of various words in a sentence. For instance, Vaswani et al. (2017) provides the explanation as a matrix where rows and columns represent a word and the

value in the cell is proportional to the self-attention in the explanation between the words at the row/column of the cell. Visualization of such a matrix can also highlight connections among words (Hoover et al. 2019). Compared to such approaches, the explanation given by XSPELLS is "instance-wide" as it is formed by prototypes showing with examples/counter-examples similar/different classification outcomes. On the other hand, those approaches return an explanation that points the attention on single words causing the classification outcomes.

Regarding counter-factual approaches, while there is a growing literature for tabular data and images (Artelt and Hammer 2019; Verma et al. 2020), to the best of our knowledge our proposal is an original contribution in the context of short text classification. A form of contrastive explanations has been proposed in the local model-specific approach of Croce et al. (2019) for a self-explaining question classification system based on LRP. Here, the top texts from the training set which contribute the most (negatively) to the decision are returned as counter-exemplars.

Finally, for explainability in Natural Language Processing beyond classification, we refer to a recent survey Danilevsky et al. (2020) and its associated living website Qian etal (2021).

3 Setting the stage

In this paper, we address the black box outcome explanation problem Guidotti et al. (2019b) in the domain of short text classification. We will only consider and experiment with short texts such as posts on social networks, brief reviews, or single sentences. These are typically categorized into two or a small number of class labels, as in sentiment classification, stance detection, hate-speech recognition, etc. Text classification is particularly challenging in these cases, with high risks of biased decisions accompanied with an urgent need for black box explanation methods. A black box model is a non-interpretable or inaccessible text classifier b which assigns a class label y to a given text x, i.e., $b(x) = y$. We assume that the black box b can be queried at will. We use the notation $b(X)$ as a shorthand for $\{b(x) \mid x \in X\}$.

Definition 1 Let b be a black box text classifier, and x a text for which the decision $y = b(x)$ has to be explained. The *black box outcome explanation problem for text classification* consists of providing an explanation $\xi \in \Xi$ belonging to a human-interpretable domain Ξ.

In the following, we introduce the key tools used in our approach.

3.1 Local explainers, factuals and counter-factuals

In the context of tabular data, a widely adopted human-interpretable domain Ξ consists of *if-then* rules. They provide conditions (in the if-part) met by the instance x to be explained, that determined the answer of the black box (then-part). Rules can also be used to provide *counter-factuals*, namely alternative conditions, not met by x, that would determine a different answer by the black box Byrne (2019). In our approach, we will build on LORE Guidotti et al. (2019a), a local explainer for tabular data that learns a decision tree from a given neighborhood Z of the instance to explain. Such a tree is a *surrogate* model of the

black box, i.e., it is trained to reproduce the decisions of the black box locally to x. LORE provides as output: (i) a *factual* rule $p \rightarrow y$, corresponding to the path p in the surrogate tree that explains why an instance x has been labeled as y by the black box b; and (ii) a set of *counter-factual* rules $p[\delta] \rightarrow y'$ explaining (minimal) changes δ in the features of x that would change the class y assigned by b to y'. In LORE, the neighborhood Z is synthetically generated using a genetic algorithm that balances the number of instances similar to x and with its same label y, and the number of instances similar to x but with a different label $y' \neq y$ assigned by b. One first problem with directly using LORE is that texts are not tabular data. We will tackle this issue by mapping a large m-dimensional space of words to a small k-dimensional space of numbers (*latent space*), a tabular space from which rules can be extracted.

3.2 Variational autoencoders for short text generation

Local explanation methods, such as LORE, audit the behavior of a black box in the neighborhood of the instance to explain. A second non-trivial problem with textual data is how to generate meaningful synthetic sentences in the neighborhood (w.r.t. semantic similarity) of the instance. We tackle this problem by adopting Variational Autoencoders (VAEs) (Kingma and Welling 2014), which have been shown to be very effective (Bowman et al. 2016) in generating diverse and well-formed (short text) sentences. A VAE is trained with the aim of learning a representation that reduces the dimensionality from the space of words to the latent space, also capturing non-linear relationships. An *encoder* ζ, and a *decoder* η are simultaneously learned with the objective of minimizing the *reconstruction loss*. Starting from the reduced encoding $z = \zeta(x)$, the VAE reconstructs a representation as close as possible to its original input $\tilde{x} = \eta(z) \simeq x$. After training, the decoder can be used with generative purposes to reconstruct instances never observed by generating vectors in the latent space of dimensionality k. The difference to standard autoencoders (Hinton and Salakhutdinov 2006) is that VAEs are trained by considering an additional limitation on the loss function such that the latent space is scattered and does not contain "dead zones". Indeed, the name "variational" comes from the fact that VAEs work by approaching the posterior distribution with a variational distribution. The encoder ζ emits the parameters for this variational distribution, in terms of a multi-factorial Gaussian distribution, and the latent representation is taken by sampling this distribution. The decoder η takes as input the latent representation and focuses on reconstructing the original input from it. The avoidance of dead zones ensures that the instances reconstructed from vectors in the latent space, e.g., posts or tweets, are semantically meaningful Bowman et al. (2016).

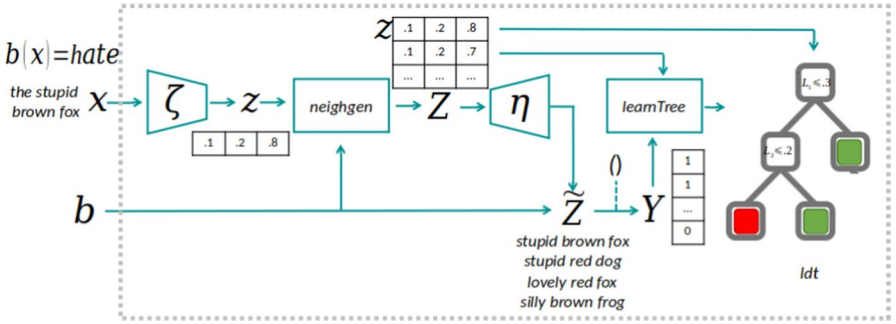

Fig. 1 XSPELLS process (part 1) on a sample input. XSPELLS takes as input the short text x and the assigned label $b(x)$. After coding the text into the latent space and generating a neighborhood, a decision tree over the latent features $(L_1, L_2, \ldots)$ is learned

Algorithm 1: XSPELLS(x, b, ζ, η)

Input : x - text to explain, b - black box, ζ - encoder, η - decoder
Output: ξ - explanation

$z \leftarrow \zeta(x)$; // encode text into the latent space
$Z \leftarrow neighgen(z, b, \zeta, \eta)$; // generate latent neighborhood
$\tilde{Z} \leftarrow \eta(Z)$; // decode neighborhood
$Y \leftarrow b(\tilde{Z})$; // classify neighborhood
$ldt \leftarrow learnTree(Z, Y)$; // learn latent surrogate decision tree
$r \leftarrow rule(z, ldt)$; // extract factual latent rule
$E, C \leftarrow explCexpl(r, Z, \tilde{Z}, Y)$; // select exemplars and counter-exemplars
$W \leftarrow mostCommon(E, C)$; // extract most common words
return $\xi = \langle E, C, W \rangle$; // return explanation

4 Explaining short text classifiers

We propose a local model-agnostic explainer for classification of short texts, called XSPELLS (*e* xplaining sentiment prediction generating exemplars in the latent space). Given a black box b, a short text x, e.g., a post on a social network, and the class label $y = b(x)$ assigned by the black box, e.g., *hate* or *neutral*, the explanation provided by XSPELLS is composed of: (i) A set of *exemplar* texts; (ii) A set of *counter-exemplar* texts; and, (iii) The set of *most common words* in exemplars and counter-exemplars. Exemplar and counter-exemplar texts respectively illustrate instances classified with the same and with a different label than x. Such texts are close in meaning to x, and they offer an understanding of what makes the black box determine the label of texts in the neighborhood of x. Exemplars help in understanding reasons for, e.g., the sentiment assigned to x. Counter-exemplars help in understanding reasons that would reverse the sentiment assigned. The most common words in the exemplars and counter-exemplars may allow for highlighting terms (not necessarily appearing in x) that discriminate between the assigned label and a different one. These components form the human-interpretable explanation $\xi \in \Xi$ for the classification $y = b(x)$ returned by XSPELLS, whose aim is to satisfy the requirements of counter-factuability, usability, and meaningfulness of an explanation (Byrne 2019; Miller 2019; Pedreschi et al. 2019).

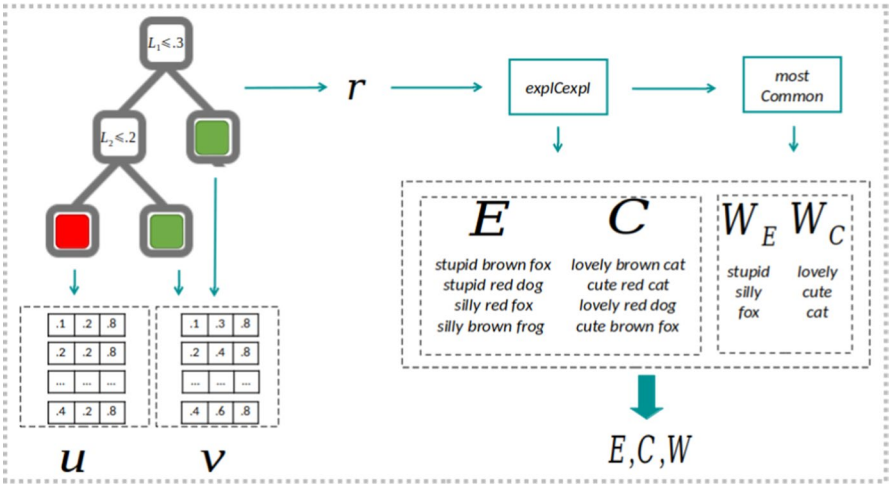

Fig. 2 XSPELLS process (part 2) on a sample input. The figure starts at the extraction of exemplars (u) and counter-exemplars (v) from the decision tree in the latent space. The output of XSPELLS is a set of exemplars and counter-exemplars, and the most common discriminative words

Besides the black box b and the text x to explain, XSPELLS is parametric in: an encoder ζ and a decoder η for representing texts in a compact way in the latent space. Algorithm 1 details XSPELLS, and Figs. 1 and 2 show the steps of the explanation process on a sample input. We describe the process in detail by a simple example x: "the stupid brown fox", classified as "hate". First, x is transformed into a low-dimensionality vector $z = \zeta(x)$ in the latent space. XSPELLS then generates a neighborhood Z of z, which is decoded back to a set of texts $\tilde{Z}$ (e.g. "stupid brown fox", "stupid red dog"). The dataset Z and the decisions of the black box on the decoded text $Y = b(\tilde{Z})$ are used to train a surrogate decision tree (in the latent space).

Then, the *explCexpl*() module selects exemplars E (e.g. "stupid brown fox", "stupid red dog") and counter-exemplars C (e.g. "lovely brown cat", "cute red cat") from Z by exploiting the knowledge extracted (i.e., the decision tree branches), and decodes them into texts. Finally, the most common words $W = W_E \cup W_C$ (here "stupid, silly, fox", "lovely, cute, cat") are extracted from E and C and the overall explanation ξ is returned. Details of each step are presented in the rest of this section.

4.1 Latent encoding and neighborhood generation

The input text x is first passed to a trained VAE ζ (line 1 of Algorithm 1), thus obtaining the latent space representation $z = \zeta(x)$. The number of latent dimensions k is kept low to avoid dimensionality problems. While XSPELLS is parametric in the encoder-decoder, we considered in the experiments two actual implementations. The first one is trained on a subset of available data. It captures the sequential information in texts by means of long short-term memory layers (LSTM) (Hochreiter and Schmidhuber 1997) for both the encoder ζ and decoder η (lines 1 and 3). In particular, the decoder η is trained to predict the next characters of a text given the previous characters of the text, with the purpose of

reconstructing the original text. The second implementation relies instead on a state-of-the-art VAE already pretrained on a large text corpus (details in Sect. 5).

XSPELLS generates a set Z of n instances in the latent feature space for a given z. The neighborhood generation function *neighgen* (line 2) can be implemented by adopting several different strategies, ranging from a purely random approach like in LIME (Ribeiro et al. 2016), to using a given distribution and a genetic algorithm maximizing a fitness function like in LORE (Guidotti et al. 2019a). XSPELLS adopts a random generation of latent synthetic instances by relying on the fact that the encoder maps uniformly the data distribution over the latent space. Duplicated instances in the random generation processes are removed, keeping only distinct ones. Next, XSPELLS uses the synthetically generated instances $\tilde{Z}$ for querying the black box b (line 4). This is made possible by turning back the latent representation to text through the decoder η Bowman et al. (2016) (line 3). We tackle the requirement of generating *local* instances by randomly generating $N \gg n$ latent instances, and then retaining in Z only the n closest instances to z, i.e., $|Z| = n$. Such an heuristics balances diversity of instances in Z with similarity w.r.t. x. The distance function used in the latent space is the Euclidean distance. The neighborhood generation *neighgen* actually returns a partitioned set $Z = Z_{=} \cup Z_{\neq}$ with $z' \in Z_{=}$ such that $b(\eta(z')) = b(\eta(z))$, and instances $z' \in Z_{\neq}$ such that $b(\eta(z')) \neq b(\eta(z))$. We further consider the problem of imbalanced distributions in Z, which may lead to weak decision trees. Class balancing between the two partitions is achieved by adopting the SMOTE Chawla et al. (2002) procedure if the proportion of the minority class is less than a predefined threshold τ.

4.2 Local latent rules and explanation extraction

Given Z and $Y = b(\tilde{Z})$, XSPELLS builds a latent decision tree *ldt* (line 5) acting as a local surrogate of the black box, i.e., being able to locally mime the behavior of b. XSPELLS adopts decision trees because decision rules can be naturally derived from a root-to-leaf path (Guidotti et al. 2019a). Indeed, the premise p of a rule $r = p \rightarrow y$ is the conjunction of the split conditions from the root to the leaf of the tree that is followed by features in z. This approach is a variant of LORE (see Sect. 3.1) but in a latent feature space. The consequence y of the rule is the class assigned at that leaf[1].

Given a text x, the explanations returned by XSPELLS are of the form $\xi = \langle E, C, W \rangle$, where: $E = \{e_1^x, \ldots, e_u^x\}$ is the set of *exemplars* ($b(e_i^x) = b(x)\ \forall i \in [1, u]$); $C = \{c_1^x, \ldots, c_v^x\}$ is the set of *counter-exemplars* ($b(c_i^x) \neq b(x)\ \forall i \in [1, v]$); and $W = W_E \cup W_C$ is a union of the set W_E of the h most frequent words in exemplars E and of the set W_E of the h most frequent words in counter-exemplars C. Here, u, v, and h are parameters that can be set in XSPELLS. Exemplars are chosen starting from the latent instances in Z which satisfy both the premise p and the consequence y of the rule $r = p \rightarrow y$ above, namely the instances $z' \in Z$ that follow the same path as z in the decision tree, and such that the $b(\eta(z')) = y$. We visualized this in Fig. 2. By construction, elements in the same leaf as the instance to explain are candidate exemplars. The u instances z' closest to z are selected, using Euclidean distance. They are decoded back to the text space $\eta(z')$ and included in E. Counter-exemplars are chosen starting from the latent instances $z' \in Z$ which do not satisfy the premise p and

[1] In theory, it might happen that $y \neq b(\eta(z)) = b(x)$, namely the path followed by z predicts a label different from $b(x)$. In our experiments, this never occurred. In such cases, XSPELLS reboots by generating a new neighborhood and then a new decision tree.

such that $b(\eta(z')) \neq b(x)$. Let us call them the set $\mathcal{A}$ of admissible counter-exemplars. The v instances in $\mathcal{A}$ closest to z are chosen. They are decoded back to the text space $\eta(z')$ and included in C. We call such an approach *distance-based counter-exemplar selection*. Again, this is visualized in Fig. 2.

4.3 Diversity

Consider the set $\mathcal{A}$ of admissible counter-exemplars. The distance-based strategy of selecting the v instances in $\mathcal{A}$ closest to z may return counter-exemplars too similar to each other. Here, we consider the problem of improving diversity of the returned instances by proposing a *diversity-based counter-exemplar selection*. The presentation is for counter-exemplars, but the approach applies to exemplars simply by considering as admissible set the instances of Z satisfying the premise and the consequence of the rule $p \rightarrow y$.

There are several ways of formulating diversity requirements. For example, Mothilal et al. (2020) combines the search for counter-factuals (in tabular instances) with diversity maximization by resorting to gradient descent for solving an optimization problem. In our case, we have a somehow simpler problem, since the set of admissible counter-factuals is given. We formulate our problem as maximizing a function over subsets of admissible counter-exemplars:

$$\underset{S \subseteq \mathcal{A} \wedge |S| \leq v}{argmax} \; h_z(S) \tag{1}$$

where $h_z(S)$ is an objective function to be optimized under a size constraint for S. We write z as sub-script to denote that $h_z(S)$ depends also on the instance z. We choose the following objective function:

$$h_z(S) = |\bigcup_{a \in S} nn_k(a)| - \lambda \sum_{a \in S} dist(a, z) \tag{2}$$

which maximizes the difference between the coverage of the k-nearest instances of counter-exemplars (a measure of *diversity*) and the total distance of counter-exemplars from z (a measure of *proximity*) regularized by a parameter λ. The distance function *dist*, which is also used in the k-nearest neighborhood function $nn_k()$, is the Euclidean distance between min-max normalized vectors. An efficient greedy algorithm with formal guarantees can be devised for the function above. For details, see Appendix 1.

Let us discuss here how to set λ. Let k be the number of nearest neighbors considered in $nn_k(a)$ (including a itself) and let $a_0 \in \mathcal{A}$ be the closest instance to z, i.e., such that the distance $d_0 = dist(a_0, z)$ is minimal. We have that $h_z(\{a_0\}) = k - \lambda \cdot d_0$, where k is the gain of adding a_0 to the empty set, and $\lambda \cdot d_0$ the cost. In order to have at least a_0 selected, it must hold that $h_z(\{a_0\}) > 0$, hence $\lambda < k/d_0$. Since $k \geq 1$, we conservatively set:

$$\lambda = 1/(2d_0) \tag{3}$$

See Appendix 1 for a simulation on the dependency of such λ over k.

Table 1 Dataset descriptions and data partitions

Dataset	No. Instances	Avg. no. Words	Train (test) size Black box	Tuning (explanation) size BVAE	OVAE
Hate	5593	20.36	4194 (1399)	1399 (1399)	1049 (350)
Polarity	10,660	21.64	7995 (2665)	2665 (2665)	1998 (667)
Youtube	1778	21.01	1378 (400)	400 (400)	302 (98)
Liar	12,788	18.38	9591 (3197)	3197 (3197)	2397 (800)
Question	5891	8.76	4418 (1473)	1473 (1473)	1104 (369)

5 Experiments

In this section, we illustrate a qualitative and quantitative experimental analysis of faithfulness, usefulness, stability and diversity properties of XSPELLS explanations. The XSPELLS system has been developed[2] in Python. It relies on the CART decision tree algorithm as provided by the `scikit-learn` library, on a VAE implemented with the `keras` library, as well as on Optimus[3], a VAE that is pretrained on a large text corpus Li et al. (2020), specifically from Wikipedia.

5.1 Experimental settings

Datasets. We conduct experiments on a total of five datasets covering a wide array of problems within the short text classification domain. These include sentiment classification, hate speech identification, spam detection, fake news detection and question classification (Li et al. 2020; Zhou et al. 2020). The *hate speech dataset* (`hate`) Davidson et al. (2017) contains tweets labeled as hate, offensive or neutral. Here, we focus on the 1430 tweets that belong to the *hate* class, and on the 4163 tweets of the *neutral* class. The *polarity dataset* (`polarity`) Pang and Lee (2005) contains 10,660 tweets about movie reviews. Half of these tweets are classified as *negative* reviews, and the other half as *positive* ones. The third dataset we look at contains comments from five YouTube videos (`youtube`) Alberto et al. (2015), either classified as spam or ham (i.e., not spam). The final labels are *no spam* and *spam* The *liar dataset* (`liar`) Wang (2017) contains 12,788 manually annotated short statements with regards to whether they contain false or real information and are taken from politifact.com. This dataset contains six classes ranging from utterly false news to completely real news. Following related work Alhindi et al. (2018), we merge the classes to convert the task to a binary classification one. The labels are named *fake news* and *real news* respectively. Finally, the *question dataset* (`question`) Li and Roth (2002) contains questions categorized into six different semantic classes. To turn the problem into binary classification, we considered a "1 vs rest" classifier. where the label defined as 1 is the one with the largest number of instances. This class label is *entity* and it reflects questions that are about a specific entity. Thus, the two emerging labels are named *entity* and *all other classes* All these datasets are remarkable examples where a black box approach is likely to be used to remove or flag posts or to even ban users, possibly in an automated way, or

[2] The source code of XSPELLS is available at: https://github.com/lstate/X-SPELLS-V2.

[3] https://github.com/ChunyuanLI/Optimus.

Table 2 Black box model accuracy and MRE of BVAE and OVAE

Dataset	Accuracy		MRE	
	RF	DNN	BVAE	OVAE
`Hate`	0.92	0.87	0.26 ± 0.15	0.95 ± 0.08
`Polarity`	0.67	0.73	0.58 ± 0.35	0.95 ± 0.07
`Youtube`	0.92	0.91	0.41 ± 0.39	0.76 ± 0.30
`Liar`	0.61	0.59	0.59 ± 0.39	0.92 ± 0.09
`Question`	0.85	0.84	0.33 ± 0.34	0.80 ± 0.20

to recommend topics based on user questions, possibly leading to wrong answers. Such extreme actions risk to hurt the free speech rights of people. Explanations of the black box decisions are then of primary relevance both to account for the action and to test or debug a black box. Datasets details are reported in Table 1 (left). We will experiment with binary (short) text classification. This is not a limitation of the proposed approach, which is able to deal with multi-class black boxes. Rather, the problem traces back to the limited size of the available datasets, which have to be partitioned into different sets for training black boxes, tuning VAEs, and testing the approach, as described next. Working with more than two classes makes it harder to train black boxes with acceptable performances.

Experimental scenarios. For each dataset, we use 75% of the available data for training a black box machine learning classifier. We call such a subset the *training set*. The remaining 25% of data, called the *test set*, is used for training/tuning the autoencoder and for explaining the black box decisions. More specifically, the test set is further split into 75% for training/tuning the autoencoder, called the *tuning set*, and 25% as the set of instances to explain, called the *explanation set*. All splits are stratified. We call this scenario the *standard case*. The VAE that we implemented, when trained on the tuning set, will be denoted as SVAE (for "standard VAE"). Also, we call OVAE the Optimus (Li et al. 2020) VAE, when it is fine-tuned using the tuning set. We will also experiment with an alternative scenario, which simulates the ideal situation when the autoencoder has a perfect knowledge of the (domain of the) instances to explain. In this scenario, both the tuning set and the explanation set are fixed to the whole test set. In particular, we call BVAE (for "best case VAE") the VAE we implemented, when it is trained on the whole test set. This scenario is realistic during the development cycle of a black box, when the developers want to debug its decisions on a set of instances representative of the population. Finally, for computational efficiency reasons, in both scenarios we selected 100 instances (98 for the youtube dataset in the case of the standard scenario) from the explanation set to calculate the quantitative evaluation metrics (fidelity, usefulness, stability, and diversity).

Black boxes. We trained and explained the following black box classifiers: Random Forests (Tan et al. 2016) (RF) as implemented by the `scikit-learn` library, and Deep Neural Networks (DNN) as implemented with the `keras` library. For the RF, we transformed texts into their TF-IDF weight vectors (Tan et al. 2016), after removing Twitter stop-words such as "rt", hashtags, URLs and usernames. For the `youtube` dataset we additionally removed emojis and texts that hold more than 140 characters (maximum length of a tweet until 2017). A randomized cross-validation search was then performed for parameter tuning. Parameters for RF models were set as follows: 100 decision trees, *Gini* split criterion,

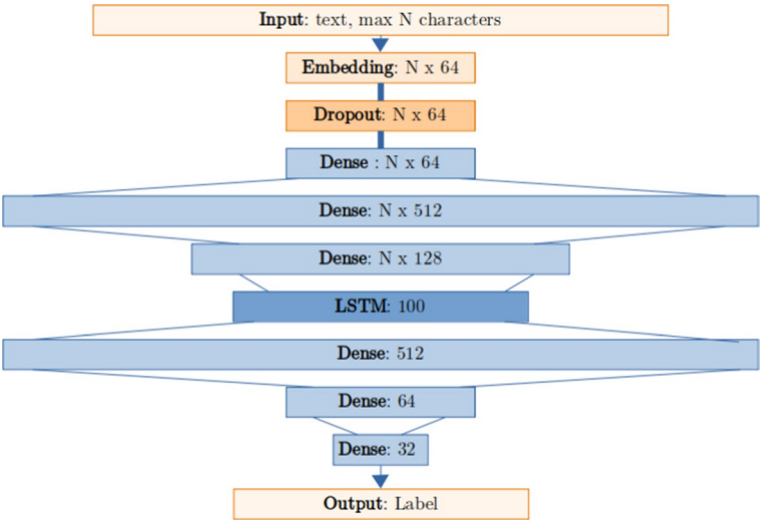

Fig. 3 Architecture of DNN black boxes. Bold text: type of layer. Numbers: layer size

$\sqrt{m}$ random features where m is the total number of features; no limit on tree depth. The DNNs adopted have the architecture shown in Fig. 3. The first layer is a dense embedding layer. It takes as input a sparse vector representation of each text (subject to same pre-processing steps as for the RF, without the TF-IDF representation) obtained by using a Keras tokenizer[4] to turn the text into an array of integers and a padder so that each vector has the same length. This way, we allow the network to learn its own dense embeddings of size 64. The first embedding layer is followed by a dropout layer at 0.25. Afterwards, the DNN is composed by three dense layers with sizes 64, 512 and 128. The central layer is an LSTM (Hochreiter and Schmidhuber 1997) that captures the sequential nature of texts and has size 100. After that, there are three dense layers with sizes 512, 64 and 32. The dense layers adopt the *ReLu* activation function. Finally, the *sigmoid* activation function is used for the final classification. We adopted *binary cross-entropy* as loss function and the *Adam* optimizer. We trained the DNN for a maximum of 100 epochs, or until the performance of the model stops improving on a held out validation dataset for more than 2 epochs. Classification accuracies are reported in Table 2.

VAEs. Here, we describe the structure and the training parameters of the VAEs we developed. We distinguish the best scenario (BVAE) from the standard scenario (SVAE). The performances of SVAE were considerably worse than BVAE (see later on). For this reason, we decided to adopt the pre-trained Optimus VAE in the standard scenario (OVAE). Experiments in the following subsections will involve only BVAE and OVAE. Table 2 reports the *Mean Reconstruction Error* (MRE) calculated as the average cosine similarity distance between an instance in the explanation set and its reconstructed text when converted to TF-IDF vectors. Since the OVAE generates texts that are similar in meaning but more diverse in words, the MRE is for all datasets higher (worse) compared to the BVAE. However, this does not mean the generated texts are not useful.

[4] https://keras.io/preprocessing/text.

Table 3 MRE of SVAE

	Training epochs	
Dataset	100	30
`Hate`	0.96 ± 0.09	0.95 ± 0.08
`Polarity`	0.96 ± 0.04	0.95 ± 0.06

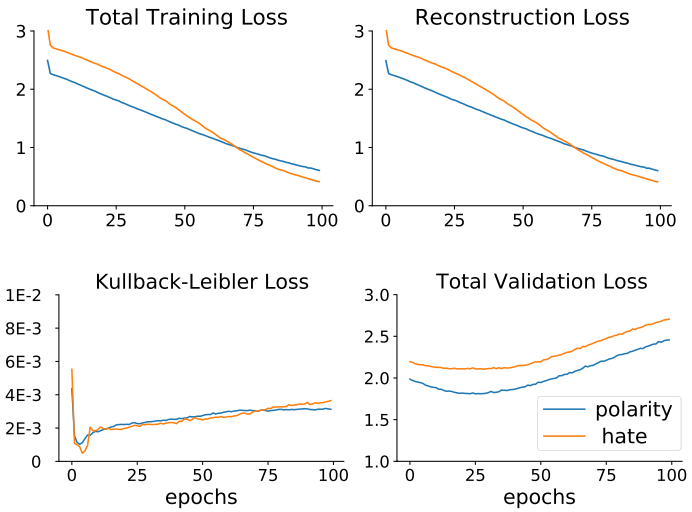

Fig. 4 Training and validation error for SVAE. The total training loss can be further deconstructed in the reconstruction loss and the Kullback-Leibler loss. The validation error increases after roughly 30 epochs

Both BVAE and SVAE have their encoders ζ and decoders η implemented as a single LSTM layer. This design choice is inspired by the widely successful usage of seq2seq models for texts viewed as sequences of characters (Sutskever et al. 2014). We fed the text into the VAE using a one-hot vectorization. In order to have control over the dimensionality of the input, and a fortiori for the computational resources required for training the VAE, we kept a maximum of 5000 distinct words for each dataset (this actually affects the `polarity` and `liar` datasets). This vocabulary was extended with the 1,000 most common English words[5] to provide to the VAE also knowledge about unseen words with respect to the training set. The resulting size of the input tensors was $33 \cdot 4947{=}163,251$ for the `hate` dataset, $49 \cdot 5287{=}259,063$ for the `polarity` dataset after Twitter stopwords removal for both of the previous datasets, $32 \cdot 1666{=}53,312$ for the `youtube` dataset, $67 \cdot 5319{=}356,373$ for the `liar` dataset, and $35 \cdot 3966{=}138,810$ for the `question` dataset. The numbers above represent the maximum text length (number of words) and the number of distinct words considered. These dimensionalities are manageable because the input consists of short texts. We considered $k{=}500$ latent features[6] for all datasets. The

[5] https://1000mostcommonwords.com/1000-most-common-english-words/

[6] Experiments (not reported due to lack of space) show that $k = 500$ is a good compromise between MRE and training time when varying $k \in \{100, 250, 500, 1000, 2500\}$.

number of training epochs of the BVAE are 200 for the `hate`, the `youtube` and `question` dataset, 250 for the `polarity` and the `liar` dataset. These numbers were chosen after observing the epoch at which the reconstruction loss stabilized.

We proceeded similarly for SVAE. As a first approach, we trained a VAE for 100 epochs for the `hate` and `polarity` datasets. Figure 4 highlights that the validation error has a minimum at roughly 30 epochs for both datasets. As expected, the MRE values after 30 epochs of training are lower (better) than after 100 epochs, see also Table 3. However, these values are considerably worse than for the BVAE (cf. Table 2). We also observed that the Kullback-Leibler loss term is much smaller than the reconstruction loss (see Fig. 4), a phenomenon also know as KL vanishing. Therefore we experimented with Kullback-Leibler annealing as e.g. described in (Bowman et al. 2016), but could not obtain good enough results to continue with the subsequent experiments using SVAE.

Finally, fine-tuning of the OVAE was done for 10 epochs and on each dataset separately. Several variants of the pretrained autoencoder are released: we used the version with 768 latent dimensions and set the parameter $\beta = 0.5$ (weighting the Kullback-Leibler loss term). This is a trade-off between reconstruction quality of the sentences and smooth generation of new data instances, i.e., between a standard autoencoder and a full variational autoencoder. We disabled SMOTE for class balancing of sentences generated by the OVAE.

Hyper-parameters. We set the following XSPELLS hyper-parameters. The neighborhood generation *neighgen* is run with N=600, n=200, τ=40%. For the latent decision tree, we used the default parameters of the CART implementation[7]. Finally, with regards to the explanation hyper-parameters, we set u=v=5 (counter-)exemplars, and h=5 most frequent words for exemplars and for counter-exemplars. Finally, unless otherwise stated, we adopt the diversity-based counter-exemplar selection, setting $k = 5$ for the $nn_k()$ function in (2).

Compared approaches. We compare XSPELLS against LIME (Ribeiro et al. 2016), and, for the qualitative part, also against ANCHOR. We cannot compare against SHAP (Lundberg and Lee 2017) because it is not immediate how to adapt it to text classifiers. As discussed in Sect. 2, we do not compare to adaptions of IntGrad (Sundararajan et al. 2017) or LRP (Bach et al. 2015), originally proposed for image classifiers, to explain text classifiers because they are not model-agnostic.

5.2 Qualitative evaluation

In this section, we qualitatively evaluate the explanations by XSPELLS and contrast them with the ones by LIME and ANCHOR. Example texts labeled as *hate*/*negative*/*spam*/*fake news*/*entity* are shown in Table 4 to Table 8. Each table contains three exemplars and two counter-exemplars for both the BVAE and the OVAE.

Table 4 reports an example for the `hate` dataset. This dataset is only used for research purposes, we clearly distance from any form of discrimination expressed in there. In particular, we avoid reproducing extremely offensive words by writing “N" instead of the N-word. Explanations based on the BVAE for DNN and RF are clearly offensive when it comes to exemplars, while the opposite is true with counter exemplars. Exemplars in the case of OVAE are instead more diverse, whilst counter-exemplars of DNN and RF coincide

[7] The `scikit-learn` library does not implement advanced features such as post-pruning (Ruggieri 2012), feature selection (Ruggieri 2019), or non-greedy induction (Bertsimas and Dunn 2017), that are designed to produce smaller and more accurate decision trees. In turn, these properties could lead to explanations closer to the instance under analysis, and to latent decision tree with higher fidelity.

Table 4 Explanations returned by XSPELLS for texts labeled as *hate* in the `hate` dataset for BVAE and OVAE. Three exemplars (E) and two counter-exemplars (C) for each tweet. Relative word frequencies over $u = v = 5$ (counter-)exemplars in parenthesis

Text	Fat ass hoe holding up the machine			
	(counter-)exemplars	E/C	$W_{=}$	$W_{\neq}$
`BVAE RF`	she is a cunt	E	bitch (.17)	one (.10)
	truth only a faggot would turn down	E	ass (.11)	inmate (.0.05)
	bitch ass N be hating on us	E	N (.11)	sums (.05)
	do you think the guinea pig was okay	C	hating (.11)	return (.05)
	an inmate sums up what return means	C	cunt (.06)	means (.05)
`BVAE DNN`	fucking with these faggot N are	E	N (.11)	let (.06)
	spending money on these hoes N	E	subtweet (.05)	coons (.06)
	subtweet me one more time you diy	E	faggot (.05)	overpriced (.06)
	let the coons be great	C	hoes (.05)	great (.06)
	only the field ready for today's game	C	diy (.05)	complete (.06)
`OVAE RF`	N don t let a dude treat you like a yellow starburst	E	ass (.08)	loaded (.05)
	justmilz delete that stupid bitches funky monkey	E	N (.04)	suit (.05)
	i highly doubt it faggot joe	E	let (.04)	getting (.05)
	it s a loaded suit i m getting the worm	C	dude (.04)	worm (.05)
	consider this racca the buccawg	C	treat (.04)	consider (.05)
`OVAE DNN`	talking heads ghetto trash like faggot masochist	E	like (.08)	loaded (.5)
	suck ass bird shit my cue is to trap u	E	faggot (.08)	suit (.05)
	this weird baby sitting on a curb looking like a faggot	E	suck (.08)	getting (.05)
	'it s a loaded suit i m getting the worm	C	talking (.04)	worm (.05)
	consider this racca the buccawg	C	heads (.04)	consider (.05)

in this case. In both cases, top words in the exemplar class tend to be very strong/offensive and they can be clearly associated to hate speech.

Table 5 shows an example for the `polarity` dataset. A clear difference between both BVAE and OVAE is the sentences length. Instances generated by the OVAE are in general longer, nevertheless meaningful. This is emphasized by a lower relative word frequency. All generated exemplar sentences are different to each other, whereas one counter-exemplar is picked up both for DNN and RF in the case of OVAE. Instances explained by the BVAE contain the top word "bad" in their factual (negative) class with a high relative word frequency[8]. It can be clearly assigned to a negative sentiment. In this example, we have two cases where the black box prediction was not correct, i.e., a misclassification occurred. This is the case for the BVAE RF and OVAE DNN. We can see that exemplars and counter-exemplars do not contrast very well against each other, i.e., do not clearly distinguish different classes. This is consistent, for instance, with a possible overfitting of the black box in the neighborhood of the instance to explain. More specifically, in the case of the

[8] For space restrictions, Table 4 to Table 6 report only two counter-exemplars. However, frequencies are computed over the $v = 5$ counter-exemplars returned by XSPELLS. This explains frequent words not appearing in the two reported sentences.

Table 5 Explanations returned by XSPELLS for texts labeled as *negative* in the `polarity` dataset for BVAE and OVAE. Three exemplars (E) and two counter-exemplars (C) for each tweet. Relative word frequencies over $u = v = 5$ (counter-)exemplars in parenthesis

Text	You can practically hear george orwell turning over			
	(counter-)exemplars	E/C	$W_{=}$	$W_{\neq}$
`BVAE RF`	- what s most offensive isn t the movie is a bad experience	E	bad (.18)	movies (.08)
	- for most movies minutes is sho but this one feels	E	offensive (.13)	go (.08)
	- what will most likely turn out to be the most movie	E	really (.06)	minutes (.04)
	- for most movies minutes it deals with a memorable	C	appealing (.06)	deals (.04)
	- what s next rob schneider dana carvey and michelle	C	characters (.06)	memorable (.04)
`BVAE DNN`	- it smacks of purely commercial motivation	E	bit (.18)	want (.1)
	- what the movie is a bit it s an actor made on a strong	E	funny (.06)	empire (.1)
	- what the movie is a bit it not not to be very funny	E	actor (.06)	lacks (.05)
	- what empire lacks in depth it makes up for with characters	C	made (.06)	depth (.05)
	- if you want to appear will suck the right to keep	C	strong (.06)	makes (.05)
`OVAE RF`	- sometimes it does go on forever and the end result is a tear numbingly dull porait of cluelessness	E	also (.05)	chew (.07)
	- barney s throwaway scenario is not only laugh out loud but it s also not only boldly funny	E	sometimes (.03)	spos (.04)
	- a puzzling coincidence spoof of a movie that appears destined to stand in future years as a journalistic footnote	E	go (.03)	sincere (.04)
	- spos a sincere emotional burst of color	C	forever (.03)	emotional (.04)
	- if you have a soft patience for jabs it works great in junk like this	C	end (.03)	burst (.04)
`OVAE DNN`	- a charmer from belgium	E	cherry (.06)	great (.06)
	- cherry on cherry lightly you can taste the bitter taste in the mouth	E	taste (.06)	soft (.03)
	- dark satire disguised as history	E	charmer (.03)	patience (.03)
	- if you have a soft patience for jabs it works great in junk like this	C	belgium (.03)	jabs (.03)
	- this formulaic spos movie is hampered by the flabby preposterous lame tangibility of its title	C	lightly (.03)	works (.03)

Table 6 Explanations returned by XSPELLS for texts labeled as *spam* in the `youtube` dataset for BVAE and OVAE. Three exemplars (E) and two counter-exemplars (C) for each tweet. Relative word frequencies over $u = v = 5$ (counter-)exemplars in parenthesis

Text	Check out this video on youtube			
	(counter-)exemplars	E/C	$W_{=}$	$W_{\neq}$
`BVAE RF`	- please suscribe i am bored of 5 subscribers	E	please (.11)	hey (.10)
	- subscribe me if u love eminem !‘ end ?‘	E	suscribe (.11)	subscribe (.0.05)
	- please subscribe me for the wet seal	E	subscribe (.05)	u (.05)
	- subscribe i ll u are watching in	C	bored (.05)	watching (.05)
	- oh my god go to 1 billion of replay	C	5 (.05)	oh (.05)
`BVAE DNN`	- subscribe and if you are watching	E	subscribe (.24)	hear (.14)
	- please subscribe me for u wet i will	E	please (.24)	rather (.07)
	- please subscribe me for u longer	E	u (.12)	propa (.07)
	- i m rather hear some propa explicit	C	wet (.12)	explicit (.07)
	- what i m shufflin as if gangnam style	C	watching (.06)	shufflin (.07)
`OVAE RF`	- best song for dancing	E	omg (.14)	youtube (.16)
	- imfao is epic music video	E	best (.11)	check (.11)
	- i like the p	E	song (.07)	u (.11)
	- like this comment on youtube	C	dancing (.07)	love (.11)
	- hey everyone check out my new song at youtube	C	eh (.07)	omg (.06)
`OVAE DNN`	- check out my black dubstep video	E	check (.12)	hi (.10)
	- follow em on twitter for the latest insta video	E	video (.08)	cross (.10)
	- yeah check out this playlist on youtube	E	black (.04)	eminem (.10)
	- the rules section shuffle 5 song videos	C	dubstep (.04)	rules (.05)
	- eminem beat eminem in the cross fire	C	channel (.04)	section (.05)

BVAE DNN, we can see that exemplars seem to praise the movie, while counter-exemplars criticize it. This is in-line with the fact that the label was in-correctly predicted as positive.

Generated sentences for the `youtube` dataset are rather short, as displayed in Table 6. Both VAEs are able to pick up key words from the spam comments as most frequent words in their explanations, e.g. “subscribe”, “check” or “video”. This is mirrored by the exemplar sentences from the BVAE, which are very similar to each other. For the OVAE, sentences are more diverse, but transport a similar meaning. For the BVAE, most frequent counter-factual words do not show a pattern whereas in the OVAE example for RF, we can repeatedly observe the words “love” and “song”. There is also a misclassification in this example in the case of the OVAE RF. It can be notable that the exemplars and counter-exemplars have switched in the meaning that they convey. Exemplars show interest, while counter-exemplars are spammy.

In Table 7 are the results for the `liar` dataset. In this example, counter exemplar sentences seem to make outlandish claims, while exemplar sentences seem more grounded. In the case of the BVAE, the top words don’t contrast each other very well. In the OVAE, we can see that more negative words appear in the exemplar case.

Finally, the results regarding the `question` dataset, can be seen in Table 8. In this example, both black boxes for the OVAE have misclassified the sentence. In the case of the BVAE, we can see that the exemplars correctly show that the answer to the question

Table 7 Explanations returned by XSPELLS for texts labeled as *fake news* in the `liar` dataset for BVAE and OVAE. Three exemplars (E) and two counter-exemplars (C) for each tweet. Relative word frequencies over $u = v = 5$ (counter-)exemplars in parenthesis

Text	Why would our president close the embassy to the vatican hopefully it is not retribution for catholic organizations opposing obamacare			
	(counter-)exemplars	E/C	$W_{=}$	$W_{\neq}$
`BVAE RF`	- medical grade marijuana not get that patient high no matter what level	E	get (.06)	first (.03)
	- with recall organizers said no federal stimulus law will cost more	E	state (.06)	principles (.03)
	- right now the top 1 percent be paid out their state lose their state	E	principles (.03)	allowed (.03)
	- more people are losing their insurance due to obamacare	C	allowed (.03)	fellow (.03)
	- with those first principles it allowed a fellow like me to get in	C	first (.03)	like (.03)
`BVAE DNN`	- mccain says the price of a gas tax holiday would be able to change	E	get (.06)	right (.11)
	- with those first principles it allowed a fellow like me to get in	E	right (.03)	top (.06)
	- why do the democrats voted an or his farm per every health care costs	E	top (.03)	30 (.06)
	- right now our depament charges taxpayers between 30 000 and 40 000	C	third (.03)	people (.06)
	- with obamacare and that was re going to some 7 000 a day on or so	C	florida (.03)	day (.06)
`OVAE RF`	- lizbeth benacquisto broke election law 105 times hiding special interest contributors	E	agency (.04)	says (.06)
	- obama won a landslide in florida under democratic control	E	women (.04)	people (.04)
	- sen sherrod brown voted to depo the federal stimulus and continues to do so	E	gop (.02)	earmarks (.04)
	- 4 billion people lost their federal jobs during dc slash	C	acknowledged (.02)	tax (.02)
	- more than a decade ago freddie mac york banned earmarks from earmarks	C	banned (.02)	breaks (.02)
`OVAE DNN`	- trump allies say he treats americans very seriously and sends them free passes	E	trump (.05)	says (.06)
	- romney voted to fund abortion in the womens health care law	E	rapist (.05)	tax (.04)
	- georgia has one of the most progressive tax codes in the world	E	allies (.02)	breaks (.04)
	- kathleen brendan wiped clean air records in 2010	C	say (.05)	earmarks (.04)
	- says pass michelle nunn voted to give special tax breaks for american oil companies	C	treats (.02)	wealth (.04)

Table 8 Explanations returned by XSPELLS for texts labeled as *entity* in the `question` dataset for BVAE and OVAE. Three exemplars (E) and two counter-exemplars (C) for each tweet. Relative word frequencies over $u = v = 5$ (counter-)exemplars in parenthesis

Text	What is money made of			
	(counter-)exemplars	E/C	$W_{=}$	$W_{\neq}$
`BVAE RF`	- what do i call a newborn kangaroo	E	lengths (.13)	mean (.13)
	- what are some chemical properties of mendelevium	E	pearl (.06)	dot (.0.05)
	- what did the yalta conference lead to	E	necklaces (.06)	word (.05)
	- what does the word fonight mean in maryland	C	yalta (.06)	g (.05)
	- what does the dot on the letter i mean	C	conference (.06)	stand (.05)
`BVAE DNN`	- what are the components of polyester	E	warlock (.07)	wear (.07)
	- what are the colors of the german flag	E	wear (.07)	father (.07)
	- what did warlock wear on his forehead	E	colors (.07)	bras (.07)
	- why do girls have to wear training bras	C	forehead (.07)	manufacture (.07)
	- what did andy hardy s father do	C	polyester (.07)	fair (.07)
`OVAE RF`	- what s the third largest city in america	E	city (.08)	cruel (.14)
	- what new york city landmark has 168 steps to its crown	E	third (.04)	man (.04)
	- how many acres of space does a tennis ball occupy	E	largest (.04)	dog (.04)
	- what is the most expensive restaurant in myle beach	C	america (.04)	cage (.04)
	- what city s newspaper boasts the most famous prints	C	much (.04)	doubles (.04)
`OVAE DNN`	- what is the literal meaning of thank you notes	E	biggest (.06)	largest (.18)
	- where does american postage go to college	E	city (.06)	cathedral (.18)
	- what s the longest english composer record	E	april (.06)	world (.12)
	- what s the name given to a group of geese	C	1998 (.06)	festival (.06)
	- what s the largest cathedral in the us	C	capital (.06)	rebranded (.06)

asked is with respect to an entity, while the counter-exemplars ask either for definitions or explanations. This is not the case for the OVAE, as the exemplars are making geographical inquiries or asking questions regarding descriptions.

We contrast explanations of XSPELLS against those from LIME reported in Table 9. The texts from all the datasets except the `liar` text are correctly explained by LIME. For the `youtube` comment, for instance, we observe that the most important words are for both black box models “check" and “out", but receive no further information. Overall, since LIME extracts words from the text under analysis, it can only provide explanations using such words. On the contrary, the (counter-)exemplars of XSPELLS consist of several and diverse texts which are (most of the times) close in meaning to the text under analysis, but including different wordings that help the user better grasp the reasons behind black box decision. A similar conclusion holds for ANCHOR, which produces as explanations a subset of the words in the text – called *anchors*. For the `youtube` example and the RF black box, it returns the anchors “out" and “youtube", with a precision of 100%. This means that when both words

Table 9 Explanations returned by LIME for texts labeled as *hate* (`hate` dataset), *negative* (`polarity` dataset), *spam* (`youtube` dataset), *fake news* (`liar` dataset), or *entity* (`question` dataset). LIME word importance in parenthesis

Tweet	Top features			
fat ass hoe holding up the machine	`RF hate`	ass (-0.43)	`DNN hate`	ass (-0.21)
		hoe (-0.13)		holding (-0.14)
		fat (-0.04)		fat (-0.10)
		the (0.03)		machine (0.07)
		up (-0.02)		up (0.04)
you can practically hear george orwell turning over	`RF polarity`	you (0.09)	`DNN polarity`	george (-0.13)
		over (-0.06)		hear (0.13)
		can (0.02)		turning (0.11)
		orwell (-0.02)		you (0.10)
		practically (-0.02)		can (0.03)
check out this video on youtube	`RF youtube`	check (0.39)	`DNN youtube`	check (0.22)
		out (0.31)		out (0.22)
		on (0.04)		youtube (-0.07)
		youtube (0.03)		on (0.01)
		this (-0.03)		this (0.01)
why would our president close the embassy to the vatican hopefully it is not retribution for catholic organizations opposing obamacare	`RF liar`	opposing (0.14)	`DNN liar`	the (0.07)
		close (0.05)		our (0.06)
		obamacare (-0.04)		president (-0.05)
		the (-0.03)		obamacare (-0.04)
		not (0.03)		would (0.03)
what is money made of	`RF question`	made (-0.30)	`DNN question`	made (-0.35)
		of (-0.22)		money (-0.29)
		is (0.12)		of (-0.28)
		what (-0.11)		what (-0.24)
		money (-0.01)		is (0.11)

are present in a randomly generated neighbor instance, then the black box labels the text as *spam*. ANCHOR outputs also exemplar texts, obtained from the random generation of the neighborhood. Differently from LIME, however, the randomness can be driven by distance in some embedding space between a word in the text and others words with the same POS tag. On our sample text, using the embedding provided by the BERT transformers Devlin et al. (2019), the following top five exemplars are returned[9] for the `youtube` text:

- cut out the video on youtube
- check out this page on youtube
- check out movie trailers on youtube
- file out play list on youtube
- get out loud online on youtube

These exemplars have the same syntactic structure of the text to explain (and the same size). On the positive side, this leads to syntactically valid texts, yet not necessarily meaningful (as the last two ones). This is not true, however, for tweets, which lack the formal structure of sentences, and then relying on POS tagging is not effective. For instance, the exemplars returned for the sample text of the `polarity` dataset are meaninginglessness. Moreover, exemplars of ANCHOR do not show much variation with respect to the text to explain. Examples of using ANCHOR on the rest of the datasets can be found on our Github repository[10].

5.3 Fidelity evaluation

Surrogate models adopted by explanation approaches should mimic the decision logic of the black box. Here, we show that XSPELLS performs better than LIME under such a perspective. We evaluate the *faithfulness* (Freitas 2013; Guidotti et al. 2019b) of the latent decision tree adopted by XSPELLS by measuring how well it reproduces the behavior of the black box b in the neighborhood of the text x to explain – a metric known as *fidelity*. Let Z be the neighborhood of z in the latent space generated at line 2 of Alg. 1 and *ldt* be the surrogate decision tree computed at line 5. The fidelity metric is $|\{y \in Z \mid ldt(y) = b(\eta(y))\}|/|Z|$, namely the accuracy of *ldt* assuming as ground truth the black box. The fidelity values over all instances in the explanation set are aggregated by taking their average and standard deviation.

We compare XSPELLS against LIME, which adopts as surrogate model a linear regression over the feature space of words and generates the neighborhood using a purely random strategy. Table 10 reports the average fidelity and its standard deviation for LIME and XSPELLS in its variants depending on the type of VAE adopted. We notice how on every dataset and on every black box XSPELLS fidelity is markedly higher than the one of LIME independently from the VAE adopted.

[9] Full details for all sample texts are available at the Github repository of XSPELLS.

[10] https://github.com/lstate/X-SPELLS-V2/blob/main/experiments/anchor_text.ipynb.

Table 10 Mean and standard deviation of fidelity. The higher the better

	RF xspells		DNN xspells	
	BVAE	OVAE	BVAE	OVAE
Hate	0.98 ± 0.01	0.97 ± 0.01	0.98 ± 0.01	0.97 ± 0.01
Polarity	0.97 ± 0.01	0.97 ± 0.01	0.97 ± 0.01	0.97 ± 0.01
Youtube	0.98 ± 0.01	0.98 ± 0.01	0.97 ± 0.01	0.97 ± 0.01
Liar	0.98 ± 0.01	0.97 ± 0.01	0.98 ± 0.01	0.97 ± 0.01
Question	0.99 ± 0.01	0.97 ± 0.01	0.98 ± 0.01	0.97 ± 0.01
	RF lime		DNN lime	
Hate	0.71 ± 0.41		0.37 ± 0.28	
Polarity	0.40 ± 0.27		0.21 ± 0.28	
Youtube	0.27 ± 0.31		0.27 ± 0.30	
Liar	0.41 ± 0.22		0.50 ± 0.35	
Question	0.52 ± 0.30		0.37 ± 0.30	

5.4 Usefulness evaluation

In this section, we investigate on the usefulness of xspells explanations for the user. How can we evaluate the usefulness of explanations? The gold standard would require to run lab experiments involving human evaluators. Inspired by Kim et al. (2016), we provide here an indirect evaluation by means of a k-Nearest Neighbor (k-NN) classifier (Tan et al. 2016). For a text x in the explanation set, first we randomly select n exemplars and n counter-exemplars from the output of xspells. Then, a 1-NN classifier[11] is trained over such (counter-)exemplars. Finally, we test 1-NN over the text x and compare the prediction of 1-NN with the label $b(x)$ predicted by the black box. In other words, the 1-NN approximates a human in assessing the (counter-)exemplars usefulness. The accuracy computed over all x's in the explanation set is a proxy measure of how good/useful are (counter-)exemplars at delimiting the decision boundary of the black box. We compare such an approach with a *baseline* (or null) model consisting of a 1-NN trained on n texts per class label, selected randomly from the training set and not including x.

The accuracy of the two approaches are reported in Fig. 5 by varying the number n of exemplars and counter-exemplars. xspells neatly overcomes the *baseline*. The difference is particularly marked for when n is small. Even though the difference tend to decrease for large n's, large-sized explanations are less useful in practice due to cognitive limitations of human evaluators. Moreover, xspells performances are quite stable w.r.t. n, i.e., even one or two exemplars and counter-exemplars are sufficient to let the 1-NN classifier distinguish the label assigned to x in an accurate way. The use of BVAE compared to OVAE does not significatively change in the accuracy of xspells.

5.5 Stability evaluation

Let us contrast xspells with lime as per the stability of their explanations. Stability is a key requirement, which heavily impacts users' trust on post-hoc explainability methods (Rudin 2019). For local approaches, the generation of the neighborhood introduces randomness

[11] Distance function adopted: cosine distance between the TF-IDF representations.

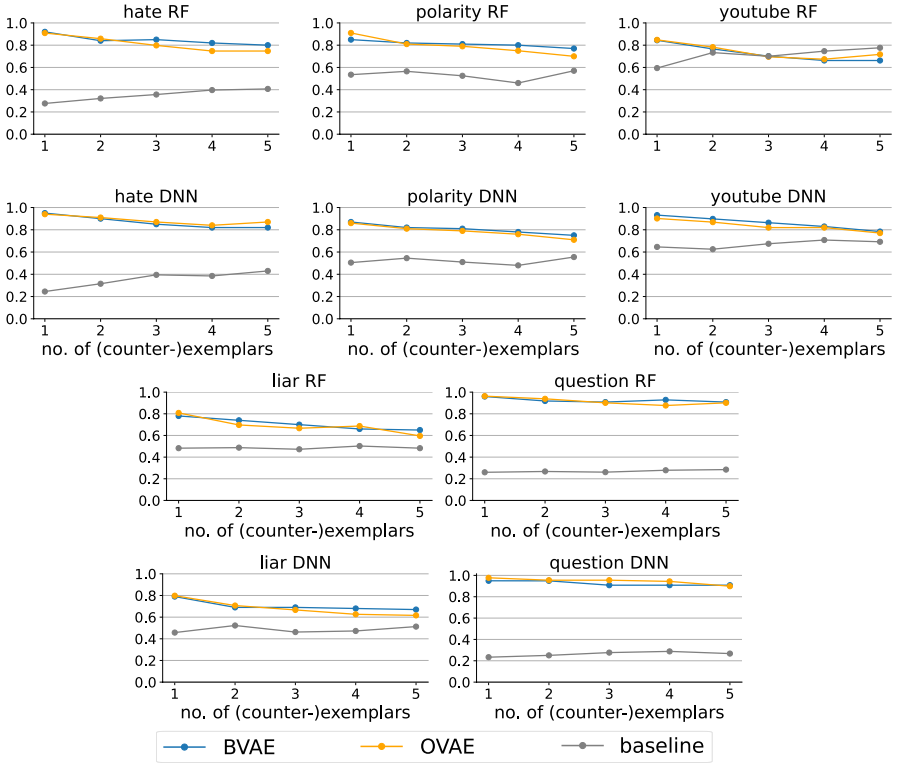

Fig. 5 Accuracy (usefulness) of xspells (with BVAE/OVAE) vs a baseline for an increasing number of exemplars and counter-exemplars adopted as neighbors. Best viewed in color

in the process, leading to different explanations for a same instance in different runs of the method, or to disproportionately different explanations for two close instances (Alvarez-Melis and Jaakkola 2018). Instability of the surrogate models learned from the neighborhood instance may exacerbate the problem Guidotti and Ruggieri (2019). Extensions of lime tackle the problem by avoiding sampling of neighbor instances in favor of training data (Zafar and Khan 2021), by adopting a denoising autoencoder to compute neighbor instances over a more robust space (Shankaranarayana and Runje 2019), or by determining the number of instances needed to statistically guarantee stability of the explanation (Zhou et al. 2021). In addition, methods using text embedding, such as ours, suffer from the variability introduced by the encoding-decoding of texts in the latent space. Several metrics of stability can be devised (Alvarez-Melis and Jaakkola 2018; Guidotti and Ruggieri 2019), either independent or specific to the explanation representation (Visani et al. 2021). A possible choice is to use sensitivity analysis with regard to how much an explanation varies on the basis of the randomness in the explanation process. We measure here stability as a relative notion, that we call *coherence*. For a given text x in the explanation set, we consider its closest text x^c and its k-th closest text x^f, again in the explanation set. A form of Lipschitz condition (Alvarez-Melis and Jaakkola 2018) would require that the distance between the explanations $e(x)$ and $e(x^f)$, normalized by the distance between x and x^f, should not be much different than the distance between the explanations $e(x)$ and $e(x^c)$, again normalized

by the distance between x and x^c. Stated in words, normalized distances between explanations should be as similar as possible. Formally, we introduce the following *coherence index*:

$$C_x = \frac{dist_e(e(x^f), e(x))/dist(x^f, x)}{dist_e(e(x^c), e(x))/dist(x^c, x)}$$

where we adopt as distance function *dist* the cosine distance between the TF-IDF representation of the texts, and as distance function $dist_e$ the Jaccard distance between the 10 most frequent words in each explanation (namely, the W set). In experiments, we set x^f to be the $k = 10$-closest text w.r.t. x. For comparison, the coherence index is computed also for LIME, with Jaccard similarity calculated between the sets of 10 words (a.k.a. features) that LIME deems more relevant.

Table 11 reports the average coherence over the explanation set. XSPELLS has a slightly worse level of coherence, yet statistically significant (p-value < 0.01) in most cases except the following ones: the `youtube` dataset and OVAE RF/DNN, of the `polarity` dataset and BVAE RF or OVAE RF/DNN, and of the `liar` dataset and BVAE RF or OVAE RF. This is expected, as LIME's explanations are subsets of the original text to explain, hence more stable across similar instances. XSPELLS trade-offs a slightly worse stability for diversity of exemplars, and, a fortiori, of the most frequent words in them. Finally, contrasting BVAE with OVAE, we observe the latter has lower mean and standard deviation values in almost all cases.

5.6 Diversity of counter-exemplars

Let us contrast here the diversity-based with the distance-based selection of counter-exemplars (see Sect. 4.3), showing that the former has a better impact on diversity of counter-exemplars. Their impact on usefulness and stability metrics are comparable (experiments not reported here for lack of space). Here, we consider their impact on the objective function $h_z()$ (see formula (2)). Since the diversity-based approach optimizes such an objective function, we expect it to perform better than the distance-based approach. The point is to quantify the degree of diversity it can capture.

Figure 6 shows the (kernel density estimates of the) distributions of $h_z(S)$ when S is the set of v counter-exemplars selected by the two strategies for each instance in the explanation set. The densities of the distance-based approach approximately resemble a normal distribution whereas the densities for the diversity-based approach are skewed towards higher values of the objective function. The difference between the two densities is statistically significant ($p < 0.01$ in the Kolmogorov-Smirnov test), irrespective of the datasets and the VAEs. This is expected, as for the diversity-based approach, the greedy algorithm adopted (see **CSG** in Appendix 1) provides a theoretical guarantee of approximating the optimal solution, while this is not the case for the distance-based approach. Finally, for the diversity-based strategy, the OVAE has a more peaked distribution than the BVAE, i.e., the objective function for the selected counter-exemplars is on average closer to the optimal solution. This can be attributed to the different latent spaces of the two VAEs.

Table 11 Mean and stdev of the coherence index $\mathcal{C}_x$. The closer to 1 the better

	RF XSPELLS		DNN XSPELLS	
	BVAE	OVAE	BVAE	OVAE
Hate	1.14 ± 0.15	1.13 ± 0.11	1.12 ± 0.12	1.14 ± 0.10
Polarity	1.10 ± 0.11	1.09 ± 0.05	1.11 ± 0.10	1.09 ± 0.05
Youtube	1.33 ± 0.48	1.19 ± 0.24	1.33 ± 0.50	1.16 ± 0.22
Liar	1.12 ± 0.12	1.09 ± 0.13	1.10 ± 0.12	1.10 ± 0.11
Question	1.24 ± 0.29	1.20 ± 0.25	1.26 ± 0.29	1.23 ± 0.22
	RF LIME		DNN LIME	
Hate	1.06 ± 0.13		1.06 ± 0.11	
Polarity	1.07 ± 0.16		1.07 ± 0.09	
Youtube	1.14 ± 0.46		1.17 ± 0.37	
Liar	1.08 ± 0.18		1.05 ± 0.11	
Question	1.10 ± 0.22		1.11 ± 0.22	

6 Conclusion

We have presented XSPELLS, a local model-agnostic explanation approach for black box short text classifiers. The key feature of XSPELLS is the adoption of variational autoencoders for generating meaningful synthetic texts in a latent space. We considered both a pre-trained variational autoencoder (OVAE) and two ones trained on a subset of the available data (SVAE and BVAE). The latent space is also revealed to be essential for inducing a decision tree which helps in characterizing exemplar and counter-factual exemplar texts. Diversity of counter-exemplars is modeled as an optimization problem. The approach advances over baseline explainers, such as LIME, which only highlight the contribution of words already in the text to explain. Experiments showed that XSPELLS also exhibits better fidelity and usefulness, while trading off stability with diversity.

The proposed approach has some clear limitations. *First*, we will consider extending the explanations returned by XSPELLS with logic rules, which convey information at a more abstract level than exemplars. Such rules can be extracted from the decision tree on the latent space, but have to be decoded back to rules on texts – a challenging task. *Second*, while the framework is general enough to cover multi-class black boxes, where the class labels are more than two, specific experimental analysis is required to properly validate the approach. For example, by comparing the usage of multi-class decision trees against transformation to binary problems (one vs. one, or one vs. rest). Extension of the framework are instead required to tackle multi-label classification, where more than one label might be assigned to an instance, and ranking of class labels (Sebastiani 2002). *Third*, XSPELLS could be extended to account for long texts. While the theoretical framework does not change, the implementation of VAEs does not scale to large dimensionalities of the input/output. A possible direction is to use word2vec embeddings (Goldberg and Levy 2014). *Fourth*, we could rely on semantic and linguistic resources (Altinel and Ganiz 2018), such a thesaurus or domain ontologies, to empower both synthetic text generation and to enrich the expressiveness of the (counter-)exemplars. *Finally*, a user study to assess the perceived usefulness of the explanations of XSPELLS would be definitively required, e.g., through crowdsourcing or lab experiments (Förster et al. 2020) or by comparing with human-annotated explanations (Wiegreffe and Marasović 2021). In fact, automatic measures have been shown to

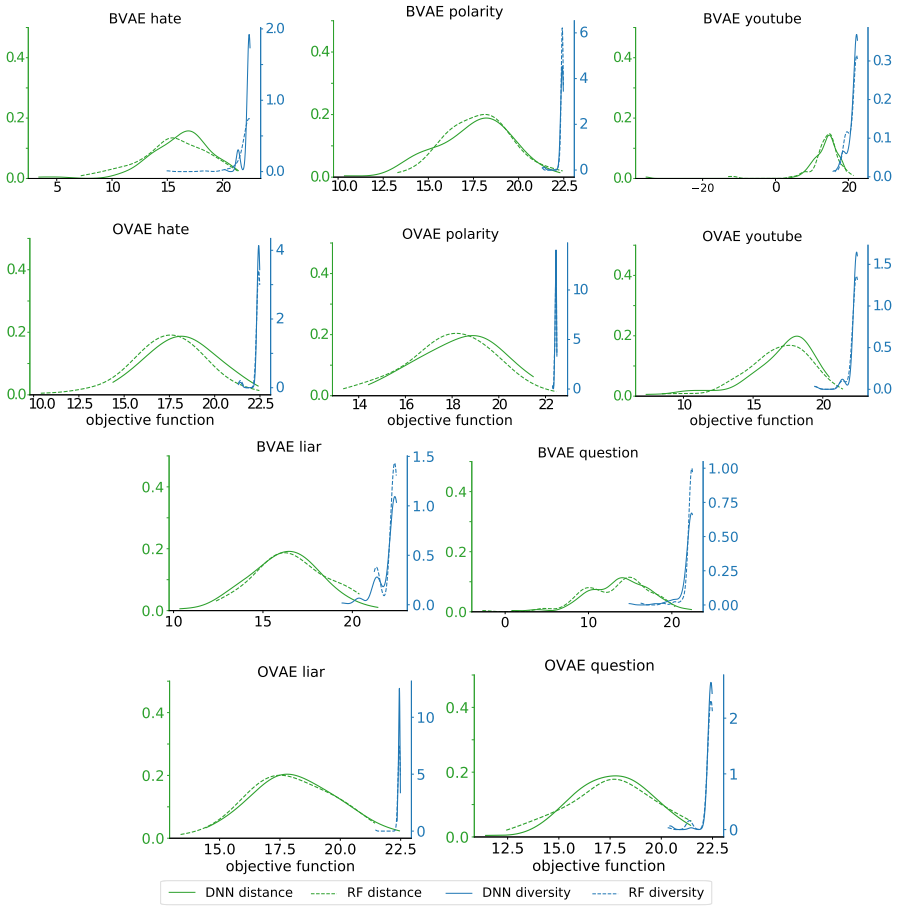

Fig. 6 Density estimation of the objective function $h_z(S)$ (see (2)) for S selected by the distance-based and by the diversity-based counter-exemplar strategies. Best viewed in color

correlate moderately with human judgements (Nguyen 2018). Even more challenging is to incorporate in the design of the explanations, cognitive and social aspects regarding the users of XSPELLS, i.e., moving towards Human-centered Explainable (AI Ehsan and Riedl 2020).

Appendix: Solving diversity through Sub-modularity

Consider a function $h()$ over sets, and let $h(c|S) = h(S \cup \{c\}) - h(S)$ be its discrete derivative. The function is sub-modular if, for $A \subseteq B$ and $c \notin B$, we have $h(c|A) \geq h(c|B)$ – a condition known as diminishing return. The objective function $h_z(S)$ in (2) is the difference

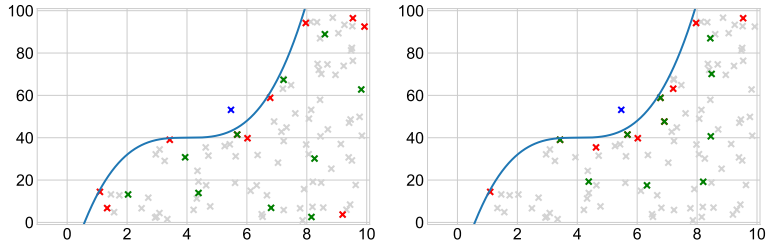

Fig. 7 Simulated example: z (blue), admissible instances in grey, results by **CSG** using $h_z()$ in green, results by **SG** using $q_z()$ in red. Left plot parameters for $h_z()$: $k = 10$ and λ as in (3); and for $q_z()$: $\lambda = 1$. Right plot parameters for $h_z()$: $k = 5$ and λ as in (3); and for $q_z()$: $\lambda = 1.5$. Best viewed in color

$f(S) - c(S)$ between a monotone non-negative sub-modular function (size of nearest neighbours) and a modular function (distance from z):

$$f(S) = |\bigcup_{a \in S} nn_k(a)| \qquad c(S) = \lambda \sum_{a \in S} dist(a, z)$$

The function $h_z(S) = f(S) - c(S)$ is then sub-modular, but it is not necessarily monotone nor non-negative. Hence the Standard Greedy (**SG**) algorithm (shown as as Algorithm 1) for monotone non-negative sub-modular optimization Nemhauser et al. (1978) does not provide any formal guarantee of approximating the optimal solution in the constrained problem (1). The maximization of such sub-modular functions has been considered recently Ene et al. (2020); Harshaw et al. (2019). Proposed algorithms return a solution $S^\star$ that approximates the best solution S^{opt} with the following guarantee:

$$f(S^\star) - c(S^\star) \geq \alpha f(S^{opt}) - c(S^{opt}) \tag{4}$$

for some $\alpha < 1$. Authors of Harshaw et al. (2019) achieve $\alpha = (1 - 1/e) \approx 0.63$, which is the best possible guarantee. Authors in Ene et al. (2020) consider the application setting where elements in S are experts to be grouped in a team, $f(S)$ is the size of the set of skills of experts in S, and the cost function $c(S)$ is the sum of the costs of team members. They propose the Cost Scaled Greedy (**CSG**) algorithm (shown as as Algorithm 2), which consists of the standard greedy algorithm (possibly, the accelerated version of Minoux (1978)) applied to a scaled objective function $h'(S) = f(S) - 2c(S)$. In our case, the discrete derivative of the scaled function is:

$$h'_z(c|S) = |\bigcup_{a \in S \cup \{c\}} nn_k(a)| - 2\lambda dist(c, x) - |\bigcup_{a \in S} nn_k(a)|$$

CSG is a simple, efficient algorithm. On the negative side, the approximation achieved w.r.t. (4) is for $\alpha = 1/2$, which is slightly lower than the best theoretical guarantee.

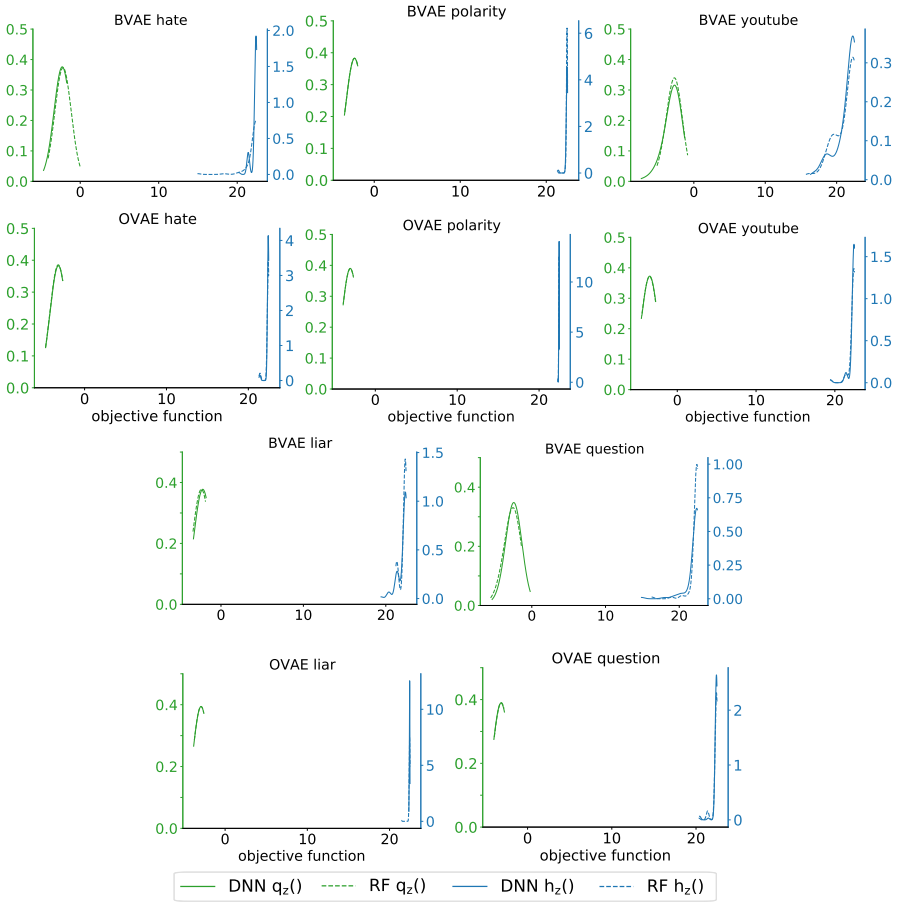

Fig. 8 Density estimation of the objective function $h_z(S)$ (resp., $q_z(S)$ for S selected by **CSG** (resp., **SG**). $\lambda = 1.5$ for $q_z()$. Best viewed in color

Algorithm 1 SG($\mathcal{A}$, v)

$S \leftarrow \emptyset$
for $i = 1$ **to** v **do**
  $a \leftarrow argmax_{c \in \mathcal{A}}\ h(c|S)$
  $S \leftarrow S \cup \{a\}$
end for
return S

Algorithm 2 CSG($\mathcal{A}$, v)

```
S ← ∅
for i = 1 to v do
    a ← argmax_{c∈A} h'(c|S)
    if h'(a|S) ≤ 0 then
        break
    end if
    S ← S ∪ {a}
end for
return S
```

Figure 7 presents the results of a simulation obtained by randomly generating 100 counter-exemplars w.r.t. the instance z (in blue) and the shown decision boundary. The green points are $v = 10$ counter-exemplars chosen by **CSG** using (the scaled derivative of) $h_z()$. The red points are $v = 10$ counter-exemplars chosen by **SG** using another intuitive objective function:

$$q_z(S) = \frac{1}{|S|^2} \sum_{a,b \in S} dist(a, b) - \frac{\lambda}{|S|} \sum_{a \in S} dist(a, z) \tag{5}$$

The idea here is to maximize the mean distance between selected counter-exemplars while minimizing their mean distance w.r.t. z. This function is not monotone nor non-negative. Moreover, $q_z(S)$ cannot be stated as $f(S) - g(S)$ for $f()$ monotone and non-negative. Hence, no formal guarantee can be stated when using **SG** for maximizing it (and neither for **CSG**).

Figure 7 shows that the standard greedy algorithm using $q_z()$ leads to instances close to the decision boundary, and possibly close to each other. The instances selected by **CSG** using $h_z()$, instead, distribute quite uniformly. The two plots in the figure shows the impact of the λ (for $q_z()$) and k parameters (for $h_z()$). Larger λ's mean larger costs in the objective function, hence instances selected by $q_z()$ are closer to z. Similar effects follow for smaller k's in the definition of $h_z()$ when fixing λ as in (3).

Finally, we contrast in Fig. 8 the distributions [12] of the objective functions $h_z(S)$ and $q_z(S)$ over the $v = 5$ counter-exemplars selected from the set of admissible ones by **CSG** and **SG** respectively. The images of $h_z()$ and $q_z()$ are different, hence we cannot compare the ranges of the densities. However, we observe that the densities of $q_z()$ are less skewed and peaked than those of $h_z()$. This is expected, as for the $h_z()$ function the **CSG** algorithm provides a theoretical guarantee of approximating the optimal solution, whilst for $q_z()$ and the standard greedy algorithm this does not hold.

Acknowledgements Orestis Lampridis would like to thank Ioannis Mollas and Grigorios Tsoumakas for their support.

Author contributions *OL* Conceptualization, Methodology, Software, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing. *LS* Methodology, Software, Validation, Investigation, Resources, Data Curation, Writing - Review & Editing, Visualization. *RG* Conceptualization, Methodology, Resources, Writing - Original Draft, Writing - Review & Editing. *SR* Conceptualization, Methodology, Supervision, Writing - Original Draft, Writing - Review & Editing, Project administration, Funding acquisition.

[12] Notice that the densities of $h_z(S)$ in Fig. 8 are precisely the ones in Fig. 6.

Funding Open access funding provided by Università di Pisa within the CRUI-CARE Agreement. This work has received funding from the European Union's Horizon 2020 research and innovation programme under Marie Sklodowska-Curie Actions (grant agreement number 860630) for the project "NoBIAS - Artificial Intelligence without Bias" (https://nobias-project.eu/ *nobias-project.eu*), and under the scheme "INFRAIA-01-2018-2019 – Integrating Activities for Advanced Communities" (grant agreement number 871042) for the project "SoBigData++: European Integrated Infrastructure for Social Mining and Big Data Analytics" (http://www.sobigdata.eu *www.sobigdata.eu*). This work reflects only the authors' views and the European Research Executive Agency (REA) is not responsible for any use that may be made of the information it contains.

Data availability `hate`: https://github.com/t-davidson/hate-speech-and-offensive-language `polarity`: https://www.cs.cornell.edu/people/pabo/movie-review-data/ `youtube`: https://archive.ics.uci.edu/ml/datasets/YouTube+Spam+Collection `liar`: https://www.cs.ucsb.edu/~william/data/liar_dataset.zip `question`: https://cogcomp.seas.upenn.edu/Data/QA/QC/

Declarations

Conflict of interest Not applicable.

Code availability https://github.com/lstate/X-SPELLS-V2

Ethics approval Not applicable.

References

T. C. Alberto, J. V. Lochter, and T. A. Almeida. Tubespam: Comment spam filtering on youtube. In *IEEE International Conference on Machine Learning and Applications (ICMLA 2015)*, pp 138–143. IEEE, 2015.

T. Alhindi, S. Petridis, and S. Muresan. Where is your evidence: Improving fact-checking by justification modeling. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pp 85–90, Brussels, Belgium, 2018. Association for Computational Linguistics.

Altinel, B., & Ganiz, M. C. (2018). Semantic text classification: A survey of past and recent advances. *Information Processing and Management, 54*(6), 1129–1153.

D. Alvarez-Melis and T. S. Jaakkola. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems (NeurIPS 2018)*, pp 7786–7795, 2018.

Arras, L., Horn, F., Montavon, G., Müller, K.-R., & Samek, W. (2017). What is relevant in a text document?: An interpretable machine learning approach. *PLoS One, 12*(8), e0181142.

A. Artelt and B. Hammer. On the computation of counterfactual explanations – A survey. arXiv:1911.07749, 2019.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one, 10*(7), e0130140.

Bertsimas, D., & Dunn, J. (2017). Optimal classification trees. *Machine Learning, 106*(7), 1039–1082.

F. Bodria, F. Giannotti, R. Guidotti, F. Naretto, D. Pedreschi, and S. Rinzivillo. Benchmarking and survey of explanation methods for black box models. *CoRR*, abs/2102.13076, 2021.

T. Bolukbasi, K. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In *Advances in Neural Information Processing Systems (NIPS 2016)*, pp 4349–4357, 2016.

S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Józefowicz, and S. Bengio. Generating sentences from a continuous space. In Y. Goldberg and S. Riezler, editors, *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, 11-12, 2016*, pp 10–21, 2016.

S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Józefowicz, and S. Bengio. Generating sentences from a continuous space. In *Conference on Computational Natural Language Learning (CoNLL 2016)*, pp 10–21. ACL, 2016.

R. M. J. Byrne. Counterfactuals in explainable artificial intelligence (XAI): evidence from human reasoning. In *Joint Conference on Artificial Intelligence (IJCAI 2019)*, pp 6276–6282. ijcai.org, 2019.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research, 16*, 321–357.

J. Chen, L. Song, M. J. Wainwright, and M. I. Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning, (ICML 2018)*, 80, pp 882–891. PMLR, 2018.

J. Clos and N. Wiratunga. Lexicon induction for interpretable text classification. In J. Kamps, G. Tsakonas, Y. Manolopoulos, L. S. Iliadis, and I. Karydis, editors, *International Conference on Theory and Practice of Digital Libraries (TPDL 2017)*, 10450 of *Lecture Notes in Computer Science*, pp 498–510. Springer, 2017.

D. Croce, D. Rossini, and R. Basili. Auditing deep learning processes through kernel-based explanatory models. In *International Joint Conference on Natural Language Processing (AACL/IJCNLP 2019)*, pp 4035–4044. ACL, 2019.

da Silva, N. F. F., Hruschka, E. R., & Jr, E. R. H. (2014). Tweet sentiment analysis with classifier ensembles. *Decision support systems, 66*, 170–179.

F. Dalvi, N. Durrani, H. Sajjad, Y. Belinkov, A. Bau, and J. R. Glass. What is one grain of sand in the desert? Analyzing individual neurons in deep NLP models. In *AAAI Conference on Artificial Intelligence (AAAI 2019)*, pp 6309–6317. AAAI Press, 2019.

M. Danilevsky, K. Qian, R. Aharonov, Y. Katsis, B. Kawas, and P. Sen. A survey of the state of explainable AI for Natural Language Processing. In K. Wong, K. Knight, and H. Wu, editors, *International Joint Conference on Natural Language Processing (AACL/IJCNLP 2020)*, pp 447–459. ACL, 2020.

D. Danks. The value of trustworthy AI. In *AAAI/ACM Conference on AI, Ethics, and Society (AIES 2019)*, pp 521–522. ACM, 2019.

T. Davidson, D. Warmsley, M. W. Macy, and I. Weber. Automated hate speech detection and the problem of offensive language. In *International Conference on Web and Social Media (ICWSM 2017)*, pp 512–515. AAAI Press, 2017.

J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (NAACL-HLT 2019)*, pp 4171–4186. ACL, 2019.

F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning. arXiv:1702.08608, 2017.

U. Ehsan and M. O. Riedl. Human-centered explainable AI: towards a reflective sociotechnical approach. In C. Stephanidis, M. Kurosu, H. Degen, and L. Reinerman-Jones, editors, *HCI International Conference (HCII 2020)*, 12424 of *Lecture Notes in Computer Science*, pp 449–466. Springer, 2020.

A. Ene, S. M. Nikolakaki, and E. Terzi. Team formation: Striking a balance between coverage and cost. *CoRR*, abs/2002.07782, 2020.

M. Förster, M. Klier, K. Kluge, and I. Sigler. Evaluating explainable artifical intelligence - what users really appreciate. In *European Conference on Information Systems (ECIS 2020)*, 2020.

Freitas, A. A. (2013). Comprehensible classification models: A position paper. *SIGKDD Explorations, 15*(1), 1–10.

Y. Goldberg and O. Levy. word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. *CoRR*, abs/1402.3722, 2014.

Guidotti, R., Monreale, A., Giannotti, F., Pedreschi, D., Ruggieri, S., & Turini, F. (2019). Factual and counterfactual explanations for black box decision making. *IEEE Intelligent Systems, 34*(6), 14–23.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2019). A survey of methods for explaining black box models. *ACM Computing Surveys, 51*(5), 1–42.

R. Guidotti and S. Ruggieri. On the stability of interpretable models. In *International Joint Conference on Neural Networks (IJCNN 2019)*, pp 1–8. IEEE, 2019.

C. Harshaw, M. Feldman, J. Ward, and A. Karbasi. Submodular maximization beyond non-negativity: Guarantees, fast algorithms, and applications. In *International Conference on Machine Learning (ICML 2019)*, 97, pp 2634–2643. PMLR, 2019.

Hemmatian, F., & Sohrabi, M. K. (2019). A survey on classification techniques for opinion mining and sentiment analysis. *Artificial Intelligence Review, 52*(3), 1495–1545.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science, 313*(5786), 504–507.
Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.
B. Hoover, H. Strobelt, and S. Gehrmann. exbert: A visual analysis tool to explore learned representations in transformers models. arXiv:1910.05276, 2019.
B. Kim, O. Koyejo, and R. Khanna. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in Neural Information Processing Systems (NIPS 2016)*, pp 2280–2288, 2016.
D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR 2014)*, 2014.
Korde, V., & Mahender, C. N. (2012). Text classification and classifiers: A survey. *International Journal of Artificial Intelligence & Applications, 3*(2), 85.
Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L. E., & Brown, D. E. (2019). Text classification algorithms: A survey. *Information, 10*(4), 150.
O. Lampridis, R. Guidotti, and S. Ruggieri. Explaining sentiment classification with synthetic exemplars and counter-exemplars. In *Discovery Science (DS 2020)*, 12323 of *Lecture Notes in Computer Science*, pp 357–373. Springer, 2020.
C. Li, X. Gao, Y. Li, B. Peng, X. Li, Y. Zhang, and J. Gao. Optimus: Organizing sentences via pre-trained modeling of a latent space. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pp 4678–4699. ACL, 2020.
J. Li, W. Monroe, and D. Jurafsky. Understanding neural networks through representation erasure. arXiv:1612.08220, 2016.
Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P. S. Yu, and L. He. A survey on text classification: From shallow to deep learning. *CoRR*, abs/2008.00364, 2020.
X. Li and D. Roth. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.
Linardatos, P., Papastefanopoulos, V., & Kotsiantis, S. (2021). Explainable AI: A review of machine learning interpretability methods. *Entropy, 23*(1), 18.
B. Liu and L. Zhang. A survey of opinion mining and sentiment analysis. In *Mining Text Data*, pp 415–463. Springer, 2012.
S. M. Lundberg and S. Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NIPS 2017)*, pp 4765–4774, 2017.
Malgieri, G., & Comandé, G. (2017). Why a right to legibility of automated decision-making exists in the GDPR. *International Data Privacy Law, 7*(4), 243–265.
Miller, T. (2019). Explanation in Artificial Intelligence: Insights from the social sciences. *Artificial Intelligence, 267*, 1–38.
Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning-based text classification: A comprehensive review. *ACM Computing Surveys, 54*(3), 1–40.
M. Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques*, pp 234–243. Springer, 1978.
I. Mollas, N. Bassiliades, and G. Tsoumakas. Lionets: Local interpretation of neural networks through penultimate layer decoding. In *Machine Learning and Knowledge Discovery in Databases – Workshops (ECML-PKDD 2019)*, pp 265–276. Springer, 2019.
R. K. Mothilal, A. Sharma, and C. Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Conference on Fairness, Accountability, and Transparency (FAT* 2020)*, pp 607–617. ACM, 2020.
Nemhauser, G. L., Wolsey, L. A., & Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions - I. *Mathematical Programming, 14*(1), 265–294.
D. Nguyen. Comparing automatic and human evaluation of local explanations for text classification. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (NAACL-HLT 2018)*, pp 1069–1078. ACL, 2018.
Ntoutsi, E., et al. (2020). Bias in data-driven Artificial Intelligence systems - An introductory survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 10*(3), e1356.
Olteanu, A., Castillo, C., Diaz, F., & Kiciman, E. (2019). Social data: Biases, methodological pitfalls, and ethical boundaries. *Frontiers Big Data, 2*, 13.
B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pp 115–124. ACL, 2005.
D. Pedreschi, F. Giannotti, R. Guidotti, A. Monreale, S. Ruggieri, and F. Turini. Meaningful explanations of black box AI decision systems. In *AAAI Conference on Artificial Intelligence (AAAI 2019)*, pp 9780–9784. AAAI Press, 2019.

K. Qian, M. Danilevsky, Y. Katsis, B. Kawas, E. Oduor, L. Popa, and Y. Li. XNLP: A living survey for XAI research in Natural Language Processing. In *International Conference on Intelligent User Interfaces (IUI 2021)*, pp 78–80. ACM, 2021.

M. T. Ribeiro, S. Singh, and C. Guestrin. "Why should I trust you?": Explaining the predictions of any classifier. In *ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD 2016)*, pp 1135–1144. ACM, 2016.

M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence (AAAI 2018)*, pages 1527–1535. AAAI Press, 2018.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence, 1*(5), 206–215.

S. Ruggieri. Subtree replacement in decision tree simplification. In *International Conference on Data Mining (SDM 2012)*, pp 379–390. SIAM, 2012.

Ruggieri, S. (2019). Complete search for feature selection in decision trees. *J. Mach. Learn. Res., 20*, 104:1-104:34.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys, 34*(1), 1–47.

Selbst, A. D., & Powles, J. (2017). Meaningful information and the right to explanation. *International Data Privacy Law, 7*(4), 233–242.

S. M. Shankaranarayana and D. Runje. ALIME: Autoencoder based approach for local interpretability. In *Intelligent Data Engineering and Automated Learning (IDEAL)*, 11871 of *Lecture Notes in Computer Science*, pp 454–463. Springer, 2019.

A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning (ICML 2017)*, pp 3145–3153. PMLR, 2017.

Skrlj, B., Martinc, M., Kralj, J., Lavrac, N., & Pollak, S. (2021). tax2vec: Constructing interpretable features from taxonomies for short text classification. *Computer Speech & Language, 65*, 101104.

Song, G., Ye, Y., Du, X., Huang, X., & Bie, S. (2014). Short text classification: A survey. *Journal of Multimedia, 9*(5), 635–643.

M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning (ICML 2017)*, pp 3319–3328. PMLR, 2017.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, pp 3104–3112, 2014.

P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Pearson Education India, 2016.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS 2017)*, pp 5998–6008, 2017.

S. Verma, J. P. Dickerson, and K. Hines. Counterfactual explanations for machine learning: A review. *CoRR*, abs/2010.10596, 2020.

G. Visani, E. Bagli, F. Chesani, A. Poluzzi, and D. Capuzzo. Statistical stability indices for LIME: Obtaining reliable explanations for machine learning models. *Journal of the Operational Research Society*, page to appear, 2021.

W. Y. Wang. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *ACL (2)*, pp 422–426. Association for Computational Linguistics, 2017.

S. Wiegreffe and A. Marasović. Teach me to explain: A review of datasets for explainable NLP. arXiv:2102.12060 0, 2021.

Xu, B., Guo, X., Ye, Y., & Cheng, J. (2012). An improved random forest classifier for text categorization. *J. Comput., 7*(12), 2913–2920.

Zafar, M. R., & Khan, N. (2021). Deterministic local interpretable model-agnostic explanations for stable explainability. *Machine Learning and Knowledge Extraction, 3*(3), 525–541.

X. Zhang, J. J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems (NIPS 2015)*, pp 649–657, 2015.

Zhou, X., Gururajan, R., Li, Y., Venkataraman, R., Tao, X., Bargshady, G., Barua, P. D., & Kondalsamy-Chennakesavan, S. (2020). A survey on text classification and its applications. *Web Intelligence, 18*(3), 205–216.

Z. Zhou, G. Hooker, and F. Wang. S-LIME: Stabilized-LIME for model explanation. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2021)*, pp 2429–2438. ACM, 2021.