The 35th Canadian Conference on Artificial Intelligence
DOI: 0

Canadian AI 2022
30 May–3 June 2022, Toronto

A Modularized Framework for Explaining Black Box Classifiers for Text Data

Mahtab Sarvmaili[†,*], Riccardo Guidotti[◇], Anna Monreale[◇], Amilcar Soares[‡], Zahra Sadeghi[†]
Fosca Giannotti[◇], Dino Pedreschi[◇] and Stan Matwin[†,§]
[†] Dalhousie University, Halifax, Canada
[‡] Memorial University of Newfoundland, St. John's, Canada
[◇] ISTI-CNR, Pisa, Italy
[§] Institute of Computer Sciences, Polish Academy of Sciences, Warsaw, Poland

Abstract

The cumbersome amount of textual data produced in social media and in the new digital life makes the usage of automatic decision systems necessary for acting on text. The most widely adopted natural language processing approaches guarantee high accuracy but are black-box systems, that hide the logic of their internal decision processes. Since in various applications there is the need to unveil the reasons for the classification of different texts, the urge to explain black-box behaviour is growing among scientists. Thus, we propose a local model-agnostic method for interpreting text classifiers. Our method explains the decision of a text classifier on a given document by generating similar samples in its vicinity. The new samples are generated by replacing words of the document under analysis with their synonyms, antonyms, hyponyms, hypernyms, and definitions. Finally, these synthetic texts are used to train a decision tree that enables the user to identify important words explaining the classification outcome. An inspection of the synthetic documents generated by our proposal together with a set of words appropriately highlighted explain why the black box assigns a certain label to a given document. Deep and wide experimentation on various datasets and classifiers shows the effectiveness of our proposal and that its performance overcomes state-of-the-art methods.

Keywords: Explainable AI, Text Classifiers Explanation, Sentence Highlighting, Interpretable Machine Learning

1. Introduction

Textual data is one of the most widely widespread data type. Indeed, as humans, we use natural language translated into text to communicate, store information, express opinions etc. The novel interconnected society produces ton of terabyte of textual data every day. This enormous amount of information needs to be managed through automatic decision systems. Automatic text classification has been widely adopted for sentiment classification, fake news detection, spam alert, etc. To handle these non-trivial goals, many complex machine learning systems have been designed, ranging from language models [1–3], to machine translation [4], and text generation [3].These systems have achieved outstanding performance thanks to deep learning approaches designed for Natural Language Processing (NLP). The weakness of these approaches is that they are indeed a mystery for the users due to to their difficult comprehensive internal structure, as well as to their sheer size, they are often referred to as *"black box"* models [5]. Besides, the widespread adoption of machine learning algorithms has increased the necessity of *trust* these models in order to employ them for critical decision-making scenarios [6]. Despite the considerable interest in explainable methods in visual and tabular domain, a limited research has been conducted in textual field [7]. As a consequence, in this paper, we propose DICTA, a modularizeD model-agnostic framework for the explanatIon of black box Classifiers for Text dAta. A novelty aspect of DICTA with respect to the interpretability literature is that it draws on robust and generally used

[*] mahtab.sarvmaili@dal.ca

NLP tools to explain classifiers working on texts in natural language. Given a black box classifier, a text document under analysis, and the decision of the classifier on the document, DICTA returns an explanation of the classification in terms of words on the document under analysis highlighted depending on their importance, which is measured as responsibility for the classification outcome. In contrast to state-of-the-art textual explainers, DICTA exploits the notion of *influential sentences* to audit the black box classifier and to extract the explanation. With influential sentence we refer to a sentence that has a high impact on the classification outcome of a document. DICTA exploits such sentences to generate a set of similar documents by exploiting WordNet [8]. In particular, DICTA randomly replaces words of influential sentences by their *semantic replacement* which is the collection of synonyms, antonyms, hyponyms, hypernyms, and definitions. Thus, DICTA preserves the structure of the original text as it generates synthetic text samples through the WordNet ontology. DICTA explores the behavior of black box model by training a decision tree on a simplified representation of the synthetic texts. The tree is exploited for highlighting the words more responsible for the class label of a document. The main advantages of DICTA are (i) fast, simple and easily customizable neighborhood generation, (ii) simple and understandable factual and counterfactual explanations [9, 10], (iii) and modular design. We conducted experiments on four sentiment analysis benchmark datasets (IMDB, Amazon, Yelp, and U.S. Airline Tweets) and four text classifiers to assess different desiderata for model-agnostic explainers, namely explanation trustfulness, synthetic neighborhood compactness, plausibility, and sentimental agreement. Moreover, we developed an evaluation method for comparing the degree of agreement between the salient scores sentences extracted by the explanation algorithms and their sentimental polarity obtained from a sentiment lexicon. The goal was to facilitate *(i)* comparing DICTA with other explanation methods that do not locally explore the behavior of black box model, and *(ii)* measuring how much the estimated salient scores are similar to the actual sentiment of words. The rest of the paper is organized as follows. Section 2 discusses related works. Section 3 illustrates the proposed method. Section 4 presents the experimental results. Finally, Section 5 concludes the paper discussing known limitations.

2. Related works

The extensive application of deep and complex machine learning models on different domains [11] increases the need to understand the decisions taken by the systems and adopting these models in order to employ them for critical decision-making. This resulted in a growing research on the explainability methods for complex and obscure models. From a top-level perspective, they are categorized as *model-specific* versus *model-agnostic*. The two most well-known local model-agnostic explainers that can also be applied to text are LIME [12] and SHAP [13]. LIME randomly generates synthetic neighborhood texts and trains a linear model on such instances as an interpretable local surrogate.The problem of using LIME with text is that the text neighborhood are generated by randomly removing words, possibly generating meaningless sentences [14]. SHAP is another model agnostic approach that takes advantage of the Shapely value estimation [15] and tests different combinations of words to understand their importance in the decision outcome which randomly removes words from the text under analysis. Therefore, in both cases, the black box is audited with texts that can be implausible, meaningless, or adversarial as they could be potential outliers with respect to the original training set of the machine learning model. DICTA departs from these weaknesses by exploiting an ad-hoc neighborhood generation guaranteeing the creation of realistic sentences by design. X-SPELLS [**lampridis2020explaining**] overcomes the limitation of LIME and SHAP through the usage of a Variational Auto-Encoder for the neighborhood generation of texts that similar to [9], is moved into a latent space. This

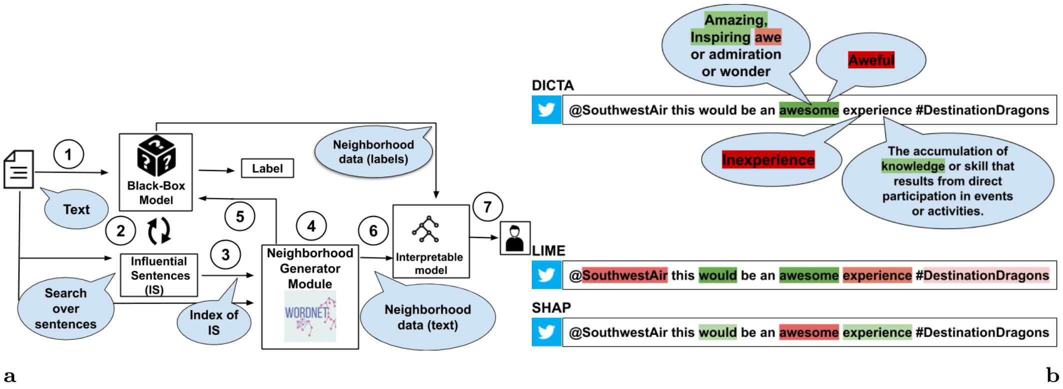

1
Text
Black-Box Model
Label
Neighborhood data (labels)
2
5
4
6
7
Influential Sentences (IS)
3
Search over sentences
Index of IS
Neighborhood Generator Module
WORDNET
Interpretable model
Neighborhood data (text)
DICTA
Amazing, Inspiring awe or admiration or wonder
Aweful
@SouthwestAir this would be an awesome experience #DestinationDragons
Inexperience
The accumulation of knowledge or skill that results from direct participation in events or activities.
LIME
@SouthwestAir this would be an awesome experience #DestinationDragons
SHAP
@SouthwestAir this would be an awesome experience #DestinationDragons
a
b

Figure 1. (a) Overview of DICTA framework. DICTA takes as input a textual document (step 1), classifies it with a black box and extracts the most influential sentences impacting more on the probability label (step 2). It matches words in the influential sentences with a possible semantic replacement using an ontology (e.g., WordNet) and generates a synthetic dataset (steps 3–5). Finally, it trains a local decision tree on the synthetic neighborhood, exploits the tree to retrieve the importance of the words used for classification, and returns them to the final user (steps 6–7). (b) Explanations of a document labeled as "Positive" by a Bidirectional-GRU on Airlines tweets dataset with DICTA, LIME and SHAP. Positive and negative impacts are highlighted with green and red shades respectively.

strategy seems more easily usable on continuous visual domains; however, for text data, the small perturbation in the feature space can drastically change the meaning of a given example and result in inaccurate examples [16]. Additionally modern text pocessing techniques require billions of hyperparameters, e.g., BERT [1], therefore using another complex language model to explain the behavior of a black box can bring an unnecessary additional source of complexity for the user. Moreover, DICTA does not assume any constraints on the length of documents or their problem domain as opposed to [17, 18]. LioNets [19] and exBERT [20] are model-specific explainers specifically designed for text classifiers that try to locally estimate the behavior of a deep neural network by generating a neighborhood at the penultimate layer of the network. These approaches rely on the choice of a reference example that strongly depends on the problem's domain. DICTA overcomes the limitation of these approaches by being a model-agnostic explainer and not requiring a reference sample but only an ontology.

3. DICTA

This section presents DICTA, a modularizeD model-agnostic framework for the explanatIon of black box Classifiers for Text dAta. Let $d = \langle S_1, \ldots, S_n \rangle$ be a document represented as a sequence of sentences $S_i = \langle w_1, \ldots, w_m \rangle$, with $1 \leq i \leq n$ and where any w_j with $1 \leq j \leq m$) is a word. Explaining the decision of a black box model f on a given document d, i.e., $f(d) = y$, means presenting an explanation e, that belongs to a human-understandable domain E. The proposed explanation method is the next step in the line of research on local model-agnostic methods originated from [9, 12, 21]. Thus, the idea of DICTA is to unveil the reason for classification of a trained text classifier by studying its behavior on the synthetic neighborhood of a given document. In other words, DICTA locally estimates, for every classified document d, the decision boundary of a complex decision function f. More specifically, the explanation e produced by DICTA locally approximates the decision boundary of f around d by highlighting the words more responsible for the decision $f(d) = y$ by exploiting the idea of *semantic replacement*. Thus, d syntax remains relatively unchanged

Algorithm 1: DICTA(d, f, R, T)

```
Input  : $d$ - document to explain, $f$ - black box function, $k$ - nbr. influential
         sentences, $T$ - words ontology, $n$ - neighborhood size
Output: $e$ - explanation
1  $y_p \leftarrow f_p(d)$;                                  // get probability of prediction
2  $A \leftarrow \emptyset$;                                 // init. infl. sent. scores
3  for $S_i \in d$ do
4      $d' \leftarrow remove(S_i, d)$;                       // remove sentence
5      $A_i \leftarrow |y_p - f_p(d')|$;                     // store cand score.
6  $I \leftarrow select(\mathcal{S}_c, k)$;                  // get indexes top k sentences
7  $\mathcal{S} \leftarrow \{S_i | i \in I\}$;               // select top k sentences
8  $\mathcal{R} \leftarrow \emptyset$;                       // init. semantic replacement
9  for $S_i \in \mathcal{S}$ do
10     for $w_j \in S_i$ do
11         $\mathcal{R}_{w_j} \leftarrow repl(w_j, T)$;      // semantic replacement
12 $N \leftarrow \emptyset$;                                 // init. neighborhood
13 for $i \in [1, n]$ do
14     $d' \leftarrow copy(d)$;                              // copy the document
15     $S_i \leftarrow rndSelection(\mathcal{S})$;           // select sentence
16     $W \leftarrow rndSelection(S_i)$;                     // select words
17     for $w_j \in W$ do
18         $d' \leftarrow repalce(w_j, \mathcal{R}_{w_j}, S_i, d')$;   // replace word j in sentence i
19     $N \leftarrow N \cup \{d'\}$;                         // add to neighborhood
20 $Y \leftarrow f(N)$;                                      // classify neighborhood
21 $dt \leftarrow train(N, Y)$;                              // train decision tree
22 $e \leftarrow extractExpl(dt, f(d))$;                     // get explanation
23 return $e$;
```

while its semantic is modified. The main idea of DICTA (shown in Figure 1(a) and explained in Algorithm 1) is to study how the semantic replacement of specific words affects the classification. DICTA does not operate on all the words of a document but only on the words belonging to *influential sentences*. Influential sentences are those with the highest impact on document classification. The three main steps of DICTA for explaining the behavior of black box model are: *(i)* identification of influential sentences, *(ii)* neighborhood generation through semantic replacement, *(iii)* local interpretable surrogate training and explanation extraction.

3.1. Influential Sentences Extraction

A key component of DICTA is the identification of influential sentences with high impact on the class label of the document. Thus, influential sentences contain the more discriminative words that are essential for distinguishing the document's class label. DICTA identifies influential sentences as follows (lines 1–7 in Alg. 1, steps 1–2 in Fig. 1(a)). First of all, DICTA queries the black box f and stores the *probability* for obtaining the label $y = f(d)$ for the document under analysis d ($y_p = f_p(d)$ in Alg. 1 (line 1)). Then, for each sentence $S_i \in d$ (lines 3–5), DICTA creates a synthetic document d' as a copy of d but without the sentence S_i (line4), and it stores in the influential sentences score candidate set A the absolute deviation between y_p and $f(d')$. Finally, it identifies the indexes of the k sentences with the most significant influence and stores them in the set $\mathcal{S}$. k is one of the data-dependent hyper-parameters depending on the average/maximum number of sentences in the dataset. After this step, the influential sentences $\mathcal{S}$ and the document d are passed to the neighborhood generator process (step 3 in Fig. 1(a)).

3.2. Neighborhood Generator

The neighborhood generator process is responsible for creating synthetic documents Z similar to d on which the black box function has to be queried to understand the reasons for the label $y = f(d)$. It starts with the identification of the set of words $\mathcal{R}$ to be used for the semantic replacement (lines 8–11 in Alg. 1, steps 4–5 in Fig. 1(a)). Thus, for each influential sentence in $S_i \in \mathcal{S}$ and for each word $w_j \in S_i$, DICTA identifies the set of words to be used as a semantic replacement of w_j as $\mathcal{R}_{w_j}$ with respect to a given ontology T such that, in case of substitution with w_j, the meaning of the sentence remains the same (line 11). Given the number n of neighbors to generate, DICTA creates a copy d' of the document under analysis d (line 14). It randomly selects an influential sentence S_i (line 15) and from S_i, it randomly chooses a set of words W (line 16). Then, it replaces the selected words W with random words from their semantic replacements $\mathcal{R}_{w_j}$ (lines 17–18).Finally, the synthetic document created by DICTA through this procedure is stored in the neighborhood N. After that, DICTA audits the black box function through the synthetic neighborhood to do the classification $Y = f(N)$ (line 20). In our implementation, as ontology T, we adopted *WordNet* [8], a robust lexical database to find the semantic replacements of the words. WordNet's use is essential, as it preserves the distribution of the given document features, and the synthetic sentences are semantically similar to the original one, according to the distributed semantic principle that states that "a word is characterized by the company it keeps" [22]. The use of WordNet has two advantages: *(i) transparency*: the relations between the lexical categories are intuitive and understandable to all users regardless of their linguistic knowledge; *(ii) accessibility*: WordNet is freely available in more than 200 languages and has connectors to many programming languages/systems without any fine-tuning. WordNet used as the primary resource for neighborhood generation, makes the design of our method more attractive for its usage in limited computing conditions, while sophisticated complex language models like BERT impose a heavy computational overhead on the interpretability task.

3.3. Local Decision Tree & Explanation

After the neighborhood generation, the neighborhood N and the corresponding labels Y are employed to train an interpretable local surrogate model (line 22 in Alg. 1, step 6 in Fig. 1(a)) by training an interpretable decision tree dt (line 22) on N and Y. We adopt a decision tree to explain the black box's local behavior due to its simplicity and comprehensibility for non-expert users [23]. The decision tree is finally used for extracting the most important words responsible for the classification that composes the output explanation e (lines 23–24 in Alg. 1, step 7 in Fig. 1(a)). Thus, given the document d, DICTA extracts the words' importance by tracing the conditions triggered by d on the path from the root node to leaves on dt. The importance of words is obtained as the normalized total reduction of the Gini criterion in the decision tree dt brought by each word *Gini importance* [24]. The quantification of the importance of the words enables the building of a sort of "saliency map" that highlights the important word w_j in d with their corresponding score. Besides the words' importance, we chose the decision tree as an interpretable surrogate model because its graphical representation allows a user to visually and comprehensively trace the decision of a black box. Also, the words in the tree appear structured in a top-down format that shows close to the root the most important ones [25]. An example of explanation e

Table 1. Datasets description (left), accuracy (right).

	#docs	**#categ**	**Avg #w**	**Max #w**	**Avg #S**	**Max #S**	**#infl. sent.**	**BiLSTM**	**BiGRU**	**CNN1d**	**RF**
Yelp	700k	5	9	438	8	150	5	0.60	0.61	0.56	0.52
Amazon	278k	5	8	169	4	122	3	0.66	0.67	0.65	0.60
Airline Tweets	14k	3	7	20	2	9	2	0.78	0.77	0.76	0.70
IMDB	50k	2	13	384	10	117	6	0.86	0.86	0.85	0.79

returned by DICTA is presented in Figure 1(b). Here the relevant words for the positive class are colored in green, and the ones for the negative class are colored in red. The color intensity of the highlighted words measures the importance with respect to a specific class. The advantage of using semantic replacement is that the original structure of a document is preserved and it is easy to observe the importance of other words replaced with the highlighted ones among those in the synthetic neighborhood. On the other hand, the random elimination of words for neighborhood generation like the one performed by LIME or SHAP (Figure 1(b)) may result in too short documents that do not have the same structure and meaning with the original document. In addition, with a random elimination, a group of words may get frequently removed from the document, and their real effect may not be adequately recognized. DICTA overcomes this possibility by understanding the effect of each word by replacing it with words having the same meaning or the opposite one. Indeed, in cases in which a word is very influential for labeling a document, replacing it with a word with the opposite meaning will provide a potent effect on the probability of the class label. The comparison shows that DICTA focuses on the most important words and has a richer explanation because it attaches the set of words derived by the WordNet and highlights their impact on the text classification.

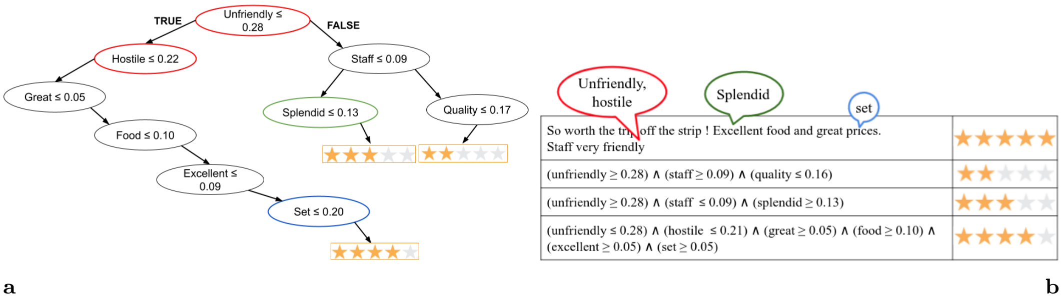
4. Experiments

In this section, we show the effectiveness of DICTA through a quantitative and qualitative evaluation[1]. First, we illustrate the experimental setting. After that, we show a quantitative evaluation on different metrics comparing DICTA with state-of-the-art local model-agnostic explainers on different datasets and text classifiers. Finally, we report a qualitative evaluation practically illustrating the benefits and the readability of the explanations returned by DICTA.

4.1. Experimental Setting

We experimented on four textual dataset typically used to train classifiers able to detect the sentiment of the documents: *(i) IMDB* movies reviews [26] that contains highly polarized opinions reviews on movies, *(ii) Yelp* [27] data that includes the businesses' reviews, *(iii) Amazon* [28] dataset of product reviews, and *(iv)* anonymous *Tweets* related to U.S airlines [29]. The number of instances, number of categories, the average/maximum number of words and sentences in the dataset are shown in Table 1 (left). We split the data into training *(80%)*, testing *(10%)*, and validation *(10%)*. We report in Table 1 also the number of informative sentences adopted by DICTA as it varies from dataset to dataset. We experimented with DICTA by explaining four text classifiers, i.e., CNN1D, BiGRU, and BiLSTM based on deep learning and a Random Forest (RF) classifier. The structure of text

[1] Python code and datasets at: https://github.com/MahtabSarvmaili/DICTA.git. Experiments were run on Ubuntu 20.04.1 LTS, Intel® Core™ i7 CPU, 16 GB DIMM DDR4 RAM.



So worth the trip off the strip ! Excellent food and great prices. Staff very friendly	★★★★★
(unfriendly ≥ 0.28) ∧ (staff ≥ 0.09) ∧ (quality ≤ 0.16)	★★☆☆☆
(unfriendly ≥ 0.28) ∧ (staff ≤ 0.09) ∧ (splendid ≥ 0.13)	★★★☆☆
(unfriendly ≤ 0.28) ∧ (hostile ≤ 0.21) ∧ (great ≥ 0.05) ∧ (food ≥ 0.10) ∧ (excellent ≥ 0.05) ∧ (set ≥ 0.05)	★★★★☆

Figure 2. (a) A local surrogate tree explaining CNN1D for a Yelp review. Due to the space limitation, some parts of the original tree are not shown. (b) Decision rules extracted from the tree. The first row of the table shows the document under analysis and its label (five-star review), and the callouts at the top of each word present words from the semantic set (green for synonyms, RED for antonyms, and blue for hyponyms and hypernyms) that was replaced with the word . The second to fourth row are the decision rules extracted from the decision tree and the final labels that are assigned to these rules. The antecedent of rules are in the left column, and their consequents are in the right column.

classifiers is as follows:*(i)* BiLSTM uses one embedding layer that is initiated with the pre-trained vectors, two layers of bidirectional LSTM, which is followed by a Dense layer and a softmax over the class labels. *(ii)* BiGRU has the same architecture, but instead of LSTM, it uses the Bidirectional GRU. CNN1D follows the architecture of [30]. We trained deep learning based models with the following parameter setting: batch size 200, word embedding dimension 200, the maximum number of unique tokens 200k, and 10 training epochs. Finally, we used RF with 150 trees. The accuracy of the various classifiers is reported in Table 1 (right). We measured the goodness of the explanations returned by DICTA in terms of the following indicators. First, *correctness* of the words importance scores with respect to the sentiment scores assigned by the sentiment lexicon VADER [31]. Second, the *fidelity* of the local surrogate model with respect to the black box classifier.Third, the *plausibility and similarity of the neighborhood* measured in terms of outliers present in the generated data and similarity between real data and synthetic neighborhoods.Details on every indicator are given in the corresponding section. We compared with the state-of-the-art explanation methods LIME [12] and SHAP [13]. In particular, due to the high degree of structural similarity between DICTA and LIME, we could perform comparisons among them for all the aforementioned measures.On the other hand, since SHAP does not train a local surrogate and does not generate a neighborhood, we limit the evaluation of DICTA and SHAP on the first measure. If not differently specified for DICTA, we adopted the following parameter setting as a result of a preliminary experimentation not reported here due to lack of space. The number of influential sentences selected for each dataset is reported in Table 1. Then, we generated neighborhoods composed by $n = 100$ synthetic documents.

4.2. **Salient Scores and Sentimental Polarity Agreement**

Inspired by [32], we report an experiment where we compare the salient scores of words with their sentimental polarity extracted from a sentiment lexicon resource. We designed this experiment to observe the impact of words on the prediction of black box model regardless of the class label. We binarized the class labels of the datasets by using the middle class as a threshold to assign the two labels. For example, in the Yelp dataset, we have 1-5 stars ratings for businesses. From 1-3 we assigned the class as a "negative" review and 4-5 as a positive review. The same procedure was used for Amazon, and the scales 1-2 as negative

Table 2. Agreement between the sentimental polarity extracted from explainers and VADER sentimental lexicon.

		Yelp	**Amazon**	**Airline Tweets**	**IMDB**
BiGRU	DICTA	0.57	0.65	0.59	0.48
	LIME	0.43	0.48	0.43	0.52
	SHAP	0.47	0.50	0.47	0.84
BiLSTM	DICTA	0.56	0.66	0.61	0.48
	LIME	0.46	0.47	0.41	0.53
	SHAP	0.47	0.46	0.49	0.85
CNN1D	DICTA	0.57	0.65	0.59	0.48
	LIME	0.46	0.43	0.37	0.52
	SHAP	0.47	0.51	0.41	0.84

and 3 as positive were used for U.S. Airline tweets. We have also binarized the importance scores of LIME and SHAP by taking the negative scores as a negative contribution towards a label value and the positive scores as a positive contribution towards a label value. Thus, for validation purposes, we extracted the sentiment scores of words from the sentiment lexicon VADER [31] and we binarized them to the *positive* and *negative* classes. On the other hand, for DICTA, we obtained the polarity of words at each node of the decision tree by examining the normalized ratio of the number of samples that fall into each class. If the class label is considered “positive”, we assigned the positive annotation to the words; otherwise, we label them as “negative”. Finally, we measured the percentage of agreement, i.e., the higher the better, between the classes provided by LIME, SHAP, and DICTA and the sentiment suggested by VADER. The results are shown in Table 2 clearly reveals that DICTA outperforms LIME and SHAP on three datasets out of four (Yelp, Amazon, and Airline Tweets). However, for IMDB dataset SHAP is the best performer, requiring a deeper investigation to understand the reasons for this result. We highlight that we gained a significant understanding of word scores when carefully exploring the word importance extracted from the decision tree. Indeed, the decision tree structure allows exploring better the connection between the input document and the lexicon composing it. In cases where the label assigned by the black box to a document is negative, the document usually contains strong negative words that commonly reside in the top nodes of the decision tree. When we trace down the tree from the root to the leaves to obtain the polarity of words, most of the neutral or positive words (that can flip the class of a document) reside in the intermediate levels or close to the leaves. Although these words reduce the number of negative instances at their level, most instances may belong to the negative class, causing the positive words assigned a negative sentiment. Hence, we highlight that traversing the tree empowers final users to understand the relation between the words and their impact on the sentiment label of a document.

4.3. **Qualitative Evaluation of Decision Rules**

We linearized the decision tree into an understandable rule form that provides a more flexible semantics for representing the classifier [33]. Rules can be extracted from the decision tree by tracing down a decision path from the root node to the leaf. An example of the trained decision tree and traversing it for an instance of Yelp reviews is given in Figure 2. Since we are using the TF-IDF vectors for training the decision tree, the real value that each feature is compared to translates to the importance of that word in the neighborhood text set. We start from the root node then we follow paths to get to the leaves. By tracing down the decision tree and following the ensuing rules, we look at the words in the order of their importance for the decision in the leaf of the tree. The salient words are served to the user in the order of their importance for the decision in the leaf of the tree (in consequent of

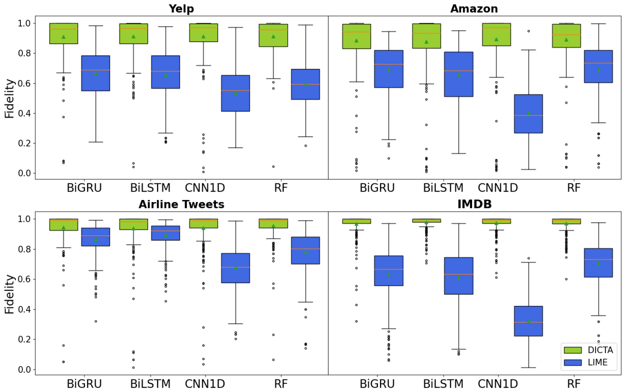


Figure 3. Fidelity (accuracy) of DICTA and LIME on datasets

Table 3. The *p−values* of Wilcoxon test showing the statistical significance of the improvement of DICTA's performance over LIME for the fidelity.

	IMDB	**Amazon**	**Yelp**	**Airline tweets**
BiLSTM	$5.33 \times e^{-29}$	$3.88 \times e^{-08}$	$1.15 \times e^{-28}$	$1.44 \times e^{-11}$
BiGRU	$1.91 \times e^{-31}$	$2.48 \times e^{-09}$	$3.13 \times e^{-29}$	$3.61 \times e^{-05}$
CNN1D	$1.88 \times e^{-41}$	$3.66 \times e^{-}28$	$2.91 \times e^{-28}$	$1.79 \times e^{-15}$
RF	$1.11 \times e^{-41}$	$4.23 \times e^{-}11$	$9.82 \times e^{-43}$	$6.56 \times e^{-20}$

a decision rule). As has been shown in the classical cognitive science literature investigating explainability of decision rules [34] understanding decision rules in this manner, which is suggestive of causality, is a better explanation of the rules (and therefore a better explanation of the model) than one consisting of an unordered set or salient words. Referring to Figure 2 (b) the user may find an explanation starting with unfriendly and hostile staff sufficient cause to understand why a restaurant is not recommended, rather than an explanation starting with the set or quality.

4.4. **Fidelity Evaluation**

We compared DICTA against LIME on text classification, measuring the *fidelity* [35, 36] of the interpretable surrogate model with respect to the black box model. The fidelity of an interpretable model indicates its faithfulness in imitating the behavior of the black box in the neighborhood of a particular data point.This is important because the meaningfulness of an explanation should be at least *locally faithful*. The *fidelity* can be measured as the accuracy of the prediction of the local surrogate model c on the neighborhood N_d generated for document d with respect to the prediction of the black box on the same set, i.e., we aim at comparing $y_c = c(N_d)$ with $y_f = f(N_d)$. As an evaluation measure, we rely on the accuracy between y_c and y_f. In particular, for DICTA c is the local decision tree, while for LIME, c is the local regressor. We calculated the fidelity by sampling uniformly at random 300 instances from the test data. We measured the accuracy of DICTA and LIME in mimicking the behavior of black box decisions for their neighborhood. We report the comparison of DICTA and LIME fidelity in the box-plots in Figure 3. The results show that DICTA outperforms LIME on all datasets and all text classifiers in imitating the black box behavior. We also evaluated the statistical significance of the improvement of DICTA's performance over LIME. To this end, we employed the Wilcoxon test to analyze the fidelity of these models. We report the p-values for all datasets and black boxes in Table 3. Very low p-values indicate that the differences between DICTA and LIME performance are significant; therefore, results in Table 3 prove a statistical evidence of DICTA's superior fidelity.

4.5. **Synthetic Neighborhoods Evaluation**

		Yelp		**Amazon**		**Airline Twitter**		**IMDB**	
		C	**L**	**C**	**L**	**C**	**L**	**C**	**L**
BiLSTM	DICTA	0.25	1.54	0.28	1.33	0.24	7.25×1e3	0.17	1.35
	LIME	0.27	1.57	0.24	1.31	0.30	1.0	0.21	1.96
BiGRU	DICTA	0.26	1.24	0.28	1.91	0.23	1.41	0.19	1.76
	LIME	0.29	1.25	0.27	1.62	0.29	1.0	0.24	3.41
CNN1D	DICTA	0.28	1.51	0.22	1.5	0.16	1.07	0.14	1.57
	LIME	0.33	1.29	0.25	2.37×1e4	0.28	1.13×1e7	0.20	1.40
RF	DICTA	0.34	1.20	0.45	2.45×1e7	0.38	1.32	0.23	1.68
	LIME	0.33	1.46	0.32	1.69	0.32	3.47	0.32	1.72

Table 4. The average cosine distance (**C**), and LOF (**L**) between the original document and neighborhood data of DICTA and LIME.

In this section we discuss the evaluation results on the quality of synthetic textual data generated as local neighborhoods. We quantitatively evaluated the density and cohesion [37] of neighborhood data to prove that DICTA produces diverse, plausible and high quality neighborhood text data in comparison to LIME. To this end, we adopted two approaches: *(i)* measuring the average *cosine distance* between the original document d and neighborhood N_d, that provides an evidence about the *similarity* of the neighborhood (cohesion), and *(ii)* measuring the Local Outlier Factor (LOF) [38], that captures the level of neighborhood density providing insights on its plausibility and diversity degree. The average cosine distance value (**C** columns) between the original document and the synthetic documents in the neighborhood shows the degree of similarity between them. The results in Table 4 show that the average cosine distance to DICTA's neighborhood is lower than LIME's neighborhood, especially for deep text classifiers. In LOF, which is an anomaly detection approach, the local density of the data is compared against its neighbors' local densities to identify similar density regions LOF. It uses the k-Nearest Neighborhood to recognize these regions. The points with a considerably lower density to their neighbors are considered outliers. We employed LOF to evaluate the neighborhood compactness and density with respect to a reference population given by the original dataset. Table 4 reports the average LOF (**L** columns) for DICTA and LIME on the four datasets. We observe that DICTA has similar or higher LOF values for most of the datasets compared to LIME, which means that the DICTA neighborhood generation, based on the WordNet, leads to exploring at least a similar or wider area of the neighborhood around the given document. If we consider both cosine distance and LOF (Table 4) values of the generated data, we can infer that although DICTA neighborhood has a higher diversity in terms of the number of different words, the generated text remains semantically similar to the original document.

5. **Conclusions**

We have presented DICTA, a model-agnostic explainer for black box text classifiers. DICTA explains a black box model's behavior by evaluating the impact of words and their semantic replacement on the class distribution of a document. In this way, each word's value is made explicit, and replacing it with its semantic replacement allows to check how it is possible to positively or negatively change the class label of a given document. Hence, the explanations provided by DICTA are more expressive and understandable than the ones provided by LIME and SHAP, which are based on features' importance. The evaluation of the synthetic neighborhoods indicates that DICTA preserves the essence of the original document and enriches it with semantically similar sentences. This feature of DICTA is essential for short documents because eliminating words like LIME or SHAP, rather than replacing it as we do, can drastically change the structure of the documents analyzed. Future work aims to study the neighborhood generation and its advantages in explaining modern text classifiers such as Transformer.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[2] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[3] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. "Language models are few-shot learners". In: *arXiv preprint arXiv:2005.14165* (2020).

[4] J. Zhu, Y. Xia, L. Wu, D. He, T. Qin, W. Zhou, H. Li, and T.-Y. Liu. "Incorporating bert into neural machine translation". In: *arXiv preprint arXiv:2002.06823* (2020).

[5] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. "A survey of methods for explaining black box models". In: *ACM computing surveys (CSUR)* 51.5 (2018), pp. 1–42.

[6] T. Miller. "Explanation in artificial intelligence: Insights from the social sciences". In: *Artificial Intelligence* 267 (2019), pp. 1–38.

[7] F. Bodria, F. Giannotti, R. Guidotti, F. Naretto, D. Pedreschi, and S. Rinzivillo. "Benchmarking and Survey of Explanation Methods for Black Box Models". In: *CoRR* abs/2102.13076 (2021).

[8] G. A. Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

[9] R. Guidotti, A. Monreale, S. Matwin, and D. Pedreschi. "Black Box Explanation by Learning Image Exemplars in the Latent Feature Space". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2019, pp. 189–205.

[10] R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, S. Ruggieri, and F. Turini. "Factual and counterfactual explanations for black box decision making". In: *IEEE Intelligent Systems* 34.6 (2019), pp. 14–23.

[11] M. Ahmed and A. N. Islam. "Deep learning: hope or hype". In: *Annals of Data Science* (2020), pp. 1–6.

[12] M. T. Ribeiro, S. Singh, and C. Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1135–1144.

[13] S. M. Lundberg and S.-I. Lee. "A unified approach to interpreting model predictions". In: *Advances in neural information processing systems*. 2017, pp. 4765–4774.

[14] R Guidotti, A Monreale, and L Cariaggi. "Investigating neighborhood generation for explanations of image classifiers". In: PAKDD. 2019.

[15] S. Lipovetsky and M. Conklin. "Analysis of regression in game theory approach". In: *Applied Stochastic Models in Business and Industry* 17.4 (2001), pp. 319–330.

[16] J. Li, S. Ji, T. Du, B. Li, and T. Wang. "Textbugger: Generating adversarial text against real-world applications". In: *arXiv preprint arXiv:1812.05271* (2018).

[17] H. Liu, Q. Yin, and W. Y. Wang. "Towards Explainable NLP: A Generative Explanation Framework for Text Classification". In: *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. Ed. by A. Korhonen, D. R. Traum, and L. Màrquez. Association for Computational Linguistics, 2019, pp. 5570–5581. DOI: 10.18653/v1/p19-1560. URL: https://doi.org/10.18653/v1/p19-1560.

[18] S. Ouyang, A. Lawlor, F. Costa, and P. Dolog. "Improving explainable recommendations with synthetic reviews". In: *arXiv preprint arXiv:1807.06978* (2018).

[19] I. Mollas, N. Bassiliades, and G. Tsoumakas. “LioNets: local interpretation of neural networks through penultimate layer decoding”. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2019, pp. 265–276.

[20] B. Hoover, H. Strobelt, and S. Gehrmann. “exbert: A visual analysis tool to explore learned representations in transformers models”. In: *arXiv preprint arXiv:1910.05276* (2019).

[21] M. Sarvmaili, A. Soares, R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, and S. Matwin. “A modularized framework for explaining hierarchical attention networks on text classifiers”. In: *Proceedings of the Canadian Conference on Artificial Intelligence* (June 8, 2021). https://caiac.pubpub.org/pub/zzjy8kzu. DOI: 10.21428/594757db.23db72bf. URL: https://caiac.pubpub.org/pub/zzjy8kzu.

[22] J. R. Firth. *Selected papers of JR Firth, 1952-59*. Indiana University Press, 1968.

[23] M. Wu, S. Parbhoo, M. C. Hughes, V. Roth, and F. Doshi-Velez. “Optimizing for Interpretability in Deep Neural Networks with Tree Regularization”. In: *arXiv preprint arXiv:1908.05254* (2019).

[24] L. Breiman. “Random Forests”. In: *Mach. Learn.* 45.1 (2001), pp. 5–32.

[25] R. Elshawi, M. H. Al-Mallah, and S. Sakr. “On the interpretability of machine learning-based model for predicting hypertension”. In: *BMC medical informatics and decision making* 19.1 (2019), pp. 1–32.

[26] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. “Learning Word Vectors for Sentiment Analysis”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, 2011, pp. 142–150. URL: http://www.aclweb.org/anthology/P11-1015.

[27] D. Tang, B. Qin, and T. Liu. “Document modeling with gated recurrent neural network for sentiment classification”. In: *Proceedings of the 2015 conference on empirical methods in natural language processing*. 2015, pp. 1422–1432.

[28] R. He and J. McAuley. “Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering”. In: *proceedings of the 25th international conference on world wide web*. 2016, pp. 507–517.

[29] A. Rane and A. Kumar. “Sentiment Classification System of Twitter Data for US Airline Service Analysis”. In: *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*. Vol. 01. 2018, pp. 769–773. DOI: 10.1109/COMPSAC.2018.00114.

[30] Y. Kim. “Convolutional neural networks for sentence classification”. In: *arXiv preprint arXiv:1408.5882* (2014).

[31] C. Hutto and E. Gilbert. “Vader: A parsimonious rule-based model for sentiment analysis of social media text”. In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 8. 1. 2014.

[32] P. Atanasova, J. G. Simonsen, C. Lioma, and I. Augenstein. “A Diagnostic Study of Explainability Techniques for Text Classification”. In: *arXiv preprint arXiv:2009.13295* (2020).

[33] Y. Freund and L. Mason. “The alternating decision tree learning algorithm”. In: *icml*. Vol. 99. Citeseer. 1999, pp. 124–133.

[34] M. Gick and S. Matwin. “The importance of causal structure and facts in evaluating explanations”. In: *Machine Learning Proceedings 1991*. Elsevier, 1991, pp. 51–54.

[35] F. Doshi-Velez and B. Kim. “Towards a rigorous science of interpretable machine learning”. In: *arXiv preprint arXiv:1702.08608* (2017).

[36] R. Guidotti, J. Soldani, D. Neri, A. Brogi, and D. Pedreschi. “Helping your docker images to spread based on explainable models”. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2018, pp. 205–221.

[37] R. Guidotti and A. Monreale. “Data-Agnostic Local Neighborhood Generation”. In: *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2020, pp. 1040–1045.

[38] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. “LOF: identifying density-based local outliers”. In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 2000, pp. 93–104.