Does It Make Sense to Explain a Black Box With Another Black Box?

Anonymous ACL submission

Abstract

Although counterfactual explanations are a popular approach to explain ML black-box classifiers, they are less widespread in NLP. Most methods find those explanations by iteratively perturbing the target document until it is classified differently by the black box. We identify two main families of counterfactual explanation methods in the literature, namely, (a) *transparent* methods that perturb the target by adding, removing, or replacing words, and (b) *opaque* approaches that project the target document into a latent, non-interpretable space where the perturbation is carried out subsequently. This article offers a comparative study of the performance of these two families of methods on three classical NLP tasks. Our empirical evidence shows that opaque approaches can be an overkill for downstream applications such as fake news detection or sentiment analysis since they add an additional level of complexity with no significant performance gain. These observations motivate our discussion, which raises the question of whether it makes sense to explain a black box using another black box.

1 Introduction

The latest advances in machine learning (ML) have led to significant advances in various natural language processing (NLP) tasks (Devlin et al., 2019; Liu et al., 2019; Sanh et al., 2019), such as text generation, fake news detection, sentiment analysis, and spam detection. These notable improvements can be partly attributed to the adoption of methods that encode and manipulate text data using latent representations. Those methods embed text into high-dimensional vector spaces that capture the underlying semantics and structure of language, and that are suitable for complex ML models.

Despite the impressive gains in accuracy achieved by modern ML algorithms (Devlin et al., 2019; Brown et al., 2020), their utility can be diminished by their lack of interpretability (Shen et al., 2020). This has, in turn, raised an increasing interest in ML explainability, the task of providing appropriate explanations for the answers of black-box ML algorithms (Jacovi, 2023). Indeed, a model could make correct predictions for the wrong reasons (Gururangan et al., 2018; McCoy et al., 2019). Unless the ML model is a white box, explaining the results of such an agent requires an explanation layer that elucidates the internal workings of the black box in a post-hoc manner.

While there are several ways to explain the outcomes of an ML model a posteriori, there has been a growing emphasis on counterfactual explanations, a domain that has experienced notable popularity over the last five years (Guidotti, 2022; Miller, 2019). A counterfactual explanation is a counter-example that is similar to the original text, but that elicits a different outcome in the black box (Wachter et al., 2018). Consider the classifier depicted in Figure 1, for sentiment analysis applied to the review "This is a good article" – classified as positive. In this toy example, a counterfactual could be the phrase "This is a **poor** article". This explanation tells us that the adjective "good" was a possible reason for this sentence to be classified as positive, and changing the polarity of that adjective may change the classifier's response.

Counterfactual explanation methods operate by increasingly perturbing the target text until the answer from the model – often a classifier – changes. Those perturbations can be conducted *transparently* by adding, removing, or changing words and syntactic groups (Martens and Provost, 2014; Yang et al., 2020; Ross et al., 2021) in the original target text as depicted in Figure 1. Since removing or adding words from a text can lead to unrealistic texts, more recent methods (Hase and Bansal, 2020; Robeer et al., 2021; S. Punla and C. Farro, 2022) embed the target text in a latent space that captures the underlying distribution of the model's training corpus. Perturbations are then carried out

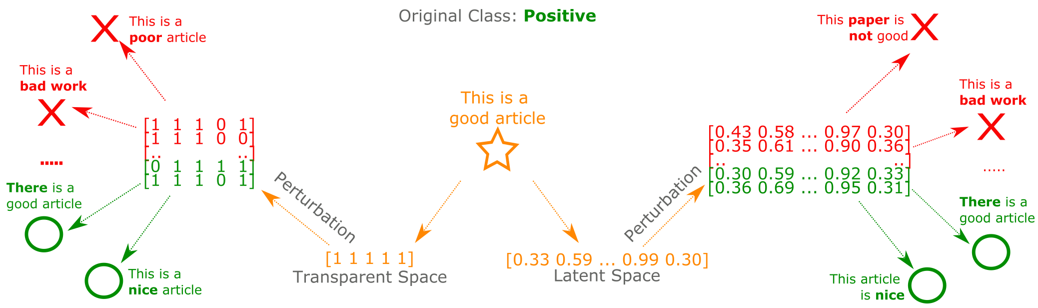


Figure 1: The mechanism employed to perturb the target documents by the transparent and opaque methods. Transparent techniques, on the left, convert the input text to a vector representation, where ‘1’ indicates the presence of the input word and ‘0’ denotes a replacement. Opaque methods, as on the right, embed words from the target text into a latent space and perturb the text in this high-dimensional space.

in this space and then brought back to the space of words to guarantee realistic counterfactual explanations. These explanation methods rely on *opaque* sophisticated techniques to compute those explanations (Li et al., 2021), which is tantamount to explaining a black box with another black box.

Based on this somehow paradoxical observation, we conduct a comparative study of various transparent and opaque post-hoc counterfactual explanation approaches. Rather than two distinct categories, the studied methods define a continuum, as some methods may combine transparent and non-interpretable techniques. Our study aims to understand whether it is worth resorting to latent approaches to explain complex ML models. The experimental results suggest that for some downstream NLP tasks, learning a latent representation for explanation purposes can be an overkill. To strengthen our point, we present and evaluate two novel transparent approaches for counterfactual explanations.

The paper is structured as follows. Section 2 surveys the existing counterfactual explanation methods. Section 3 introduces two novel transparent methods, which we then analyze in the light of the spectrum of existing transparent and opaque techniques (Section 4). We then elaborate on the experimental protocol of our comparative study in Section 5. The results of our experimentations are presented in Section 6. Section 7 discusses our findings and concludes the paper.

2 Related Works

Counterfactual explanation methods compute contrastive explanations for ML black-box algorithms by providing examples that resemble a target instance but that lead to a different answer in the black box (Wachter et al., 2018). These counterfactual explanations convey the minimum changes in the input that would modify a classifier’s outcome. Social sciences (Miller, 2019) have shown that human explanations are contrastive and Wachter et al. (2018) have illustrated the utility of counterfactual instances in computational law. When it comes to NLP tasks, a good counterfactual explanation should be fluent (Morris et al., 2020), *i.e.*, read like something someone would say, and be sparse (Pearl, 2009), *i.e.*, look like the target instance.

Counterfactual approaches have gained popularity in the last few years. As illustrated by the surveys, first by Bodria et al. (2021) and later by Guidotti (2022), around 50 additional counterfactual methods appeared in a one-year time span. Despite this surge of interest in counterfactual explanations, their study for NLP applications remains underdeveloped (Ross et al., 2021). In the following, we elaborate on the existing counterfactual explanation methods for textual data along a spectrum that spans from transparent to opaque approaches.

Transparent Approaches. Given an ML classifier and a target text (also called a document), transparent techniques compute counterfactual explanations in a binary space. Each dimension represents the presence (1) or absence (0) of a word from a given vocabulary. Hence, to perturb a text, these methods toggle on and off 0s and 1s, where 0s are tantamount to adding, removing, or replacing words until the classifier yields a different answer. This was first proposed by Martens and Provost (2014) who introduced Search for Explanations for Document Classification (SEDC), a method that removes the words for which the classifier exhibits the highest *sensitivity*. More recently, Ross et al. (2021) developed Minimal Contrastive Editing (MICE), a method that employs a Text-To-Text Transfer Transformer to fill masked sentences.

Yang et al. (2020) presented Plausible Counterfactual Instances Generation (PCIG), which generates grammatically plausible counterfactuals through edits of single words with lexicons manually selected from the economics domain.

Opaque Methods. We define opaque approaches as those perturbing the input text in a latent space in $\mathbb{R}^n$. Methods such as Decision Boundary (Hase and Bansal, 2020), xSPELLS (S. Punla and C. Farro, 2022) or cfGAN (Robeer et al., 2021) operate in three phases. First, they embed the target text onto a latent space. This is accomplished by employing specific techniques such as Variational AutoEncoder (VAE) in the case of xSPELLS, or a pre-trained language model (LM) for cfGAN. Second, while the classifier's decision boundary is not traversed, these methods perturb the latent representation of the target phrase. This is done by adding Gaussian noise in the case of xSPELLS, whereas cfGAN resorts to a Conditional Generative Adversarial Network. Finally, a decoding stage produces sentences from the latent representation of the perturbed documents.

There also exist methods such as Polyjuice (Wu et al., 2021), Generate Your Counterfactuals (GYC) (Madaan et al., 2021) and Tailor (Ross et al., 2022) that perturb text documents in a latent space, but can be instructed to change particular linguistic aspects of the target text, such as locality or grammar tense. Such methods are not particularly designed to compute counterfactual explanations but are rather conceived for other applications such as data augmentation.

Unlike pure word-based perturbation methods, latent representations are good at preserving *semantic closeness* for small perturbations. That said, these methods are not free of pitfalls. First, methods such as xSPELLS and cfGAN are deemed opaque since a latent space is not human-understandable (Shen et al., 2020). Moreover, existing latent-based approaches do not seem optimized for sparse counterfactual explanations – one of the defining features of a counterfactual. We show this through our experimental results that suggest that a minor alteration in the latent space can cause a significant alteration in the original space.

3 Two Novel Transparent Methods

Before elaborating on our study, we introduce two novel counterfactual explanation techniques, aimed to enrich the middle ground between fully opaque and fully transparent approaches. The methods are called Growing Language and Growing Net, and both depend on an iterative process that replaces words within a target text $x = (x_1, \ldots, x_d) \in X$ ($x_i \in \Sigma$ are words from a vocabulary Σ) until the predicted class of a given classifier $f : X \to Y$ changes. The goal of such a procedure is to compute sparse counterfactual explanations with the fewest modified words.

Algorithm 1 Explore

Require: target text $x = (x_1, \ldots, x_d) \in X$, classifier f;
  $\textsc{simwords}(\cdot) \to$ retrieves similar words
  Hyper-parameters: $n = 2000$
Ensure: one or multiple counterfactual instances
1: **for** $i \leftarrow 1$ **to** d **do**
2: $\quad W_i \leftarrow \textsc{simwords}(x_i, \mathit{POS}(x_i))$
3: **end for**
4: Initialize $Z = (z^1, \ldots, z^n)$ as n copies of x
5: Initialize $C \leftarrow \emptyset$; $n_m \leftarrow 0$
6: **while** $n_m < d \land C = \emptyset$ **do**
7: $\quad n_m \leftarrow n_m + 1$
8: $\quad$**for** $j \leftarrow 1$ **to** n **do** $\quad\triangleright$ For each copy of x
9: $\quad\quad$**for** $l \leftarrow 1$ **to** n_m **do**
10: $\quad\quad\quad k \leftarrow \mathit{random}(0, d)$ $\quad\triangleright\ k : z^j_k = x_k$
11: $\quad\quad\quad z^j_k \leftarrow$ random word from W_k
12: $\quad\quad$**end for**
13: $\quad\quad$**if** $f(x) \neq f(z_j)$ **then**
14: $\quad\quad\quad C \leftarrow C \cup \{z_j\}$
15: $\quad\quad$**end if**
16: $\quad$**end for**
17: **end while**
18: **return** C

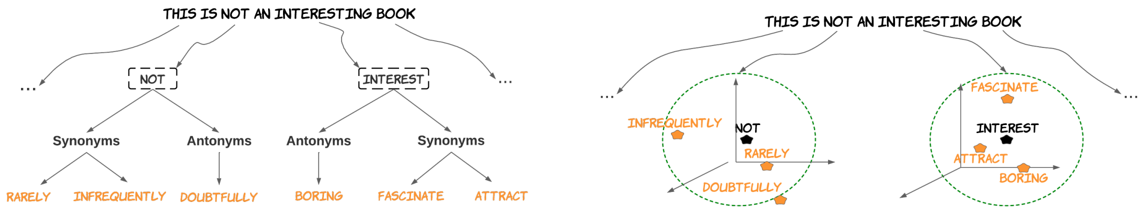
Algorithm 2 Growing Net

Require: a target text $x = (x_1, \ldots, x_d) \in X$, classifier f;
$C \leftarrow \mathit{explore}(x, f, \textsc{wn\_simwords}_{d=1}(\cdot))$
return $\mathit{argmax}_{c \in C}\, \mathit{Wu\text{-}P}(c, x)$

Algorithm 3 Growing Language

Require: target text $x = (x_1, \ldots, x_d) \in X$, classifier f;
  Hyper-parameters: $\tau = 0.02$; $\theta = 0.9$; $\theta_{min} = 0.4$;
$C \leftarrow \emptyset$
while $\theta > \theta_{min} \land C = \emptyset$ **do**
$\quad C \leftarrow C \cup \mathit{explore}(x, f, \textsc{lm\_simwords}_{\theta}(\cdot))$
$\quad \theta \leftarrow \theta - \tau$
end while
return $\mathit{argmin}_{c \in C} ||x - c||_0$

Algorithm 1 outlines the iterative exploration process employed by Growing Language and Growing Net. In the first step (lines 1 to 3), both approaches generate d sets of potential word replacements $W_1, \ldots, W_d$ for each word x_i in the target document x. Those replacements must have the same part-of-speech (POS) tag as x_i. The external module to obtain those word replacements depends on the method. These modules are detailed later. Subsequently, our methods create arti-



(a) Growing Net (b) Growing Language

Figure 2: The mechanism to compute potential word replacements in Growing Net navigates the tree structure of WordNet. Conversely, Growing Language embeds words into a latent space on which it looks for nearby words.

ficial documents iteratively (lines 6 and 17) while some words in the original document remain non-replaced ($n_m < d$), or while we have not found any counterfactuals. At each iteration, the exploration keeps n copies of the original text (x) on which we replace n_m individual words (x_k) with randomly selected words from their respective sets of potential replacements (W_k). Lines 13-15 check if the resulting phrases are counterfactual instances.

For example, consider the target review, *"This is not an interesting book"*, classified as negative by a sentiment analysis model. In the first round, our routine produces artificial reviews with only one modified word. Subsequent rounds will replace two words and so on (lines 9 to 12).

Growing Net. This method capitalizes on the rich structure of WordNet (Fellbaum, 1998) to identify potential word replacements. WordNet is a lexical database and thesaurus that organizes words and their meanings into a semantic tree of interrelated concepts. The method is described in Algorithm 2, and uses the module WN_SIMWORDS$_d$. In the exploration phase, Growing Net uses WN_SIMWORDS$_d$ to find words at a distance of at most d in the WordNet hierarchy among synonyms, antonyms, hyponyms, and hypernyms for a given word x_i to replace. This process is illustrated in Figure 2a. In our experiments we set $d = 1$ as this value already yields good results – higher values would incur longer runtimes. The exploration returns a set of counterfactuals, from which Growing Net selects the one with the highest Wu-Palmer Similarity (Wu-P) (Wei and Ngo, 2007) as final explanation. This similarity score for text relies on Wordnet, and takes into account the relatedness of the concepts in the phrase, e.g., via the path length to their most common ancestor in the hierarchy.

Growing Language. This approach leverages the power of language models (LM) to restrict the space of possible word replacements via the module LM_SIMWORDS$_\theta$ (see Algorithm 3). Given a word x_i to replace, LM_SIMWORDS$_\theta$ embeds the word onto the latent space of an LM, as illustrated in Figure 2b. Then LM_SIMWORDS$_\theta$ retrieves words whose latent representation is at a distance of θ at most. In our experiments, we initially set this threshold to 0.8 on a scale from 0 to 1. If for a given θ, Growing Language cannot find counterfactual instances, the distance threshold is relaxed, i.e., reduced by τ (set to 0.02 in our experiments), so that the exploration routine considers more words. Should multiple counterfactuals be found, Growing Language selects the one with the fewest modifications compared to the original document (minimal L0 distance). For our experiments, we employed Spacy (Honnibal and Montani, 2017), but any language model capable of embedding words and offering word distances could be applied in this context.

4 Interpretability Spectrum

We have presented counterfactual explanation techniques as either opaque or transparent. However, the landscape is more nuanced, for these techniques actually define a spectrum, which we depict in Figure 3. The spectrum spans from the most transparent methods on the left to the most opaque ones on the right. We elaborate on the various regions of this spectrum in the following.

Fully Transparent. At the leftmost end of the spectrum, we find the method SEDC (Martens and Provost, 2014), which perturbs text instances by hiding only highly sensitive words within the text. We place Growing Net on the right of SEDC, because it goes beyond simple word masking. Instead, it substitutes words judiciously via an external interpretable asset, namely Wordnet.

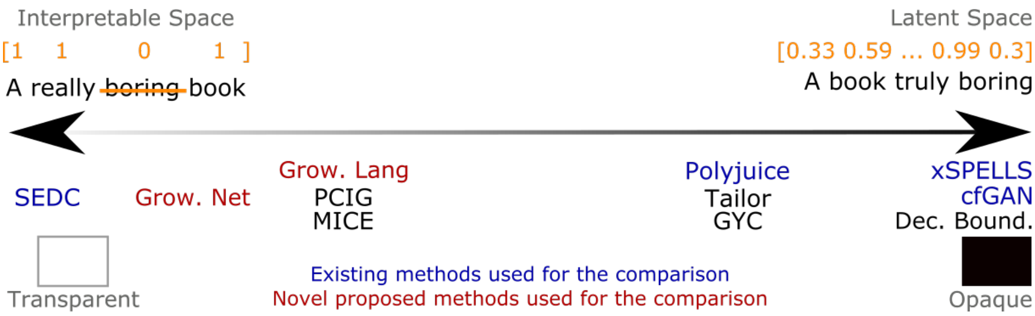


Figure 3: Spectrum for counterfactual explanation techniques that goes from the most transparent methods on the left to the most opaque on the right. Transparent methods perturb documents in a binary space; opaque methods do it in a latent space.

Transparent. Methods like PCIG (Yang et al., 2020), MICE (Ross et al., 2021), and Growing Language are considered more opaque than Growing Net, because they employ a latent space to identify semantically close word substitutions. Despite this reliance on black-box techniques, we consider them transparent because the search for counterfactuals is still carried out in the space of words.

Partially Opaque. Polyjuice, Tailor, and GYC fall in the category of partially opaque methods, as they leverage control codes to perturb the target document. Control codes are specific instructions that adapt the perturbation of the target text so that it complies with a specific task, such as translating, summarizing, or changing the tense of a text. While these modifications occur in a latent space, the inclusion of control codes provides some level of clarity regarding why a modification influences the model's prediction.

Fully Opaque. On the far right of the interpretability spectrum, we encounter fully opaque approaches such as Decision Boundary, xSPELLS and cfGAN. These methods perturb instances in a latent space, making it challenging for users to discern the underlying process of counterfactual generation.

This interpretability spectrum provides valuable insights into the transparency and opacity of counterfactual explanation methods, allowing for a more nuanced understanding of their capabilities.

5 Experimental Protocol

Having introduced the spectrum of counterfactual explanation methods across the interpretability axis, we now describe the experimental setup designed to evaluate those methods. The code of the studied methods, the datasets, and the experimental results are available at https://anonymous.4open.science/r/ebbwbb-4B55/README.md

5.1 Methods

We picked a set of representative domain-agnostic methods from all regions of the spectrum depicted in Figure 3. These include SEDC and Growing Net among the fully transparent methods, Growing Language among the transparent ones[1], Polyjuice among the partially opaque ones, and xSPELLS and cfGAN from the fully opaque group.

5.2 Tasks & Datasets

We conduct the evaluation on three popular downstream tasks: (a) spam detection in messages, (b) sentiment analysis, and (c) detection of fake news from newspaper headlines. The datasets associated to these tasks consist of two target classes, and contain between 4000 and 10660 textual documents. The average number of words in each document is between 11.8 and 20.8 as reported in Table 1. Except for the fake news dataset, we downloaded the data from Kaggle. The fake news dataset was constructed by us and its description is available in our repository https://anonymous.4open.science/r/ebbwbb-4B55/README.md.

Dataset	No. of words		Instances	Accuracy (%)		
	Total	Average		MLP	RF	BERT
Fake	19419	11.8	4025	84	84	91
Polarity	11646	20.8	10660	72	67	82
Spam	15587	18.5	8559	100	100	100

Table 1: Information about the experimental datasets. The "average" column denotes the average number of words per instance (document).

5.3 Black-box Classifiers

Our evaluation uses two distinct black-box classifiers implemented using the scikit-learn library and

[1] PCIG relies on domain specific rules from economics; MICE is computationally expensive according to the authors.

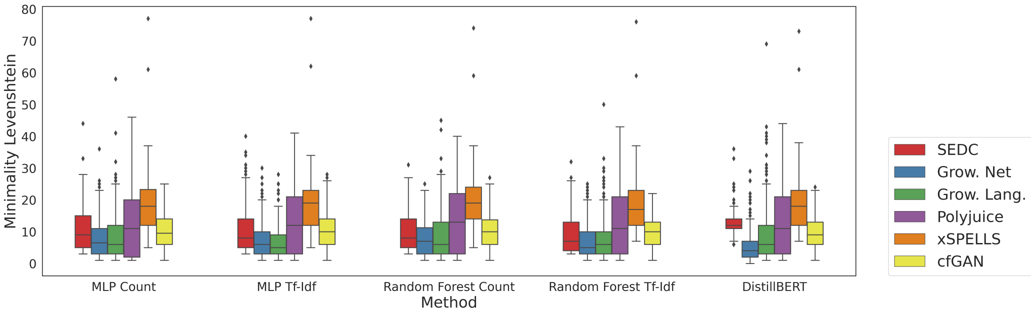


Figure 4: Minimality as the levenshtein edit distance between the closest counterfactual and the target text (↓ better).

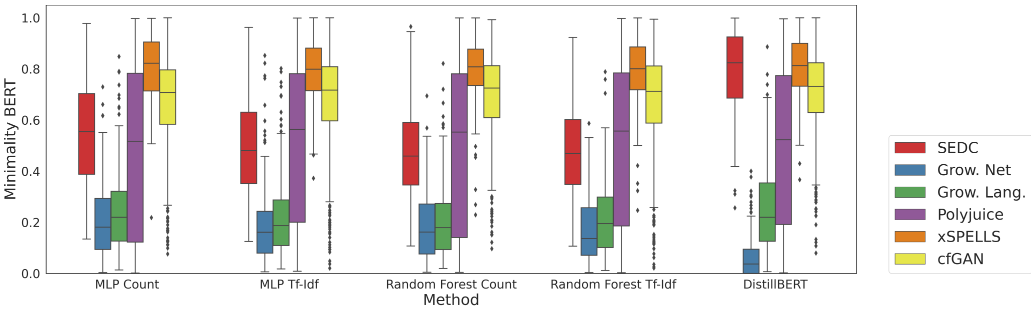


Figure 5: Minimality as the Sentence-BERT embedding distance between the closest counterfactual and the target text (↓ better).

already employed in (S. Punla and C. Farro, 2022). These black boxes are (i) a Random Forest (RF) consisting of 500 tree estimators, (ii) a multi-layer perceptron (MLP) with token counts as input, and (iii) a classifier based on DistillBERT[2]. For the RF and the MLP, we employed both the *token count* and *tf-idf* vectorizers to convert text into proper inputs for the models.

We used 70% of the instances for training, and the remaining for testing. The classifiers' test performances are shown in Table 1. The counterfactual explanations were computed for instances in those test sets.

6 Results

We now present the results of our evaluation, organized in four rounds of experiments categorized according to two aspects. First, we assess the quality of the produced counterfactual explanations based on two essential criteria: (i) **minimality**, and (ii) **plausibility**. Second, we evaluate the methods themselves in terms of (iii) **flip change**, and (iv) **runtime**. For each evaluated method and black-box classifier, we computed counterfactual explanations for 100 target texts extracted from the test sets of our datasets.

[2] https://is.gd/zljjJN

6.1 Counterfactual Quality

A high-quality textual counterfactual explanation tells us what are the most sensitive parts or aspects of the target phrase, that otherwise changed, would lead to a different classification outcome. It follows then that such an explanation must (i) incur minimal changes w.r.t the target phrase (sparse changes), and (ii) be linguistically plausible, i.e., sound like something a person would naturally write or say (Guidotti, 2022).

Minimality. We quantify the minimality criterion by measuring the distance between the counterfactual and the target sentence. Figure 4 and 5 display the results of our minimality assessments, considering both the Levenshtein distance and the cosine similarity within the embedding space of the BERT-Sentence model (Reimers and Gurevych, 2019). This dual approach ensures a comprehensive evaluation, accounting for both lexical similarity and latent features, including aspects of style.

Notably, our findings reveal that methods positioned in the middle-ground, particularly Growing Net, performed favorably compared to opaque approaches, both in terms of the number of words modified and semantic comparison. It is worth noting that xSPELLS introduced the most significant changes to the original text – contradicting one of the main functional requirements of a counterfactual explanation (Wachter et al., 2018). Simi-

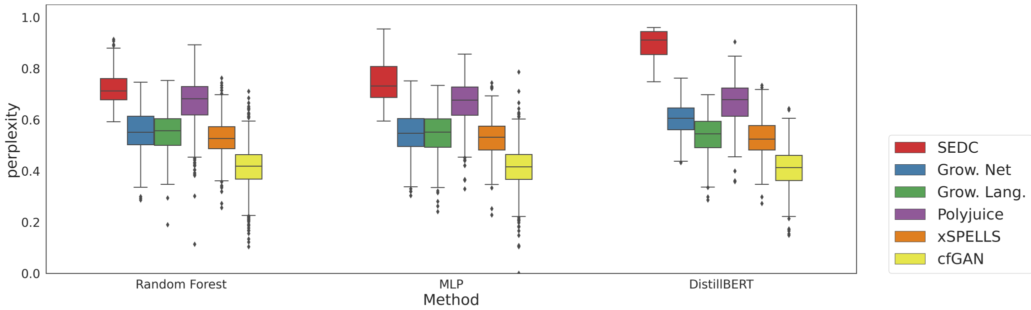


Figure 6: Perplexity as the MSE loss of a GPT model on the generated counterfactuals (↓ better).

Dataset	Fake			Spam			Polarity		
Black box	MLP	RF	BERT	MLP	RF	BERT	MLP	RF	BERT
SEDC	**0.95**	0.82	**1**	0.47	0.42	0.56	0.92	0.93	**0.98**
Grow. Net	0.90	0.8	0.88	0.44	0.29	0.84	**0.97**	**0.98**	0.90
Grow. Lang.	0.84	**0.84**	0.77	0.58	0.61	0.17	0.92	0.92	0.92
Polyjuice	0.26	0.23	0.21	0.17	0.14	0.16	0.33	0.31	0.29
xSPELLS	0.68	0.78	0.77	**0.98**	**0.95**	**0.91**	0.91	0.76	0.91
cfGAN	0.18	0.12	0.09	0.14	0.05	0.03	0.50	0.50	0.48

Table 2: Average label flip per dataset and black box of the six counterfactual methods (↑ better).

larly, we observe a high variance in the minimality of the counterfactuals generated by Polyjuice, indicating that some counterfactuals were notably distant from their corresponding target instances. While these methods introduced minor perturbations to the original text, these modifications occurred within a latent space. Nothing guarantees, however, that these minor adjustments translate into visually subtle modifications of the target phrase when the resulting phrase is brought back to the original space. As an example, consider the target text "This is one of Polanski's best films." from the polarity dataset. For the DistillBERT classifier, cfGAN returns the counterfactual "this is one of **shot kingdom intelligence' s all**", which looks completely unrelated to the target text. Conversely, the transparent method SEDC produces the counterfactual "This is one of **MASK MASK MASK**", whereas Growing Language outputs "This is one of Polanski's **worst** films." .

Additionally, we noted first that when the complexity of the classifier increases, the counterfactual explanations generated by SEDC lie farther from the original text. Secondly, we observe minor variations dependent on the vectorizer employed by the classifiers (*count* or *tf-idf*). Hence, for the subsequent phase of the evaluation, we present results exclusively for the tf-idf vectorizer.

Plausibility. While linguistic plausibility is typically evaluated through user studies (Madaan et al., 2021; Wu et al., 2021), we approximate it here following the techniques from Ross et al. (2021, 2022). Thus, we use perplexity scores based on a GPT language model (Brown et al., 2020), by calculating the average mean squared error (MSE) loss when predicting every token in the counterfactual from the previous ones. Figure 6 presents the plausibility of the counterfactuals. To enhance comparability, we normalized perplexity scores based on the maximum perplexity observed across the entire set of counterfactuals, where lower scores indicate higher plausibility. Notably, SEDC and Polyjuice generated texts with the lowest plausibility, which is expected since SEDC masks words, leading sometimes to nonsensical sentences. In contrast, cfGAN demonstrated the highest plausibility, while both Growing Net and Language achieved perplexity scores similar to those of xSPELLS.

6.2 Method Quality

We now compare the quality of the counterfactual explanation methods themselves based on (iii) label flip rate, which measures how frequently a method produces an instance classified differently by the model, and (iv) runtime, the time it takes for each method to generate a counterfactual explanation.

dataset	method	MLP	RF	BERT
fake	SEDC	31 (14)	13 (6)	15 (3)
	Grow. Net	2 (1)	1 (1)	7 (1)
	Grow. Lang.	55 (28)	55 (13)	34 (12)
	Polyjuice	38 (8)	70 (185)	29 (4)
	cfGAN	1 (0)	1 (0)	1 (0)
	xSPELLS	84 (6)	86 (7)	16 (1)
spam	SEDC	21 (13)	16 (9)	16 (6)
	Grow. Net	1 (1)	1 (1)	11 (4)
	Grow. Lang.	60 (16)	57 (14)	88 (43)
	Polyjuice	32 (7)	62 (184)	33 (15)
	cfGAN	1 (0)	1 (0)	1 (0)
	xSPELLS	219 (17)	198 (16)	22 (1)
polarity	SEDC	13 (10)	12 (9)	21 (6)
	Grow. Net	1 (1)	1 (1)	9 (2)
	Grow. Lang.	75 (33)	74 (32)	65 (29)
	Polyjuice	81 (30)	82 (48)	29 (4)
	cfGAN	1 (0)	1 (0)	1 (0)
	xSPELLS	136 (19)	115 (11)	24 (2)

Table 3: Average runtime in seconds of the studied counterfactual methods (and standard deviation).

Label flip rate. Table 2 provides an overview of the label flip results. It is noteworthy that except for the spam dataset, transparent methods achieve the highest label flip rate. This highlights the effectiveness of replacing words with antonyms as a means to discover counterfactuals. Additionally, xSPELLS exhibits strong performance for the spam dataset and similar label flip rates to transparent methods on polarity. We also emphasize that both Growing Net and Growing Language can be fine-tuned for a more exhaustive search by adjusting their parameters, for example by lowering the minimal similarity threshold (θ_{min} in Alg. 3) or by going further in WordNet's tree structure (higher d in Alg. 2). While this can enhance the label flip rate, it may result in longer runtimes.

Runtime. Finally, Table 3 details the average and standard deviation of the runtime for each counterfactual explanation method across datasets and classifiers. Notably, cfGAN and Growing Net emerged as the fastest methods for generating counterfactuals. However, it is important to note that cfGAN requires the training of the Variational AutoEncoder (VAE) on each specific dataset, a process that incurs long training times. The time needed for fine-tuning varies, ranging from 4300 seconds for fake news title detection to 6755 seconds for spam detection. Furthermore, we observe that xSPELLS and Growing Language exhibit the slowest runtime performance. Growing Language, for instance, requires approximately 60 seconds to generate a single counterfactual, while xSPELLS exhibits runtimes, ranging from 16 seconds for fake news detection to 219 seconds for spam detection. These results reveal that, in contrast to opaque methods such as xSPELLS, transparent approaches like Growing Net are fast enough for real-time explainability.

7 Discussion & Conclusion

Our evaluation provides valuable insights into the landscape of counterfactual explanations for downstream NLP tasks. One of the most striking findings is that complexity, often associated with the use of neural networks and latent spaces, does not necessarily equate to superior performance in this context. Surprisingly, our results demonstrate that simpler approaches, characterized by a systematic and judicious strategy for word replacement, consistently yield satisfactory outcomes across all quality dimensions. The results of our study prompt a deeper reflection on the optimal strategies for generating counterfactual explanations in the field of NLP. It invites readers to embrace simplicity and transparency whenever the constraints of the application allow it.

Furthermore, our findings underscore the critical importance of transparency and interpretability in AI and ML, especially in high-stakes applications. The paradox of explaining a black box with another one calls into question the development of opaque approaches when transparent methods suffice, or when transparency is one of the goals in the first place. When focused on NLP applications, our results also call for reflection on the meaning and goal of explanations. If the task is to understand which aspects of a text should change to get a different outcome, a counterfactual explanation that drastically changes every word in the text may not be understandable. On the contrary, a counterfactual based on simple word-masking, albeit simple, may be perceived as implausible. This could hamper the goal of explanations as a means to elicit trust in users.

We therefore expect our findings to encourage the development of more transparent and interpretable AI systems that foster trust and accountability in every step of the AI-driven decision-making processes, either for prediction, recommendation, or explanation. Last but not least, we believe that the lessons drawn from this paper could be naturally ported to other explanation paradigms.

Limitations

We remind the reader that the evaluation was conducted on three well-studied downstream applications, namely polarity analysis, fake news detection, and spam detection. Our results might therefore not generalize to other NLP tasks in specialized domains or different languages. While this work puts transparent approaches in the spotlight, our results suggest that plausible counterfactual examples need external domain-adapted knowledge either in the form of language models or knowledge graphs. These may not always be available though. Finally, our evaluation was based on popular criteria and metrics for counterfactual explanations. Specialized applications may still take into account additional criteria such as diversity or actionability.

References

Francesco Bodria, Fosca Giannotti, Riccardo Guidotti, Francesca Naretto, Dino Pedreschi, and Salvatore Rinzivillo. 2021. Benchmarking and survey of explanation methods for black box models. *CoRR*, abs/2102.13076.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proc. NeurIPS*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pages 4171–4186. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Riccardo Guidotti. 2022. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proc. NAACL-HLT*. Association for Computational Linguistics.

Peter Hase and Mohit Bansal. 2020. Evaluating explainable AI: which algorithmic explanations help users predict model behavior? In *Proc. ACL*. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural lanugage understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Alon Jacovi. 2023. Trends in explainable AI (XAI) literature. *CoRR*, abs/2301.05433.

Ziqiang Li, Rentuo Tao, Jie Wang, Fu Li, Hongjing Niu, Mingdao Yue, and Bin Li. 2021. Interpreting the latent space of gans via measuring decoupling. *IEEE Trans. Artif. Intell.*, 2(1):58–70.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*.

Nishtha Madaan, Inkit Padhi, Naveen Panwar, and Diptikalyan Saha. 2021. Generate your counterfactuals: Towards controlled counterfactual generation for text. In *Thirty-Fifth Conference on Artificial Intelligence, AAAI, Conference on Innovative Applications of Artificial Intelligence, IAAI, The Symposium on Educational Advances in Artificial Intelligence, EAAI*. AAAI Press.

David Martens and Foster J. Provost. 2014. Explaining data-driven document classifications. *MIS Q.*, 38(1):73–99.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proc. ACL*. Association for Computational Linguistics.

Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.*, 267:1–38.

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proc. EMNLP*. Association for Computational Linguistics.

Judea Pearl. 2009. Causal inference in statistics: An overview. *Statistics Surveys*, 3(none):96 – 146.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proc. EMNLP*. Association for Computational Linguistics.

Marcel Robeer, Floris Bex, and Ad Feelders. 2021. Generating realistic natural language counterfactuals. In *Findings EMNLP*. Association for Computational Linguistics.

Alexis Ross, Ana Marasovic, and Matthew E. Peters. 2021. Explaining NLP models via minimal contrastive editing (mice). In *Findings ACL/IJCNLP*. Association for Computational Linguistics.

Alexis Ross, Tongshuang Wu, Hao Peng, Matthew E. Peters, and Matt Gardner. 2022. Tailor: Generating and perturbing text with semantic controls. In *Proc. ACL*. Association for Computational Linguistics.

Candida S. Punla and Rosemarie C. Farro. 2022. Are we there yet?: An analysis of the competencies of BEED graduates of BPSU-DC. *International Multidisciplinary Research Journal*, 4(3):50–59.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. 2020. Interpreting the latent space of gans for semantic face editing. In *Proc. CVPR*. Computer Vision Foundation / IEEE.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2018. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law and Technology*, 31(2):841–87.

Xiao-Yong Wei and Chong-Wah Ngo. 2007. Ontology-enriched semantic space for video search. In *Proc. International Conference on Multimedia*. ACM.

Tongshuang Wu, Marco Túlio Ribeiro, Jeffrey Heer, and Daniel S. Weld. 2021. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *Proc. ACL/IJCNLP*. Association for Computational Linguistics.

Linyi Yang, Eoin M. Kenny, Tin Lok James Ng, Yi Yang, Barry Smyth, and Ruihai Dong. 2020. Generating plausible counterfactual explanations for deep transformers in financial text classification. In *Proc. COLING*. International Committee on Computational Linguistics.