Expliquer une boîte noire sans boîte noire

Julien Delaunay, Luis Galárraga, Christine Largouët

Expliquer une boîte noire sans boîte noire

Julien Delaunay* — **Luis Galárraga*** — **Christine Largouët** **

* *Université de Rennes, Inria/IRISA Rennes, France*
** *Université de Rennes, Institut Agro/IRISA Rennes, France*

RÉSUMÉ. *Les méthodes d'explication contrefactuelle sont des approches populaires pour expliquer les algorithmes d'apprentissage automatique. Ces explications encodent les modifications nécessaires dans un document cible pour modifier la prédiction d'un classificateur. La plupart de ces méthodes trouvent ces explications en perturbant de manière itérative le document cible jusqu'à ce qu'il soit classifié différemment par la boîte noire. Nous identifions deux principales familles d'approches contrefactuelles dans la littérature, à savoir (a) les méthodes « transparentes » qui perturbent la cible en ajoutant, en supprimant ou en remplaçant des mots, et (b) les techniques « opaques » qui projettent le document cible dans un espace latent non interprétable dans lequel la perturbation est ensuite effectuée. Cet article propose une étude comparative des performances de ces deux familles de méthodes sur trois tâches classiques en traitement du langage naturel. Nos résultats montrent que pour les applications telles que la détection de fausses informations ou l'analyse des sentiments, les approches contrefactuelles opaques peuvent rajouter un niveau de complexité sans amélioration significative.*

MOTS-CLÉS : *explicabilité, interpretabilité, contrefactuel, traitement automatique des langues.*

TITLE. *Explaining a Black Box Without a Black Box*

ABSTRACT. *Counterfactual Explanation Methods are popular approaches to explain ML black-box classifiers. A counterfactual explanation encodes the smallest changes required in a target document to modify a classifier's output. Most counterfactual 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 approaches in the literature, namely, (a)* transparent *methods that perturb the target by adding, removing, or replacing words, and (b)* opaque *techniques that project the target document onto a latent 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 counterfactual approaches can be overkill for applications such as fake news detection or sentiment analysis since they add a supplementary level of complexity with no significant improvement.*
KEYWORDS: *Explainability, Interpretability, Counterfactual, Natural Language Processing.*

1. Introduction

Les progrès récents en apprentissage automatique ont considérablement transformé de nombreuses tâches en traitement automatique du langage naturel (TALN) (Liu *et al.*, 2019 ; Devlin *et al.*, 2019 ; Sanh *et al.*, 2019), notamment la génération de texte, la détection de fausses informations, l'analyse des sentiments et la détection de spams. Ces améliorations peuvent être en partie attribuées à l'adoption de méthodes qui encodent et qui manipulent les données textuelles à l'aide de représentations latentes. Ces méthodes intègrent le texte dans des espaces vectoriels de haute dimension qui capturent la sémantique sous-jacente et la structure du langage, ce qui convient aux modèles de *Machine Learning* (ML) complexes.

Toutefois, cette avancée en précision des algorithmes modernes, tels que les modèles *Transformers* (Devlin *et al.*, 2019), s'accompagne souvent d'une limitation en termes d'interprétabilité (Shen *et al.*, 2020). Cette dépendance à l'égard de modèles boîtes noires a suscité un intérêt croissant pour l'explicabilité des modèles d'apprentissage automatique, c'est-à-dire la capacité à fournir des explications aux prédictions des algorithmes (Jacovi, 2023). En effet, certains de ces résultats peuvent être remis en question, car ces modèles exploitent des informations lexicales (et d'autres heuristiques) présentes dans les ensembles de données, ce qui peut les amener à donner des réponses correctes pour de mauvaises raisons (Gururangan *et al.*, 2018 ; McCoy *et al.*, 2019). À moins que le modèle d'apprentissage automatique ne soit une boîte blanche, expliquer les résultats de cet agent nécessite l'introduction d'une couche d'explication qui interprète le fonctionnement interne de la boîte noire *a posteriori*. Cette démarche est couramment désignée « explicabilité *post hoc* ».

Il existe plusieurs moyens d'expliquer les résultats d'un modèle d'apprentissage automatique *a posteriori*. Parmi les différentes approches, les explications contrefactuelles ont gagné en popularité au cours des cinq dernières années (Miller, 2019 ; Guidotti, 2022). Prenons l'exemple représenté dans la figure 1, d'un classificateur d'analyse de sentiments appliqué à la critique de livre et le commentaire « Ceci est un bon article » – classifié comme positif. Une explication contrefactuelle est un contre-exemple similaire au texte original, mais qui suscite une prédiction différente par la boîte noire (Wachter *et al.*, 2018). Dans cet exemple fictif, un contre-exemple pourrait être la phrase « Ceci est un **mauvais** article ». Grâce à cette explication, la technique contrefactuelle transmet que l'adjectif « bon » était une raison possible pour laquelle cette phrase a été classifiée comme positive, et changer la polarité de cet adjectif peut modifier la réponse du classificateur.

Dans la littérature, les méthodes d'explication contrefactuelle fonctionnent généralement en perturbant itérativement le texte cible jusqu'à ce que la réponse du modèle change (Verma *et al.*, 2020 ; Guidotti, 2022). Ces perturbations peuvent être réalisées de manière « transparente » en ajoutant, en supprimant ou en modifiant des mots et des groupes syntaxiques (Martens et Provost, 2014 ; Yang *et al.*, 2020 ; Ross *et al.*, 2021) dans le texte cible original, comme illustré dans la figure 1. Étant donné que la suppression ou l'ajout de mots dans un texte peut conduire à des textes irréalistes,

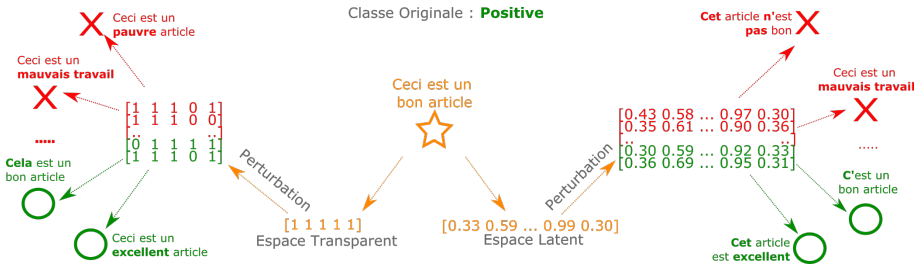

FIGURE 1. *Le mécanisme utilisé pour perturber les documents cibles par les méthodes transparentes et opaques. L'instance cible est représentée par la phrase « Ceci est un bon article », tandis que les autres textes sont des documents textuels artificiels. Les techniques transparentes, à gauche, convertissent le texte d'entrée en une représentation vectorielle, où « 1 » indique la présence du mot d'origine et « 0 » indique un remplacement. Les méthodes opaques, à droite, intègrent les mots du texte cible dans un espace latent et perturbent le texte dans cet espace multidimensionnel.*

des méthodes plus récentes (Hase et Bansal, 2020 ; Robeer *et al.*, 2021 ; Lampridis *et al.*, 2022) convertissent le texte cible dans un espace latent qui capture la distribution sous-jacente du corpus d'entraînement du modèle. Les perturbations sont ensuite effectuées dans cet espace puis ramenées à l'espace des mots pour garantir des explications contrefactuelles réalistes. Ces méthodes d'explication reposent sur des techniques « opaques » sophistiquées pour calculer ces explications (Li *et al.*, 2021), ce qui revient à expliquer une boîte noire avec une autre boîte noire.

Sur la base de cette observation quelque peu paradoxale, nous menons une étude comparative de différentes approches transparentes et opaques d'explication contrefactuelle *a posteriori*, afin de mettre en lumière les avantages de l'une par rapport à l'autre. Nos analyses empiriques ont révélé que, pour certaines tâches en TALN, telles que la détection de spams, la détection de fausses informations ou l'analyse des sentiments, l'apprentissage d'une représentation compressée peut être inutile. Pour illustrer ce point et à titre de preuve de concept, nous avons développé deux techniques d'explication contrefactuelle transparentes qui surpassent les méthodes opaques. Cela s'explique en grande partie par le fait que les approches opaques génèrent souvent des explications contrefactuelles non intuitives, c'est-à-dire des contre-exemples qui ne ressemblent en rien au texte cible. Cette démarche va à l'encontre non seulement de la nature des explications contrefactuelles, mais soulève également des questions sur le véritable niveau de transparence atteint lorsqu'on explique une boîte noire avec une autre boîte noire.

Ainsi, les contributions clés de ce document sont les suivantes :

1) la proposition d'un spectre évaluant la complexité des explications contrefactuelles, offrant une perspective nuancée sur ces méthodes ;

2) une étude comparative de différentes méthodes contrefactuelles représentant chacune une partie du spectre.

Le document est structuré comme suit. La section 2 définit les méthodes opaques et transparentes. Ensuite, la section 3 examine les méthodes existantes d'explication contrefactuelle. La section 4 présente deux nouvelles méthodes transparentes, que nous analysons ensuite à la lumière du spectre des techniques transparentes et opaques existantes (section 5). Nous détaillons ensuite le protocole expérimental de notre étude comparative dans la section 6. Les résultats de nos expérimentations sont présentés dans la section 7. La section 8 discute de nos conclusions et conclut le document.

2. Méthodes transparentes *vs* méthodes opaques

Dans l'introduction, nous avons catégorisé les techniques d'explication contrefactuelle comme étant soit opaques soit transparentes. Nous définissons maintenant ces notions de manière formelle.

2.1. *Méthodes transparentes*

Implicitement, les méthodes d'explication contrefactuelle transparentes modélisent un texte $x \in X$ de longueur d avec des mots d'un vocabulaire Σ, sous forme d'une matrice binaire creuse de dimension $|\Sigma| \times d$. Ici, $x_{ij} = 1$ signifie que le i-ème mot du vocabulaire Σ apparaît à la j-ème position dans x. Un texte perturbé z est alors obtenu comme une perturbation additive z :

$$z = X + \epsilon, \quad \text{avec} \quad z_{ij} = max(0, min(1, x_{ij} + \epsilon_{ij})),$$

où ϵ est une matrice de bruit telle que ϵ_{ij} est restreinte à trois valeurs : -1 pour supprimer le mot $i \in [1, \ldots, |\Sigma|]$ à la position $j \in [1, \ldots, d]$, 0 pour ne rien faire, et 1 pour ajouter le mot i à la position j. L'opération de découpage $max(0, min(1, \cdot))$ garantit que z est également une matrice binaire.

2.2. *Méthodes opaques*

Les méthodes opaques génèrent des explications contrefactuelles candidates z' en ajoutant du bruit à la représentation du texte cible $x \in X$ dans un espace latent. Si nous désignons une telle représentation par $g(x)$, cela s'exprime comme $z' = g^{-1}(g(x) + \epsilon)$, où $g : X \to \mathbb{R}^{d'}$ est une fonction de transformation dans un espace latent en $\mathbb{R}^{d'}$ (pour un hyperparamètre d' donné), et $\epsilon \in \mathbb{R}^{d'}$ est un vecteur de bruit. Les méthodes opaques doivent également définir la fonction inverse g^{-1} qui mappe un vecteur de nombres réels en un texte.

3. État de l'art

Les méthodes d'explication contrefactuelle génèrent des explications pour les algorithmes d'apprentissage automatique de boîtes noires en fournissant des exemples ressemblant à une instance cible mais conduisant à une prédiction différente par la boîte noire (Wachter *et al.*, 2018). Ces explications transmettent les changements minimaux dans le document en entrée qui modifieraient la prédiction d'un classificateur. Les sciences sociales (Miller, 2019) ont montré que les explications humaines sont contrastives, et Wachter *et al.* (2018) ont illustré l'utilité des instances contrefactuelles en droit informatique. En ce qui concerne les tâches de TALN, une bonne explication contrefactuelle doit être fluide (Wu *et al.*, 2021), c'est-à-dire qu'elle doit se lire comme quelque chose qu'une personne pourrait dire, et parcimonieuse (Verma *et al.*, 2020), c'est-à-dire qu'elle doit ressembler étroitement à l'instance cible.

Les approches contrefactuelles ont gagné en popularité au cours des dernières années. Comme l'illustrent les revues de la littérature, entre celle de Bodria *et al.* (2023) et celle de Guidotti (2022), environ 50 méthodes contrefactuelles supplémentaires sont apparues en l'espace d'un an. Malgré cette vague d'intérêt pour les explications contrefactuelles, leur étude pour les applications de TALN reste peu développée (Ross *et al.*, 2021). Dans ce qui suit, nous détaillons les méthodes d'explication contrefactuelle existantes pour les données textuelles le long d'un spectre qui va des approches transparentes aux approches opaques.

Approches transparentes. Étant donné un classificateur d'apprentissage automatique et un texte cible (également appelé document), les techniques transparentes génèrent des explications contrefactuelles dans un espace binaire. Chaque dimension représente la présence (1) ou l'absence (0) d'un mot issu d'un vocabulaire donné. Ainsi, pour perturber un texte, ces méthodes activent et désactivent les 0 et les 1, où les 0 reviennent à ajouter, supprimer ou remplacer des mots jusqu'à ce que le classificateur fournisse une réponse différente. Cette approche a été initialement proposée par Martens et Provost (2014) qui ont introduit Search for Explanations for Document Classification (SEDC), une méthode qui supprime les mots pour lesquels le classificateur présente la plus grande *sensibilité*. Il s'agit des mots qui influencent le plus la prédiction du classificateur. De manière similaire, les méthodes d'explication basées sur l'attribution de caractéristiques telles que LIME (Ribeiro *et al.*, 2016) et SHAP (Lundberg et Lee, 2017), les deux méthodes d'explication les plus populaires (Jacovi, 2023), masquent aléatoirement des mots du texte cible. Plus récemment, Ross *et al.* (2021) ont développé Minimal Contrastive Editing (MICE), une méthode qui utilise un Text-To-Text Transfer *Transformer* pour remplir les phrases masquées. Yang *et al.* (2020) ont présenté Plausible Counterfactual Instances Generation (PCIG), qui génère des contre-exemples grammaticalement plausibles en modifiant des mots à l'aide de lexiques sélectionnés manuellement dans le domaine économique. Étant donné que ces méthodes sont adaptées à des tâches spécifiques ou nécessitent une sélection manuelle, nous avons exclu ces méthodes de nos expériences.

Méthodes opaques. Nous définissons les approches opaques comme celles qui perturbent le texte d'entrée dans un espace latent en $\mathbb{R}^n$. Des méthodes telles que Decision Boundary (Hase et Bansal, 2020), xSPELLS (Lampridis *et al.*, 2022) ou CounterfactualGAN (Robeer *et al.*, 2021) opèrent en trois phases. Tout d'abord, elles intègrent l'instance cible dans un espace latent, par exemple, à l'aide d'un AutoEncodeur Variationel (VAE) dans le cas de xSPELLS, et d'un modèle de Réseau Générateur Antagoniste Conditionnel (CGAN) pour CounterfactualGAN. Ensuite, tant que la frontière de décision du classificateur n'est pas franchie, ces méthodes perturbent la représentation latente de la phrase cible. Cette perturbation se fait par l'ajout d'un bruit gaussien dans le cas de xSPELLS, tandis que CounterfactualGAN fait appel à un CGAN. Enfin, une étape de décodage génère des phrases à partir de la représentation latente des documents perturbés.

Il existe également des méthodes telles que Polyjuice (Wu *et al.*, 2021), Generate Your Counterfactuals (GYC) (Madaan *et al.*, 2021) et Tailor (Ross *et al.*, 2022) qui perturbent des documents textuels dans un espace latent, comme un modèle de langage masqué et un *Transformer*, mais qui peuvent être instruites pour changer des aspects linguistiques particuliers du texte cible, tels que la localité ou le temps grammatical. De telles méthodes ne sont pas spécialement conçues pour calculer des explications contrefactuelles, mais elles sont plutôt conçues pour de multiples applications telles que l'augmentation de données.

Contrairement aux méthodes de perturbation basées uniquement sur les mots, les représentations latentes préservent bien la « proximité sémantique » pour de petites perturbations. Cependant, ces méthodes ne sont pas exemptes de pièges. Tout d'abord, des méthodes telles que xSPELLS et CounterfactualGAN sont considérées comme opaques car un espace latent n'est pas compréhensible par les humains (Shen *et al.*, 2020). Par conséquent, il existe des méthodes qui génèrent des explications pour l'espace latent (Li *et al.*, 2021). Ainsi, nous nous interrogeons sur le bien-fondé de l'utilisation de mécanismes non directement compréhensibles par les humains pour leur expliquer des classificateurs complexes. De plus, les approches existantes basées sur les mécanismes latents ne semblent pas optimisées pour des explications contrefactuelles parcimonieuses, comme nous le prouvons par des résultats expérimentaux montrant qu'une légère modification dans un espace latent peut entraîner une modification significative dans l'espace d'origine.

4. Méthodes d'explication contrefactuelle pour les données textuelles

Avant de développer notre étude, nous présentons deux nouvelles techniques d'explication contrefactuelle visant à enrichir le terrain entre les approches entièrement opaques et celles entièrement transparentes. Ces méthodes sont appelées *Growing Language* et *Growing Net*, et toutes deux reposent sur un processus itératif qui remplace des mots au sein d'un texte cible $x = (x_1, \ldots, x_d) \in X$ ($x_i \in \Sigma$ représentant des mots d'un vocabulaire Σ) jusqu'à ce que la classe prédite par un classificateur

Algorithme 1 Exploration

Entrée: une instance cible $x = (x_1, \ldots, x_d) \in X$,
un classifieur boîte noire $f : X \to Y$,
MOTSSIM$(\cdot, POS(\cdot)) \to$ une fonction qui retourne les mots similaires à un mot en entrée ;
Hyperparamètres : $n = 2000$
Résultat: une ou plusieurs instances contrefactuelles

1: Initialiser $W = (W_1, \ldots, W_d)$, ensembles de mots candidats
2: **pour** $i \leftarrow 1$ **a** d **faire**
3: $\quad W_i \leftarrow$ MOTSSIM$(x_i, POS(x_i))$
4: **fin pour**
5: Initialiser $Z = (z_1, \ldots, z_n)$ comme n copies de x
6: Initialiser $C \leftarrow \emptyset$; $r \leftarrow 0$
7: **tant que** $r < d \vee C = \emptyset$ **faire**
8: $\quad r \leftarrow r + 1$
9: $\quad$ **pour** $j \leftarrow 1$ **a** n **faire** ▷ Pour chaque copie de x
10: $\quad\quad$ **pour** $l \leftarrow 1$ **a** r **faire**
11: $\quad\quad\quad k \leftarrow$ *aléatoire*$(0, d)$ ▷ $k : z_j^k = x_k$
12: $\quad\quad\quad z_j^k \leftarrow$ mot aléatoire de W_k
13: $\quad\quad$ **fin pour**
14: $\quad\quad$ **si** $f(x) \neq f(z_j)$ **alors**
15: $\quad\quad\quad C \leftarrow C \cup \{z_j\}$
16: $\quad\quad$ **fin si**
17: $\quad$ **fin pour**
18: **fin tant que**
19: **retourner** C

donné $f : X \to Y$ change. L'objectif d'une telle procédure est de calculer des explications contrefactuelles parcimonieuses avec le moins de mots modifiés possible.

L'algorithme 1 décrit le processus d'exploration itératif utilisé par *Growing Language* et *Growing Net*. Dans la première étape (lignes 1 à 4), les deux approches génèrent d ensembles de remplacements potentiels $W_1, \ldots, W_d$ pour chaque mot x_i dans le document cible x. Ces remplacements doivent avoir la même nature ou étiquette grammaticale que x_i. Le module externe permettant d'obtenir ces remplacements dépend de la méthode, et ces modules sont détaillés ultérieurement. Ensuite, nos méthodes créent de manière itérative des documents artificiels (lignes 7 à 18), illustrés sous forme d'une structure arborescente dans la figure 2. Ces documents sont générés tant que certains mots dans le document original restent non remplacés ($r < d$), ou tant que nous n'avons pas trouvé de contrefactuels ($C = \emptyset$). À chaque itération, l'exploration conserve n copies du texte original (x) sur lesquelles nous remplaçons r mots individuels (x_k) par des mots sélectionnés au hasard dans leurs ensembles respectifs de remplacements potentiels (W_k). La ligne 11 assure que le mot remplacé provient bien de la phrase originale pour effectivement remplacer r mots au lieu d'un

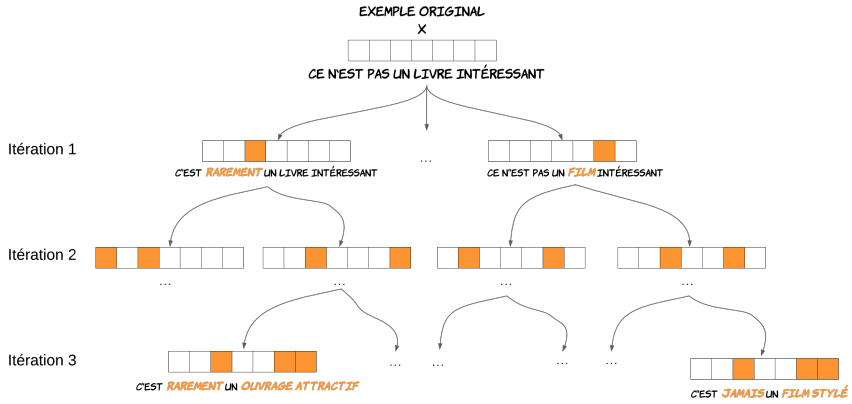

FIGURE 2. *Structure en arborescence de l'algorithme utilisé pour perturber de manière itérative le document cible. À chaque tour, un mot du texte cible est remplacé de manière itérative par un mot de son ensemble de mots de remplacement potentiels. Ainsi, à chaque tour successif, le nombre de mots remplacés pour la génération de documents artificiels augmente.*

mot déjà remplacé. Enfin, les lignes 14 à 16 vérifient si les phrases résultantes sont des instances contrefactuelles.

Prenons l'exemple de la critique cible classée comme négative par un modèle d'analyse de sentiments : *« Ce n'est pas un livre intéressant »* (figure 2). Lors du premier tour, *Growing Language* et *Growing Net* génèrent des documents artificiels en modifiant un seul mot. Les tours suivants impliquent le remplacement de deux mots, et ainsi de suite. Dans ce processus, des contre-exemples sont identifiés, et le plus proche est renvoyé comme explication. Ces méthodes ont pour priorité de produire des contre-exemples proches du document original afin de fournir des explications concises et significatives.

4.1. *Growing Net*

Algorithme 2 Growing Net

Entrée: un texte cible $x = (x_1, \ldots, x_d) \in X$,
Un classifieur boîte noire f ;
1: $C \leftarrow$ *exploration*$(x, f, \text{WN\_MOTSSIM}_{t=1}(\cdot))$
2: **retourner** *argmax*$_{c \in C}$*Wu-P*(c, x)

Growing Net tire parti de la structure riche de WordNet (Fellbaum, 1998) pour construire des ensembles de mots étroitement liés. WordNet est une base de données

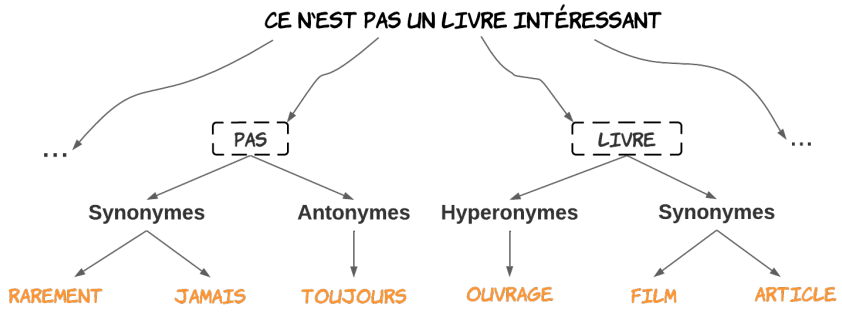

FIGURE 3. *Diagramme représentant les mécanismes de l'approche Growing Net. En exploitant la structure arborescente de WordNet, Growing Net génère des ensembles de mots pouvant remplacer chaque terme du document cible. À travers des itérations successives, les mots du texte cible sont remplacés jusqu'à ce que les contrefactuels soient découverts.*

lexicale et un thésaurus qui organise les mots et leurs significations dans un arbre sémantique de concepts interconnectés. La méthode est décrite dans l'algorithme 2 et utilise le module WN_MOTSSIM. Dans la phase d'exploration, *Growing Net* utilise WN_MOTSSIMt pour trouver des mots à une distance d'au plus t dans la hiérarchie WordNet parmi les synonymes, les antonymes, les hyponymes et les hyperonymes pour un mot donné x_i à remplacer. Ce processus est illustré dans la figure 3. Dans nos expériences, nous avons fixé $t = 1$ car cette valeur donne déjà de bons résultats – des valeurs plus élevées entraîneraient des temps d'exécution plus longs. L'exploration renvoie un ensemble de contrefactuels, parmi lesquels *Growing Net* sélectionne celui avec la plus grande similarité de Wu-Palmer (Wu-P) (Wei et Ngo, 2007) comme explication finale. Ce score de similarité pour le texte s'appuie sur WordNet et prend en compte la parenté des concepts dans la phrase, par exemple via la longueur du chemin jusqu'à leur ancêtre le plus commun dans la hiérarchie.

4.2. *Growing Language*

Growing Language exploite la puissance des grands modèles de langue pour restreindre l'espace des remplacements potentiels de mots via le module LM_MOTSSIMθ (algorithme 3). Les grands modèles de langue sont de puissants systèmes d'intelligence artificielle de traitement du langage naturel et sont utilisés dans ce contexte pour incorporer les mots dans une représentation numérique, permettant ainsi de mesurer la similarité entre les mots dans un espace latent. Étant donné un mot x_i à remplacer, LM_MOTSSIMθ incorpore le mot dans l'espace latent d'un modèle de langue, comme illustré dans la figure 4. Ensuite, LM_MOTSSIM$_\theta$ récupère les mots dont la

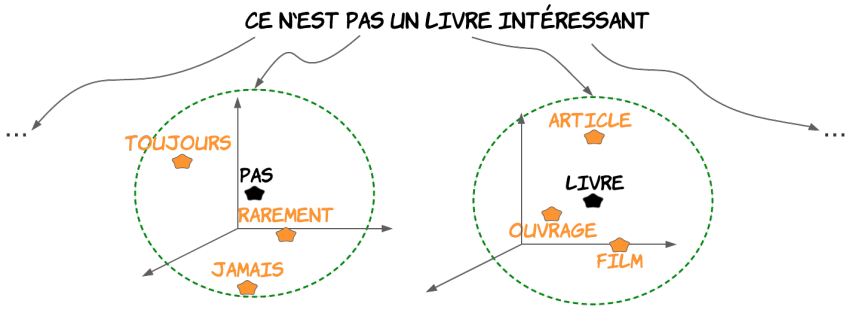

FIGURE 4. *Schéma du fonctionnement de la méthode* Growing Language*. Les mots présents dans le texte cible sont transformés en une représentation latente grâce à l'utilisation d'un modèle de langue de grande envergure. Dans cet espace latent, les mots ayant des similitudes deviennent des remplacements potentiels pour la génération de documents artificiels. À chaque itération, le nombre de mots remplacés dans le document augmente.*

Algorithme 3 Growing Language

Entrée: un texte cible $x = (x_1, \ldots, x_d) \in X$,
Un classifieur boîte noire f ;
Hyperparamètres : $\tau = 0.02$; $\theta = 0.9$; $\theta_{min} = 0.4$;

$C \leftarrow \emptyset$
tant que $\theta > \theta_{min} \wedge C = \emptyset$ **faire**
    $C \leftarrow C \cup \textit{exploration}(x, f, \text{LM\_MOTSSIM}_\theta(\cdot))$
    $\theta \leftarrow \theta - \tau$
fin tant que
retourner $\textit{argmin}_{c \in C} ||x - c||_0$

représentation latente est à une distance d'au plus θ. Dans nos expériences, nous avons initialement fixé ce seuil à 0,8 sur une échelle de 0 à 1. Si, pour un θ donné, *Growing Language* ne parvient pas à trouver des instances contrefactuelles, le seuil de distance est relâché, c'est-à-dire réduit de τ (fixé à 0,02 dans nos expériences), afin que la routine d'exploration considère plus de mots. Si plusieurs contrefactuels sont trouvés, *Growing Language* sélectionne celui avec la plus petite distance par rapport au document original (selon le modèle de langage). Pour nos expériences, nous avons utilisé le modèle en_core_web_md de la bibliothèque Spacy (Honnibal et Montani, 2017), mais tout modèle de langage capable d'incorporer des mots et d'offrir des distances entre les mots pourrait être utilisé dans ce contexte.

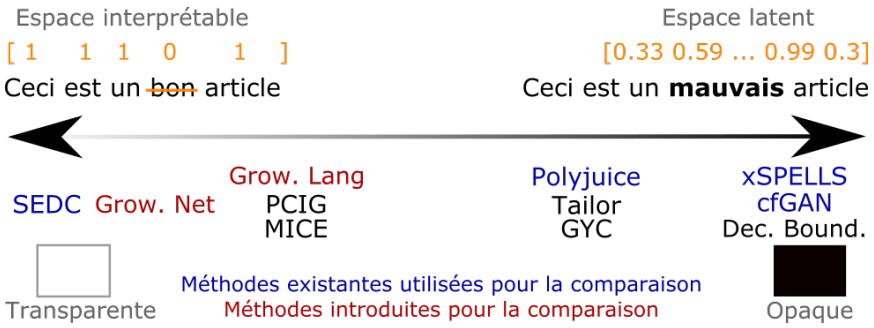

FIGURE 5. *Spectre des techniques d'explication contrefactuelle allant des méthodes les plus transparentes à gauche (par exemple, SEDC) aux méthodes les plus opaques telles que xSPELLS, en passant par nos méthodes en rouge. Les méthodes transparentes perturbent les documents dans un espace binaire ; celles opaques le font dans un espace latent.*

5. Échelle d'interprétabilité

Nous soulignons que la catégorisation « transparente » ou « opaque » d'une méthode d'explication contrefactuelle définit les deux extrémités d'un continuum, que nous représentons dans la figure 5. Cette échelle s'étend des méthodes les plus transparentes à gauche aux méthodes les plus opaques à droite. On distingue deux catégories de méthodes transparentes : les méthodes **complètement transparentes** et les méthodes **partiellement transparentes**. Dans la première catégorie, les individus peuvent comprendre pourquoi l'ajout de bruit à un mot produit un résultat spécifique, comme le remplacement d'un mot par son antonyme. En revanche, pour les méthodes partiellement transparentes, la compréhension de l'utilisation d'un bruit ϵ_i peut rester partiellement obscure. Il existe également deux catégories de méthodes opaques : les méthodes **partiellement opaques** et les méthodes **complètement opaques**. Dans la première catégorie, il est possible de comprendre partiellement l'objectif de la perturbation dans l'espace latent, c'est-à-dire que l'objectif de ϵ est compréhensible, notamment avec l'aide de codes de contrôle. À l'opposé, il est difficile, voire impossible, de comprendre l'objectif de la perturbation dans l'espace latent des méthodes totalement opaques, c'est-à-dire que ϵ est insaisissable. Nous détaillons les différentes régions de cette échelle ci-dessous.

Transparence complète. À l'extrémité gauche de l'échelle, nous trouvons la méthode SEDC (Martens et Provost, 2014), qui perturbe les instances de texte en masquant uniquement les mots très sensibles dans le texte. Nous plaçons *Growing Net* à droite de SEDC, car il va au-delà d'un simple masquage de mots. Au lieu de cela, il tire parti des connaissances et de la structure en arborescence de WordNet pour sélectionner des substitutions de mots de manière plus judicieuse.

Transparence partielle. Des méthodes comme PCIG (Yang *et al.*, 2020), MICE (Ross *et al.*, 2021) et *Growing Language* sont considérées comme plus opaques que *Growing Net*, car elles utilisent un espace latent pour identifier des substitutions de mots sémantiquement proches. Cependant, nous les considérons transparentes car leurs explications générées préservent la structure du document tout en révélant quels mots devraient être remplacés et par quels autres mots.

Opacité partielle. Polyjuice, Tailor et GYC relèvent de la catégorie des méthodes partiellement opaques, car elles s'appuient sur des codes de contrôle pour perturber le document cible. Ces codes agissent comme des instructions spécifiques qui adaptent la perturbation du texte cible afin qu'elle soit conforme à une tâche spécifique, telle que la traduction, le résumé ou la modification du temps grammatical d'un texte. Bien que ces modifications se produisent dans un espace latent, l'inclusion de codes de contrôle fournit un certain niveau de clarté sur la manière dont une modification influence la prédiction du modèle.

Opacité complète. À l'extrême droite de l'échelle d'interprétabilité, nous rencontrons des approches totalement opaques telles que Decision Boundary, xSPELLS et CounterfactualGAN. En effet, ces méthodes perturbent les instances dans un espace latent, rendant difficile pour les utilisateurs de discerner le processus sous-jacent de génération des contrefactuels.

Cette échelle de complexité offre des informations précieuses sur la transparence et sur l'opacité des méthodes d'explication contrefactuelle, permettant une compréhension plus nuancée de leurs capacités.

6. Informations expérimentales

Après avoir introduit l'échelle des méthodes d'explication contrefactuelle le long de l'axe de l'interprétabilité, nous décrivons maintenant le protocole expérimental conçu pour évaluer ces méthodes. Le code des méthodes étudiées, les ensembles de données et les résultats expérimentaux sont disponibles sur GitHub[1], ce qui est essentiel pour reproduire nos expériences et pour comprendre la méthodologie. Dans cette section, nous fournissons donc un compte rendu détaillé du processus de génération contrefactuelle, des ensembles de données utilisés, des classificateurs sélectionnés pour l'explication et des métriques appliquées dans nos expériences.

6.1. *Génération contrefactuelle*

Nous avons sélectionné un ensemble de méthodes agnostiques au domaine, représentatives de toutes les régions de l'échelle représentée dans la figure 5. Celles-ci comprennent SEDC et *Growing Net* parmi les méthodes totalement transparentes,

1. `https://github.com/j2launay/ebbwbb`

Growing Language parmi les méthodes partiellement transparentes, Polyjuice parmi les méthodes partiellement opaques, et xSPELLS et cfGAN parmi le groupe des méthodes totalement opaques.

Il est important de noter que nous avons exclu de notre analyse ultérieure les méthodes PCIG et MICE, chacune ayant fait l'objet d'une exclusion motivée par des considérations spécifiques. Dans le cas de PCIG, cette méthode se base sur des règles spécifiques au domaine de l'économie, ce qui limite sa pertinence pour nos ensembles de données variés. En ce qui concerne MICE, elle fait appel à des modèles de *Transformer* pour identifier les remplacements de mots pertinents sur le plan sémantique, ce qui est coûteux en termes de calcul selon les auteurs. Cette complexité va à l'encontre de notre objectif de privilégier des méthodes transparentes plus simples.

De plus, nous avons délibérément écarté des méthodes adversariales de notre analyse telles que Morris *et al.* (2020). Ces méthodes sont conçues pour induire des erreurs dans les prédictions des modèles plutôt que dans un but explicatif. Ainsi, nous les avons exclues pour souligner la différence fondamentale de leur objectif par rapport aux méthodes d'explication contrefactuelle. Nous avons privilégié des approches axées sur la compréhension des décisions du modèle plutôt que sur la manipulation de ses prédictions. De manière analogue, nous avons retiré Linguistically-Informed Transformation (LIT), introduite par Li *et al.* (2020), une méthode qui vise à générer automatiquement des jeux de contrastes. LIT vise à produire des documents en dehors de la distribution des données, les rendant ainsi non réalistes et éloignés de notre objectif d'explications fidèles et pertinentes.

Les informations détaillées concernant l'implémentation, les versions et les hyperparamètres de chaque méthode d'explication contrefactuelle utilisée dans nos expériences sont disponibles en Annexe A.

6.2. *Jeux de données*

Pour nos expériences, nous avons utilisé trois jeux de données conçus pour trois applications différentes : (a) la détection de spams dans les messages, (b) l'analyse de sentiments, et (c) la détection de fausses informations dans les titres d'articles de journaux. Chacun de ces ensembles de données comporte deux classes cibles et contient entre 4 000 et 10 660 documents textuels. Le nombre moyen de mots dans chaque document se situe entre 11,8 et 20,8, comme indiqué dans le tableau 1.

En ce qui concerne le jeu de données de détection de fausses informations, nous l'avons construit en utilisant de vrais titres d'articles de journaux provenant d'un jeu de données[2] avec des titres fabriqués à partir d'un jeu de données de fausses informations[3]. Ce jeu de données combiné est disponible publiquement sur notre

2. `https://www.kaggle.com/datasets/rmisra/news-category-dataset`
3. `https://www.kaggle.com/competitions/fake-news/overview`

GitHub[4]. Quant aux jeux de données de polarité (Pang et Lee, 2005) et de détection de spams (Gómez Hidalgo *et al.*, 2006), nous les avons obtenus à partir de Kaggle. Nous avons divisé chaque ensemble de données en ensembles d'entraînement et de test en utilisant la fonction de la bibliothèque scikit-learn : $train\_test\_split$ avec une taille de test de 30 % et une graine aléatoire de 1.

Nom	Nombre de mots			Instances	Modèle		
	Total	Moyenne	σ		Réseau neur.	Forêt aléat.	BERT
Fake	19 419	11,8	3,2	4 025	84 %	84 %	91 %
Polarity	11 646	20,8	9,3	10 660	72 %	67 %	82 %
Spam	15 587	18,5	10,6	8 559	100 %	100 %	100 %

TABLEAU 1. *Informations concernant les jeux de données expérimentaux. Les trois colonnes sous « Nombre de mots » représentent respectivement (a) le nombre total de mots distincts dans l'ensemble du jeu de données, (b) le nombre moyen de mots par phrase, et (c) l'écart type. La colonne « Instances » indique le nombre de documents textuels par jeu de données. Les dernières colonnes montrent la précision moyenne des trois classificateurs pour chaque jeu de données.*

6.3. *Classificateurs boîte noire*

Notre évaluation utilise deux classificateurs boîte noire distincts implémentés à l'aide de la bibliothèque scikit-learn et déjà employés (Lampridis *et al.*, 2022). Ces boîtes noires sont (i) une forêt aléatoire (RF) composée de 500 estimateurs d'arbres, (ii) un perceptron multicouche (MLP) avec autant de neurones qu'il y a de mots dans le jeu de données, et (iii) un classifieur basé sur DistillBERT[5]. Pour la RF et le MLP, nous avons utilisé à la fois les vectoriseurs *matrice de comptes d'occurrences* et *tf-idf* pour convertir le texte en entrées appropriées pour les modèles.

Nous avons entraîné ces classificateurs sur 70 % du jeu de données, et leur précision a été testée sur les 30 % restants. Nous avons également sélectionné l'instance cible à expliquer dans ce jeu de test. Sur l'ensemble des jeux de données, la précision moyenne de ces trois classificateurs varie de 67 % à 100 %. Les résultats détaillés sont présentés dans le tableau 1.

Toutes les expériences ont été réalisées sur un serveur équipé d'un processeur Intel(R) Xeon(R) Gold 5220 CPU (2,20 GHz, 18 cœurs, 24 MB de cache L3) et de 96 GB de mémoire vive (DDR4).

4. `https://github.com/j2launay/ebbwbb`
5. `https://is.gd/zljjJN`

7. Résultats

Nous présentons maintenant les résultats de notre évaluation, organisés en quatre séries d'expériences catégorisées selon deux aspects. Tout d'abord, nous évaluons la qualité des explications contrefactuelles produites en fonction de deux critères essentiels : (i) la **minimalité**, et (ii) la **plausibilité**. Ensuite, nous évaluons les méthodes elles-mêmes en termes de (iii) **capacité de changement de classification**, et de (iv) **temps d'exécution**. Pour chaque méthode évaluée et chaque classificateur boîte noire, nous avons généré des explications contrefactuelles pour 100 textes cibles extraits des ensembles de tests de nos jeux de données.

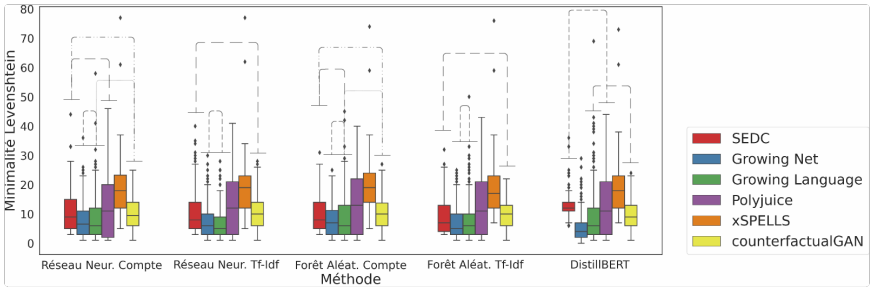

FIGURE 6. *Minimalité mesurée comme la distance d'édition de Levenshtein entre le contrefactuel le plus proche et le texte cible (↓ meilleur). Les barres en pointillé représentent les paires de méthodes qui présentent des différences de minimalité* ***non*** *statistiquement significatives.*

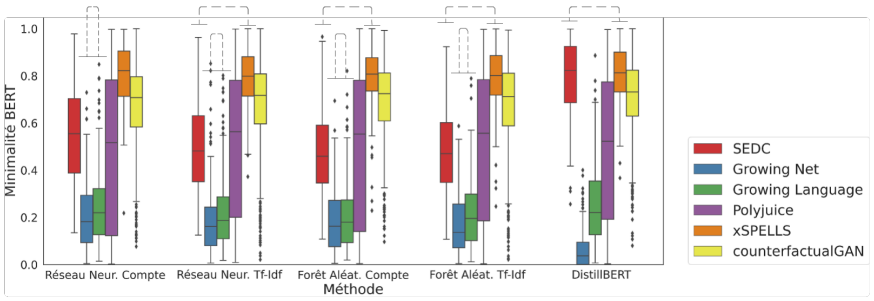

FIGURE 7. *Minimalité mesurée comme la distance d'intégration de Sentence-BERT entre le contrefactuel le plus proche et le texte cible (↓ meilleur). Les barres en pointillé représentent les paires de méthodes qui présentent des différences de minimalité* ***non*** *statistiquement significatives selon l'analyse* post hoc.

7.1. *Qualité des contrefactuels*

Une explication contrefactuelle textuelle de haute qualité nous indique quels sont les parties ou les aspects les plus sensibles de la phrase cible qui, autrement modifiés, conduiraient à un résultat de classification différent. Comme nous l'avons mentionné dans la section 3, il en découle alors qu'une telle explication doit (i) entraîner des changements minimaux par rapport à la phrase cible (changements parcimonieux), et (ii) être linguistiquement plausible, c'est-à-dire ressembler à quelque chose qu'une personne écrirait ou dirait naturellement (Guidotti, 2022).

7.1.1. *La minimalité*

Nous quantifions le critère de minimalité en mesurant la distance entre le contrefactuel et la phrase cible. Les figures 6 et 7 présentent les résultats de nos évaluations de minimalité, en considérant à la fois la distance de Levenshtein, une mesure du nombre de modifications nécessaires pour transformer une chaîne de texte en une autre, et la similarité cosinus dans l'espace d'intégration du modèle BERT-Sentence (Reimers et Gurevych, 2019). Cette approche double assure une évaluation exhaustive, tenant compte à la fois de la similarité lexicale et des caractéristiques latentes, y compris des aspects de style.

Nos résultats révèlent que les méthodes positionnées dans la zone intermédiaire, en particulier *Growing Net*, ont donné des résultats favorables par rapport aux approches opaques, tant en termes du nombre de mots modifiés que de comparaison sémantique. Il est à noter que xSPELLS a introduit les modifications les plus significatives dans le texte original, contredisant ainsi l'un des principaux critères fonctionnels d'une explication contrefactuelle (Wachter *et al.*, 2018). De même, nous observons une forte variance dans la minimalité des contrefactuels générés par Polyjuice, indiquant que certains contrefactuels étaient notablement éloignés de leurs instances cibles correspondantes. Bien que ces méthodes aient introduit des perturbations mineures dans le texte original, ces modifications se sont produites dans un espace latent. Rien ne garantit cependant que ces ajustements mineurs se traduisent par des modifications visuellement subtiles de la phrase cible lorsque la phrase résultante est ramenée à l'espace d'origine. À titre d'exemple, considérons le texte cible « *This is one of Polanski's best films.* » du jeu de données sur la polarité. Pour le classifieur DistillBERT, CounterfactualGAN retourne le contrefactuel « *this is one of* ***shot kingdom intelligence's all*** », qui semble complètement sans rapport avec le texte cible. En revanche, la méthode transparente SEDC produit le contrefactuel « *This is one of* ***MASK MASK MASK*** », tandis que *Growing Language* produit « *This is one of Polanski's* ***worst*** *films.* » D'autres exemples de contrefactuels générés par chaque méthode, sont présentés en Annexe B.

Nous avons ainsi noté que lorsque la complexité du classifieur augmente, les explications contrefactuelles générées par SEDC s'éloignent davantage du texte original. Ensuite, nous observons des variations mineures dépendantes du vectoriseur utilisé par les classifieurs (*matrice de comptes d'occurrences* ou *tf-idf*). Nous justifions ainsi le choix du vectoriseur *tf-idf* en tant que référence dans notre analyse, car il présente

des différences de minimalité significatives, comme représentées graphiquement dans les figures 6 à 8 par un nombre plus petit de barres en pointillé indiquant les différences non significatives. Pour la phase ultérieure de l'évaluation, nous présentons exclusivement les résultats obtenus avec le vectoriseur *tf-idf*.

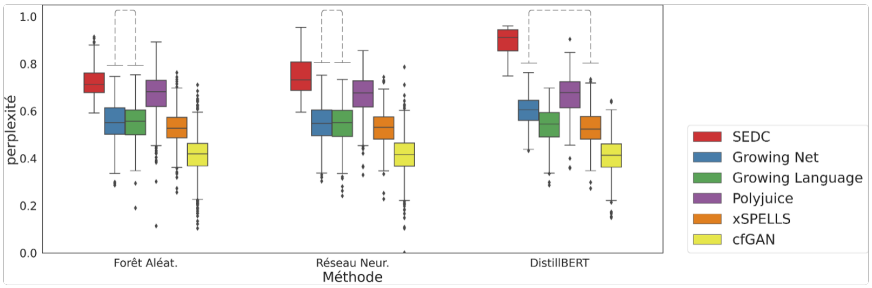

FIGURE 8. *Perplexité comme l'erreur quadratique moyenne d'un modèle GPT sur les contrefactuels générés (↓ meilleur). Les barres en pointillé indiquent les paires de méthodes pour lesquelles les différences de perplexité* ***ne sont pas*** *statistiquement significatives.*

7.1.2. *La vraisemblance linguistique*

Alors que la plausibilité linguistique est généralement évaluée à travers des études utilisateurs (Madaan *et al.*, 2021 ; Wu *et al.*, 2021), nous l'évaluons ici en suivant les techniques de Ross *et al*. (Ross *et al.*, 2021 ; Ross *et al.*, 2022). Ainsi, nous utilisons des scores de perplexité basés sur un modèle linguistique GPT (Brown *et al.*, 2020), en calculant l'erreur quadratique moyenne (MSE) lors de la prédiction du mot suivant dans le contrefactuel à partir des mots précédents. La figure 8 présente la plausibilité des contrefactuels. Pour améliorer la comparabilité, nous avons normalisé les scores de perplexité en fonction de la perplexité maximale observée sur l'ensemble des contrefactuels, où des scores plus bas indiquent une plausibilité plus élevée. Nous avons effectué une analyse *post hoc* en utilisant un test *t* avec une correction Bonferroni pour évaluer les différences statistiques entre les catégories des variables prédictives. Cette analyse a révélé que seuls *Growing Language*, *Growing Net*, et xSPELLS ne présentent pas de différence statistique significative après l'ajustement.

En particulier, SEDC et Polyjuice ont généré des textes avec la plus faible plausibilité, ce qui est attendu puisque SEDC masque des mots, conduisant parfois à des phrases dépourvues de sens. En revanche, CounterfactualGAN a démontré la plus haute plausibilité, tandis que *Growing Net* et *Growing Language* ont obtenu des scores de perplexité similaires à ceux de xSPELLS.

Jeu de données	Fausses informations			Détection de spams			Polarité		
Modèles	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
counterfactualGAN	0.18	0.12	0.09	0.14	0.05	0.03	0.50	0.50	0.48

TABLEAU 2. *Moyenne des changements de classification par jeu de données et boîte noire des six méthodes contrefactuelles (↑ meilleur)*

7.2. *Qualité des méthodes*

Nous comparons maintenant la qualité des méthodes d'explication contrefactuelle elles-mêmes en fonction de deux critères : (iii) la capacité de changement de classification, qui mesure la fréquence à laquelle une méthode parvient à produire avec succès un contrefactuel, c'est-à-dire une instance classée différemment par le modèle, et (iv) le temps d'exécution, mesuré comme le temps nécessaire à chaque méthode pour générer une explication contrefactuelle.

7.2.1. *La capacité de changement de classification*

Le tableau 2 donne un aperçu des résultats du taux de changement de classification, indiquant la capacité des méthodes à trouver un contrefactuel pour un texte donné. Il est important de noter que, en raison du faible nombre de mots par jeu de données (entre 11,8 et 20,8), il est plus difficile pour les méthodes de trouver un contrefactuel. En effet, le nombre de mots à remplacer est plus restreint. Nous observons ainsi qu'à l'exception du jeu de données sur la détection de spams, les méthodes transparentes atteignent le taux de changement de classification le plus élevé. Cette constatation souligne l'efficacité de remplacer des mots par leurs antonymes comme moyen de découvrir des contrefactuels. De plus, xSPELLS présente des performances solides pour le jeu de données sur la détection de spams et affiche des taux de changement de classification similaires aux méthodes transparentes sur la détection de la polarité.

Il est crucial de souligner que le jeu de données de détection de spams présente une difficulté accrue en raison de la présence de nombreux caractères spéciaux et de l'utilisation de la nature informelle du langage SMS. Cette complexité rend la génération de contrefactuels plus ardue. En outre, nous notons que *Growing Net* et *Growing Language* peuvent être ajustés pour une recherche plus exhaustive en modifiant leurs paramètres, par exemple, en réduisant le seuil de similarité minimale (θ_{min} dans l'algorithme 3) ou en explorant davantage la structure arborescente de WordNet

(augmentation de t dans l'algorithme 2). Bien que de tels ajustements puissent améliorer le taux de basculement d'étiquette, cela peut néanmoins entraîner des temps d'exécution plus longs.

Jeu de données	Méthode	Réseau neur.	Forêt aléat.	BERT
Fausses informations	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)
	counterfactualGAN	1 (0)	1 (0)	1 (0)
	xSPELLS	84 (6)	86 (7)	16 (1)
Détection de spams	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)
	counterfactualGAN	1 (0)	1 (0)	1 (0)
	xSPELLS	219 (17)	198 (16)	22 (1)
Détection de sentiments	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)
	counterfactualGAN	1 (0)	1 (0)	1 (0)
	xSPELLS	136 (19)	115 (11)	24 (2)

TABLEAU 3. *Durée moyenne en secondes des méthodes contrefactuelles étudiées pour générer un contrefactuel (et écart-type)*

7.2.2. *Temps d'exécution*

Enfin, nos résultats concernant le temps d'exécution se trouvent dans le tableau 3. Le tableau détaille la moyenne et l'écart type du temps d'exécution pour chaque méthode d'explication contrefactuelle à travers les jeux de données et les classificateurs. De manière notable, counterfactualGAN et *Growing Net* se sont révélées être les méthodes les plus rapides pour générer des contrefactuels. Cependant, il est important de noter que counterfactualGAN nécessite l'entraînement du *Generative Adversarial Network* (GAN) sur chaque jeu de données spécifique, un processus qui nécessite un temps d'entraînement significatif. Le temps nécessaire pour l'optimisation varie, allant de 4 300 secondes pour la détection de titres de fausses informations à 6 755 secondes pour la détection de spams.

De plus, nous observons que xSPELLS et *Growing Language* présentent les performances les plus lentes en termes de temps d'exécution. *Growing Language*, par exemple, nécessite environ 60 secondes pour générer un seul contrefactuel, tandis que xSPELLS affiche des temps d'exécution allant de 16 secondes pour la détection de fausses informations à 219 secondes pour la détection de spams. Ces résultats révèlent que, contrairement aux méthodes opaques telles que xSPELLS, les approches

transparentes comme *Growing Net* sont suffisamment rapides pour une explicabilité en temps réel.

8. Discussion et conclusion

Notre évaluation fournit des perspectives précieuses sur le paysage des explications contrefactuelles pour les tâches de traitement automatique du langage naturel (TALN). L'une des découvertes les plus frappantes est que la complexité, souvent associée à l'utilisation de réseaux neuronaux et d'espaces latents, n'est pas nécessairement égale à une performance supérieure dans ce contexte. De manière surprenante, nos résultats montrent que des approches plus simples, caractérisées par une stratégie systématique et judicieuse de remplacement de mots, produisent des résultats satisfaisants sur plusieurs dimensions de qualité. Les résultats de notre étude incitent à une réflexion approfondie sur les stratégies optimales pour générer des explications contrefactuelles dans le domaine du TALN. Cela invite les lecteurs à réfléchir aux implications plus larges de nos découvertes et à leurs conséquences pour le développement d'approches transparentes par rapport à l'amélioration des méthodes opaques. Le choix entre ces approches doit être fait judicieusement, en tenant compte des exigences spécifiques et des contraintes de l'application en question.

De plus, nos conclusions soulignent l'importance cruciale de la transparence et de l'interprétabilité en intelligence artificielle (IA) et en apprentissage automatique. À mesure que nous évoluons dans le paysage complexe de modèles d'IA de plus en plus sophistiqués, la nécessité de la transparence, de la responsabilité et de la confiance devient primordiale, surtout dans des applications à enjeux élevés où les décisions humaines sont influencées par les recommandations de l'IA. Le paradoxe de l'explication d'une boîte noire par une autre soulève des questions pertinentes sur l'équilibre entre la complexité du modèle, son interprétabilité et ses performances. Cela remet en question le développement d'approches opaques lorsque des méthodes transparentes suffisent, ou lorsque la transparence est l'un des objectifs dès le départ.

Lorsqu'on se concentre sur les applications de TALN, nos résultats appellent également à une réflexion sur la signification et l'objectif des explications. Si la tâche consiste à comprendre quels aspects d'un texte devraient changer pour obtenir un résultat différent, une explication contrefactuelle qui modifie radicalement chaque mot dans le texte peut ne pas être compréhensible. Au contraire, une explication contrefactuelle basée sur un masquage simple de mots, bien que simple, peut être perçue comme implausible. Cela pourrait entraver l'objectif des explications en tant que moyen de susciter la confiance chez les utilisateurs. Nous nous attendons donc à ce que nos conclusions encouragent le développement de systèmes d'IA plus transparents et interprétables qui favorisent la confiance et la responsabilité à chaque étape des processus de prise de décision pilotés par l'IA, que ce soit pour la prédiction, pour la recommandation ou pour l'explication. Enfin, nous croyons que les enseignements tirés de cet article pourraient être naturellement étendus à d'autres paradigmes d'explication.

Limitations

Notre évaluation s'est concentrée sur trois domaines d'application : l'analyse de sentiments, la détection de fausses informations et la détection de spams. Par conséquent, la généralisation de nos résultats à d'autres tâches de traitement du langage naturel dans des domaines spécialisés ou dans des langues différentes pourrait être limitée. Bien que notre étude mette en avant les approches transparentes, la génération de contre-exemples plausibles repose souvent sur des connaissances externes adaptées au domaine, que ce soit sous la forme de modèles linguistiques ou de graphes de connaissances. La disponibilité de ces ressources peut varier, ce qui peut influencer l'applicabilité de ces méthodes dans différents contextes. Enfin, notre évaluation s'est basée sur des critères et sur des métriques largement utilisés pour les explications contrefactuelles. Des applications spécialisées pourraient prendre en compte des critères supplémentaires, tels que la diversité ou l'applicabilité, pour évaluer de manière exhaustive la performance des méthodes d'explication contrefactuelle.

9. Bibliographie

Bodria F., Giannotti F., Guidotti R., Naretto F., Pedreschi D., Rinzivillo S., « Benchmarking and survey of explanation methods for black box models », *Proc. Data Mining and Knowledge Discovery*, 2023.

Brown T. B., Mann B., Ryder N., Subbiah M., Kaplan J., Dhariwal P., Neelakantan A., Shyam P., Sastry G., Askell A., Agarwal S., Herbert-Voss A., Krueger G., Henighan T., Child R., Ramesh A., Ziegler D. M., Wu J., Winter C., Hesse C., Chen M., Sigler E., Litwin M., Gray S., Chess B., Clark J., Berner C., McCandlish S., Radford A., Sutskever I., Amodei D., « Language Models are Few-Shot Learners », *in* H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (eds), *Proc. NeurIPS*, 2020.

Devlin J., Chang M., Lee K., Toutanova K., « BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding », *in* J. Burstein, C. Doran, T. Solorio (eds), *Proc. NAACL-HLT*, Association for Computational Linguistics, 2019.

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

Gómez Hidalgo J. M., Bringas G. C., Sánz E. P., García F. C., « Content based SMS spam filtering », *Proc. Symposium on Document Engineering*, Association for Computing Machinery, New York, NY, USA, 2006.

Guidotti R., « Counterfactual explanations and how to find them : literature review and benchmarking », *Data Mining and Knowledge Discovery*, 2022.

Gururangan S., Swayamdipta S., Levy O., Schwartz R., Bowman S. R., Smith N. A., « Annotation Artifacts in Natural Language Inference Data », *in* M. A. Walker, H. Ji, A. Stent (eds), *Proc. NAACL-HLT*, Association for Computational Linguistics, 2018.

Hase P., Bansal M., « Evaluating Explainable AI : Which Algorithmic Explanations Help Users Predict Model Behavior ? », *in* D. Jurafsky, J. Chai, N. Schluter, J. R. Tetreault (eds), *Proc. ACL*, Association for Computational Linguistics, 2020.

Honnibal M., Montani I., « spaCy 2 : Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing », 2017.

Jacovi A., « Trends in Explainable AI (XAI) Literature », *CoRR*, 2023.

Lampridis O., State L., Guidotti R., Ruggieri S., « Explaining short text classification with diverse synthetic exemplars and counter-exemplars », *Machine learning*, 2022.

Li C., Shengshuo L., Liu Z., Wu X., Zhou X., Steinert-Threlkeld S., « Linguistically-Informed Transformations (LIT) : A Method for Automatically Generating Contrast Sets », *in* A. Alishahi, Y. Belinkov, G. Chrupala, D. Hupkes, Y. Pinter, H. Sajjad (eds), *Proc. of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP*, Association for Computational Linguistics, 2020.

Li Z., Tao R., Wang J., Li F., Niu H., Yue M., Li B., « Interpreting the Latent Space of GANs via Measuring Decoupling », *IEEE Trans. Artif. Intell.*, 2021.

Liu Y., Ott M., Goyal N., Du J., Joshi M., Chen D., Levy O., Lewis M., Zettlemoyer L., Stoyanov V., « RoBERTa : A Robustly Optimized BERT Pretraining Approach », *CoRR*, 2019.

Lundberg S. M., Lee S., « A Unified Approach to Interpreting Model Predictions », *Proc. NIPS*, 2017.

Madaan N., Padhi I., Panwar N., Saha D., « Generate Your Counterfactuals : Towards Controlled Counterfactual Generation for Text », *Proc. IAAI, The Symposium on Educational Advances in Artificial Intelligence, EAAI*, AAAI Press, 2021.

Martens D., Provost F. J., « Explaining Data-Driven Document Classifications », *MIS Q.*, 2014.

McCoy T., Pavlick E., Linzen T., « Right for the Wrong Reasons : Diagnosing Syntactic Heuristics in Natural Language Inference », *in* A. Korhonen, D. R. Traum, L. Màrquez (eds), *Proc. ACL*, Association for Computational Linguistics, 2019.

Miller T., « Explanation in Artificial Intelligence : Insights from the Social Sciences », *Artif. Intell.*, 2019.

Morris J. X., Lifland E., Yoo J. Y., Grigsby J., Jin D., Qi Y., « TextAttack : A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP », *in* Q. Liu, D. Schlangen (eds), *Proc. EMNLP*, Association for Computational Linguistics, 2020.

Pang B., Lee L., « Seeing stars : Exploiting class relationships for sentiment categorization with respect to rating scales », *Proc. ACL*, 2005.

Reimers N., Gurevych I., « Sentence-BERT : Sentence Embeddings using Siamese BERT-Networks », *Proc. EMNLP*, Association for Computational Linguistics, 2019.

Ribeiro M. T., Singh S., Guestrin C., « "Why Should I Trust You ?" : Explaining the Predictions of Any Classifier », *Proc. SIGKDD*, ACM, 2016.

Robeer M., Bex F., Feelders A., « Generating Realistic Natural Language Counterfactuals », *Findings EMNLP*, Association for Computational Linguistics, 2021.

Ross A., Marasovic A., Peters M. E., « Explaining NLP Models via Minimal Contrastive Editing (MiCE) », *in* C. Zong, F. Xia, W. Li, R. Navigli (eds), *Findings ACL/IJCNLP*, Association for Computational Linguistics, 2021.

Ross A., Wu T., Peng H., Peters M. E., Gardner M., « Tailor : Generating and Perturbing Text with Semantic Controls », *in* S. Muresan, P. Nakov, A. Villavicencio (eds), *Proc. ACL*, Association for Computational Linguistics, 2022.

Sanh V., Debut L., Chaumond J., Wolf T., « DistilBERT, a distilled version of BERT : smaller, faster, cheaper and lighter », *ArXiv*, 2019.

Shen Y., Gu J., Tang X., Zhou B., « Interpreting the Latent Space of GANs for Semantic Face Editing », *Proc. CVPR*, Computer Vision Foundation / IEEE, 2020.

Verma S., Dickerson J. P., Hines K., « Counterfactual Explanations for Machine Learning : A Review », *NeurIPS 2020 Workshop : ML Retrospectives, Surveys & Meta-Analyses ML-RSA*, vol. abs/2010.10596, 2020.

Wachter S., Mittelstadt B., Russell C., « Counterfactual explanations without opening the black box : Automated decisions and the GDPR », *Harvard Journal of Law and Technology*, vol. 31, n° 2, p. 841-87, 2018.

Wei X., Ngo C., « Ontology-enriched semantic space for video search », *Proc. International Conference on Multimedia*, ACM, 2007.

Wu T., Ribeiro M. T., Heer J., Weld D. S., « Polyjuice : Generating Counterfactuals for Explaining, Evaluating, and Improving Models », *in* C. Zong, F. Xia, W. Li, R. Navigli (eds), *Proc. ACL/IJCNLP*, Association for Computational Linguistics, 2021.

Yang L., Kenny E. M., Ng T. L. J., Yang Y., Smyth B., Dong R., « Generating Plausible Counterfactual Explanations for Deep Transformers in Financial Text Classification », *Proc. COLING*, International Committee on Computational Linguistics, 2020.

Annexe A. Détails de l'implémentation

Nous commençons par décrire les six méthodes de génération contrefactuelle utilisées pour générer des contrefactuels. Nous comblons le fossé avec deux méthodes, *Growing Net* et *Growing Language*, qui mettent en œuvre une stratégie similaire à celle des méthodes transparentes existantes. Cependant, elles le font avec moins de complexités computationnelles. Nous avons adapté le code utilisé pour générer des contrefactuels pour les trois méthodes transparentes et l'avons rendu disponible sur GitHub[6]. En revanche, nous avons utilisé le code original pour les méthodes opaques, comme décrit ci-dessous.

SEDC : nous avons modifié le code utilisé pour le masquage des mots afin de garantir sa compatibilité avec les modèles de classification qui ne produisent pas de probabilités de classe. Cette version modifiée du code est accessible sur notre GitHub en tant que variante de la classe de méthode contrefactuelle. Cette classe propose de choisir parmi SEDC, *Growing Net* ou *Growing Language*, toutes spécialisées dans la génération d'explications transparentes.

Polyjuice : pour générer des contrefactuels, nous avons utilisé le code disponible dans le lien officiel `https://github.com/tongshuangwu/polyjuice`. Nous avons utilisé les hyperparamètres par défaut avec l'utilisation de tous les codes de contrôle pour perturber les textes de chaque ensemble de test jusqu'à ce que nous trouvions 100 instances classées différemment par le modèle.

xSPELLS : nous avons utilisé la version V2 de xSPELLS, disponible sur GitHub `https://github.com/lstate/X-SPELLS-V2`, avec les hyperparamètres par défaut.

counterfactualGAN : nous avons utilisé le code fourni dans la page de sortie officielle de l'article, accessible à l'adresse `https://aclanthology.org/2021.findings-emnlp.306/`. Nous avons exécuté counterfactualGAN (cfGAN) avec les hyperparamètres par défaut.

Cette approche complète de la génération de contrefactuels garantit un ensemble diversifié de méthodes à évaluer et à comparer dans nos expériences.

Annexe B. Exemples illustratifs de contrefactuels

Nous présentons dans cette section quelques exemples de contrefactuels générés pour chaque méthode et chaque ensemble de données.

Annexe B.1. *Détection de fausses informations*

Texte original : *Obama To Apply For Political Asylum In Moneygall*

6. `https://github.com/j2launay/ebbwbb`

SEDC : ***MASK MASK MASK*** *For Political Asylum In Moneygall*

Growing Net : *Obama To* ***hold*** *For Political Asylum In Moneygall*

Growing Language : *Obama To Apply For* ***Hilarious*** *Asylum In Moneygall*

Polyjuice : *Obama is* ***expected*** *To apply for political asylum in* ***Guantanamo Bay***

cfGAN : **N/A**

xSPELLS : ***why most states are struggling to***

Annexe B.2. *Détection de spams dans des SMS*

Texte original : *Sunshine Quiz Wkly Q! Win a top Sony DVD player if u know which country the Algarve is in ? Txt ansr to 82277. aPS1.50 SP :Tyrone*

SEDC : ***MASK MASK*** *Wkly Q! Win* ***MASK*** *top* ***MASK MASK MASK*** *if u know* ***MASK MASK*** *the Algarve is in ?* ***MASK*** *ansr* ***MASK****.* ***MASK*** *Tyrone*

Growing Net : ***sun test*** *Wkly Q!* ***bring home*** *a* ***clear*** *Sony* ***videodisc musician*** *if u know which country the Algarve is in ? Txt ansr to 82277 . aPS1.50 SP : Tyrone*

Growing Language : ***Europe John*** *Wkly Q!* ***Rest*** *a* ***technical Dr Laptop*** *player if* ***sis*** *know which country the Algarve* ***feels*** *in ? Txt ansr to 82277 . aPS1.50* ***Gen. – Michigan***

Polyjuice : ***shine*** *quiz wkly q ! win* ***wkly***

cfGAN : ***##cher week wk mobile two !*** *win a* ***as earthhayaphonic*** *if u know which country the* ***chance week o*** *is in ?* ***tt opposed and send fin gives***

xSPELLS : ***you were your each re not supposed and collect is good way u any please send 50 reply after***

Annexe B.3. *Détection de sentiments dans un commentaire*

Texte original : *This is one of Polanski's best films*

SEDC : *This is one of* ***MASK MASK MASK***

Growing Net : *This is one of Polanski's* ***ill*** *films*

Growing Language : *This is one of Polanski's* ***worst*** *films*

Polyjuice : *This is one of Polanski's* ***worst movies***

cfGAN : *This is one of* ***shot kingdom intelligence's all***

xSPELLS : **N/A**