RÉPUBLIQUE DU CAMEROUN
****
Paix - Travail - Patrie
****
UNIVERSITÉ DE YAOUNDÉ I
****
FACULTÉ DES SCIENCES
****
DÉPARTEMENT D'INFORMATIQUE
****

REPUBLIC OF CAMEROON
****
Peace - Work - Fatherland
****
UNIVERSITY OF YAOUNDE I
****
FACULTY OF SCIENCE
****
DEPARTMENT OF COMPUTER SCIENCE
****

EXPLICABILITE CONTREFACTUELLE BASEE SUR LES AUTOENCODEURS VARIATIONNELS

En vue de l'obtention du diplôme de Master Recherche en Informatique

Option :
Science des données

Présenté par :
KENMOE KAMBU FRANCLINE SYLVIANE

21U2511 :
21U2511

Supervisé par :
Pr.Norbert Tsopze

Année académique 2021 / 2022

Université de Yaoundé 1
Département d'Informatique

EXPLICABILITE CONTREFACTUEL BASE SUR LES AUTOENCODEURS VARIATIONNELS

Présenté par :
KENMOE KAMBU FRANCLINE SYLVIANE

Matricule :
21U2511 **Supervisé**

par :

Pr.Norbert Tsopze

Année académique 2021/ 2022

DÉDICACES

Je dédie ce mémoire à ma famille .

REMERCIEMENTS

Avant tout, je remercie l'Éternel Dieu pour la vie, la santé et pour la grâce qu'il m'a accordée afin que j'aille jusqu'au bout de ce travail. Je témoigne ma gratitude et ma reconnaissance à toutes les personnes ayant contribué à la réalisation de ce travail, notamment :

Au Pr Norbert Tsopze à qui j'adresse mes sincères remerciements pour avoir accepté de travailler avec moi, pour son suivi, sa disponibilité, ses conseils de vie, ses encouragements et le cadre convivial qu'il m'a offert. Aux membres de mon jury de soutenance, pour leur entière disponibilité et pour l'honneur qu'ils m'accordent en acceptant d'évaluer ce travail.

À l'Université de Yaoundé 1, particulièrement à la Faculté des Sciences pour la qualité des enseignements durant mon parcours.

À tous les enseignants du Département d'Informatique pour leur disponibilité et la transmission de leur savoir.

À la communauté scientifique pour tous les articles scientifiques mis à notre disposition car ces articles constituent la base de notre travail.

À toute ma famille pour leur présence et leur soutien, et particulièrement à mes mamans Mafotsing Jeanne et Wanwou Rachelle pour l'éducation qu'elles m'ont donnée, pour leurs prières,

À mon tendre époux Kamga Serges Raoul pour sa présence, sa patience, son soutien morale et financiers, et ma petite fille Djemna Kamga Josepha Sergiane pour sa patience.

À mon grand frére Chedjou Kambu Sylvano et à tous mes petits frères et petites sœurs leurs soutiens sans faille et leurs amours inconditionnels;

À Dr Azangueset pour son soutien, ses conseils, ses encouragements et son amour; je vous aime.

À tous mes amis pour leurs présences, particulièrement à Ekwalé Béril Brandow pour son aide Acharné dans le travail, leurs orientations, leurs encouragements et leurs relectures; également à tous mes camarades de promotion pour nos échanges. Et pour terminer, je remercie tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

RÉSUMÉ

Une explication contrefactuelle se présente sous la forme d'une version modifiée de la donnée à expliquer qui répond à la question : que faudrait-il changer pour obtenir une prédiction différente? Dans notre présent rapport, nous travaillons sur une maniérée de produire les voisinages des textes courts afin d'avoir des contrefactuelles qui sont proches du texte court de départ. Le but de cette étude est d'expliquer la méthodologie utilisée par xpells, qui est celle de générer les contrefactuelles à partir des auto-encodeurs variationnels. Ces auto-encodeurs permettent la génération automatique de plusieurs voisinages. Parmi ces voisinages seront extraits les contrefactuelles. Cependant, nous constatons que le processus d'extraction produit des contrefactuelles insensées comparées à la phrase initiale. Pour répondre à ces insuffisances, nous proposons d'extraire les voisinages à partir des données brutes et produire des contrefactuelles proches de la phrase originale.

ABSTRACT

A counterfactual explanation takes the form of a modified version of the data to be explained that answers the question : what would have to be changed to obtain a different prediction? In our present report, we are working on a way of producing short text neighborhoods in order to obtain counterfactuals that are close to the original short text. The aim of this study is to explain the methodology used by xpells, which is to generate counterfactuals from variational auto-encoders, which allows the variational auto-encoder to automatically generate several neighborhoods which then extract the counterfactuals, but we find that this gives us more counterfactuals that do not go in the same direction as the starting sentence, hence the proposed solution of extracting the neighborhoods. from field data

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ABRÉVIATIONS

IA : Intelligence Artificielle
VAE : Auto-Encodeur Variationnel
ML : Machine Learning
CF : Contrefactuel

CHAPITRE 1

INTRODUCTION GÉNÉRALE

L'explicabilité [2], dans le domaine de l'intelligence artificielle, est un sujet d'une importance croissante. Il s'agit de comprendre comment les modèles d'IA prennent leurs décisions et pourquoi ils les prennent de cette manière. Cependant, l'explicabilité traditionnelle se limite souvent à fournir une explication a : c'est-à-dire expliquer pourquoi le modèle a pris une décision particulière après qu'elle ait été prise.

C'est là que l'explicabilité contrefactuelle entre en jeu. L'explicabilité contrefactuelle vise à aller au-delà de l'explication des décisions prises. Elle embrasse l'idée de pouvoir explorer différentes situations alternatives qui pourraient avoir conduit à une décision différente. En d'autres termes, il s'agit de répondre à la question comme l'a si bien dit Sahil Verma et al [6] : "Qu'aurait-il fallu changer dans les entrées du modèle pour obtenir une sortie différente?".

Ce concept d'explicabilité contrefactuelle est particulièrement utile dans les domaines où les décisions prises par les modèles d'IA peuvent avoir un impact significatif sur les individus. Par exemple, dans les domaines de la santé, de la finance ou de la justice, il est essentiel de pouvoir comprendre les raisons derrière les décisions prises par les modèles d'IA afin de garantir la transparence, la confiance et l'équité comme la dit Victor Guyomard et al [1]. L'explicabilité contrefactuelle offre également la possibilité de réaliser des audits de modèles, d'identifier les biais indésirables et d'améliorer la qualité des prédictions. En permettant aux utilisateurs de visualiser différentes trajectoires de décisions potentielles, elle favorise une meilleure compréhension des limites et des potentiels problèmes des modèles d'IA. C'est ainsi que nous allons travailler sur la méthode de génération des voisinages : cette méthode consiste à explorer le voisinage d'un exemple donné en effectuant des perturbations graduelles sur les caractéristiques afin de comprendre comment ces variations influencent la prédiction du modèle. On peut ainsi analyser les différences entre les exemples similaires pour comprendre pourquoi certains sont classés différemments par le modèle. Nous avons travaillé sur le modéle XPELLS qui utilise une approche agnostique locale pour expliquer les décisions des modéles de classification de boites noires dans la classification de textes courts. Cette derninère utilise les textes courts se basant sur les auto-encodeurs variationnels pour générer les factuelles et contre-factuelles [2]. Cependant, celle-ci rencontre plusieurs limites telles la production des contre-factuelles et factuelles qui n'ont rien à voir avec la phrase de départ, ainsi que la production massive de plus de factuelles que de contrefactuelles. Notre probléme est de savoir comment faut-il faire pour produire des contrefactuellles qui sont reéllement des voisinages de la phrase de départ.

Dans l'optique de mieux structurer et cerner notre travail, le présent document se subdivise en (4) quatre chapitres. Dans le premier chapitre il sera question pour nous de présenter les généralités et les concepts de base. Dans le second, nous présenterons la méthologie utilisée. Le troisième chapitre sera consacrer à la description de notre expérimentation et la présentation de nos resultats. Enfin, nous allons clôturer par une conclusions générale et les perspectives.

CHAPITRE 2

GÉNÉRALITÉS ET ÉTAT DE L'ART

2.1 Introduction

De nos jours, plusieurs personnes imposent de devoir expliquer une décision administrative obtenue par un traitement automatique. Le décret d'application semble très contraignant sur la précision de celle-ci. Cela peut concerner l'emploi, l'éducation, la santé, l'assurance, le traitement judiciaire, un emprunt....., etc. Cependant, une telle explication n'aide pas la personne à décider ce qu'elle doit faire par la suite pour améliorer ses chances d'être approuvées à l'avenir si la décision a été défavorable après le traitement automatique. En outre, les algorithmes d'apprentissage automatique sont de plus en plus compréhensible dans le temps en utilisant le concept d'explicablité. Cependant, expliquer ne suffit pas, car le problème véritable est de savoir comment faire pour trouver satisfaction à l'avenir en cas de désaccord. Ainsi se pose le problème de l'explicabilté contrefactuelle

2.2 Généralités

Dans cette section, nous présentons les éléments essentiels autour de l'explicabilité contrefactuelle utilisant les auto-encodeurs variationnels, à savoir l'explicabité, les réseaux de neurones, l'autoencodeur variationnel, l'explicabilité contrefactuelle, et comment on utilise les auto-encodeurs variationnels pour générer les voisinages parmi lesquels se trouvent les factuelles et les contrefactuelles.

2.2.1 Explicabilité

se réfère à la capacité de comprendre et d'expliquer les résultats ou les décisions générées par des systèmes ou des modèles. En d'autres termes, c'est la mesure dans laquelle un système ou un modèle peut être compris par les utilisateurs ou les observateurs, en fournissant des explications claires et compréhensibles sur la façon dont il a pris une décision particulière ou produit un certain résultat.

2.2.2 Factuelle et Contrefactuelle

Le terme "factuel" fait référence à ce qui est basé sur des faits vérifiables et objectifs, plutôt que sur des opinions, des croyances ou des spéculations. Une information factuelle repose sur des éléments concrets et vérifiables, tels que des données provenant de sources fiables, des statistiques, des études scientifiques ou des faits historiques. Il s'agit de données ou d'informations qui peuvent être prouvées ou vérifiées de manière objective, indépendamment des opinions ou des interprétations subjectives.

La notion de contrefactuelle se réfère à une situation hypothétique qui diffère de la réalité. Elle consiste à se poser la question de ce qui se serait passé si les conditions ou les événements étaient différents. En utilisant des modèles contrefactuels, il est possible d'analyser comment les changements dans les variables ou les paramètres auraient pu influencer les résultats ou les décisions. Cela permet de mieux comprendre les relations causales et d'évaluer l'impact potentiel de différentes actions ou stratégies

2.2.3 Explicabilité contrefactuelle

L'explicabilité contrefactuelle fait référence à la capacité d'expliquer pourquoi un événement donné s'est produit de manière différente de ce qui s'est réellement produit. Elle consiste à évaluer les facteurs ou les circonstances qui auraient pu conduire à un résultat alternatif, en identifiant les actions ou les variables qui auraient pu entraîner un changement dans le résultat final. L'explicabilité contrefactuelle vise à comprendre les mécanismes causaux qui sous-tendent un événement particulier, en proposant des hypothèses ou des scénarios contrefactuels pour expliquer les résultats observés.

2.2.4 Arbre de décision

Un arbre est un modèle d'apprentissage automatique utilisé en intelligence artificielle pour prendre des décisions en fonction d'un ensemble de conditions ou de caractéristiques. Il est composé d'un ensemble de nœuds et de branches. Les nœuds représentent les différentes conditions ou les attributs sur lesquels les décisions sont basées. Les branches connectent ces nœuds, indiquant les résultats possibles de chaque condition. À chaque nœud, une décision est prise en fonction de la valeur de l'attribut spécifié et l'arbre se déroule jusqu'à ce qu'une décision finale soit prise.

2.2.5 Cosinus de similarité

Le cosinus de similarité est une mesure utilisée pour évaluer la similarité entre deux vecteurs dans un espace vectoriel. Cette méthode est souvent utilisée dans le domaine du traitement automatique du langage naturel.

2.2.6 Distance

La distance est une mesure de la séparation ou de l'écart entre deux points, objets ou vecteurs dans un espace donné. Elle quantifie la différence entre ces points ou objets en termes de magnitude et de direction. La distance peut être calculée dans différents domaines tels que la géométrie euclidienne, l'analyse des données, la théorie des graphes, etc.

2.2.7 Auto-encodeur

Un auto-encodeur est un type d'algorithme d'apprentissage automatique non supervisé utilisé pour apprendre une représentation efficace et compacte des données en les comprimant dans un espace latent de dimensions réduites.

Il se compose de deux parties principales :

Encodeur : Cette partie convertit les données d'entrée en une représentation de dimensions réduites, généralement appelée vecteur de codage ou code latent. L'objectif de l'encodeur est de réduire la dimensionnalité des données tout en préservant les caractéristiques importantes. Décodeur : Cette partie reconvertit le code latent en une reconstruction des données d'origine. L'objectif du décodeur est de produire une reconstruction aussi proche que possible des données d'entrée initiales après compression dans l'espace latent. L'auto-encodeur vise à minimiser l'erreur de reconstruction entre les données d'entrée et les données reconstruites. La principale idée est que, en contraignant le modèle à reconstruire les données originales, les représentations apprises dans l'espace latent captureront les principales caractéristiques des données. Les auto-encodeurs peuvent être utilisés pour diverses tâches, notamment la compression de données, la réduction de dimensionnalité, la détection d'anomalies ou la génération de nouvelles données similaires à l'ensemble d'entraînement.

2.2.8 Auto-encodeur variationnel

Un VAE est un type de réseau de neurones artificiels utilisé pour la génération de données et l'apprentissage non supervisé. Il est basé sur la variante de l'auto-encodeur, qui est un type de modèle d'apprentissage automatique utilisé pour la réduction de dimensionnalité des données. Le VAE introduit un concept fondamental appelé "espace latent", qui représente un espace de représentation intermédiaire dans lequel les données peuvent être compressées et reconstruites. Par rapport à un auto-encodeur classique, le VAE introduit également une distribution probabiliste dans cet espace latent. L'objectif principal d'un VAE est d'apprendre à générer de nouvelles données similaires à celles sur lesquelles il a été entraîné. Il utilise la génération d'échantillons aléatoires dans l'espace latent et décode ces échantillons pour créer de nouvelles données qui ressemblent à celles d'origine. Cette propriété de génération de données également permet d'explorer l'espace latent et de générer de nouvelles observations qui peuvent ne pas avoir été présentes dans l'ensemble d'apprentissage. En résumé, un auto-encodeur variationnel est un modèle d'apprentissage non supervisé capable de générer de nouvelles données similaires à celles sur lesquelles il a été entraîné, en utilisant un espace latent probabiliste.

2.3 État de l'art

Ramaravind K. Mothila et al [4] disent que les explications contrefactuelles efficaces doivent remplir les propriétés suivantes : la diversité et la faisabilité, la proximité et la validité en tenant compte des contraintes des utilisateurs, c"est ainsi que pour évaluer l'efficacité des contrefactuelles, ils fournissent des métriques permettant d'évaluer et aussi comparent les méthodes basées sur les contrefactuelles à d'autres méthodes locales.

2.3.1 Génération des contrefactuels

En effet d' aprés Ramaravind K. Mothila et al [4] l'explication à une décision ne suffit pas, car en général l'explication sur le refus de la candidature à un emploi n'aide pas la personne à décider ce qu'elle doit faire ensuite pour améliorer ses chances d'être approprié dans l'avenir, d'où l'importance de l'explicabilité contrefactuelles et ils disent qu'un contrefactuel doit respecter cette structure soit une caractéristique X et la sortir correspondante alors, on a :
Entrainez un modèle de machine learning à partir des données d'entrainement

Entrée Originale (X) : Entrez un vecteur (cela peut être une phrase, des nombres, de l'image...
ET)
Sortie des vecteurs : la sortie des vecteurs dans le modèle de machine renvoie une décision pas favorable
Modification; modifier là caractérise la plus minime possible du vecteur que le résultat soit favorable
Résultat : sortie favorable

Moteur de génération des contrefactuels

L'entrée de notre problème est un modèle d'apprentissage automatique entraîné, f , et une instance, x. Nous aimerions générer un ensemble de k exemples contrefactuels, $c1,c2,...,ck,$ de telle sorte qu'ils conduisent tous à une décision différente de x. L'instance (x) et tous les exemples CF ($c1,c2,...,ck$) sont d-dimensionnels. une explication contrefactuelle est une perturbation de l'entrée pour générer une sortie différente y par le même algorithme.D'apres Sandra Wachter et al [7] un contrefactuel doit répondre à celle formule :

$$c = argminyloss(f(c),y) + |x - c|]$$

où la première partie (yloss) pousse le contrefactuel c vers une prédiction différente de celle de l'instance d'origine, et la deuxième partie maintient le contrefactuel proche de l'instance d'origine.

La contrainte de diversité et de faisabilité donne divers exemples de contrefactuels Bien que les chances qu'au moins un exemple soit exploitable pour l'utilisateur sont minimes, les exemples peuvent finir par modifier un large ensemble de fonctionnalités, ou maximiser la diversité en considérant de grands changements par rapport à l'entrée d'origine, d'où un bon contrefactuel doit répondre à la formule de la diversité suivante

$$diversity = det(K)]$$

avec

$$k_{i,j} = \overline{1 + dist1(c_i,c_j)}$$

ou c^i et c_j sont des exemples de contrefactuelles la dist c^i,c_j est la différence entre deux exemple de contrefactuels apres avoir parlé de la diversité on continue avec comme propriéte la proximité qui permet la recherche des es exemples CF les plus proches de l'entrée d'origine car elles sont les plus utiles pour un utilisateur. elle répond à la formule suivante :

$[Proximity = {}_{k}^{1}\mathrm{P}_{n}{}_{i=1}\, dist(c_i,x)]$ x etant le vecteur originale ,c_i le contrefactuel ,k nombre de contrefactuels
La propriété de faisabilité de la parcimonie est étroitement liée à la proximité : combien de fonctionnalités un utilisateur doit-il modifier pour passer à la classe contrefactuelle. Intuitivement, un exemple contrefactuel sera plus réalisable s'il apporte des modifications à un nombre inférieur de fonctionnalités.

Contraintes des utilisateurs. Un exemple contrefactuel peut être proche dans l'espace des fonctionnalités, mais peut ne pas être réalisable en raison des contraintes du monde réel. Il est donc logique de permettre à l'utilisateur de fournir des contraintes sur la manipulation des fonctionnalités. Ils peuvent être spécifiés de deux manières. Premièrement, tant que la contrainte de la boîte noire est sur les plages réalisables pour chaque fonctionnalité, dans lesquelles les exemples contrefactuelles

doivent être recherchés. Un exemple d'une telle contrainte est : « les revenus ne peuvent pas augmenter au-delà de 200 000 ». Alternativement, un utilisateur peut spécifier les variables qui peuvent être modifiées. Avec la diversité et la proximité, ils définissent une fonction de perte pour la génération des contrefactuels. $c(x) = argmin_k \frac{1}{} \sum_{i=1}^{n} yloss(f(c),y) + \frac{\gamma}{k} \frac{1}{} \sum_{i=1}^{n} dist(c_i,x)] - \gamma_2 dppdiversity(c_1,...,c_k)$

où c_i est un exemple contrefactuel (CF), k est le nombre total de CF à générer, f (.) est le modèle ML (une boîte noire pour les utilisateurs finaux), yloss(.) est une métrique qui minimise la distance entre La prédiction de f (.) pour cis et le résultat souhaité y (généralement 1 dans nos expériences), d est le nombre total de fonctionnalités d'entrée, x est l'entrée d'origine et diversity(.) est la métrique de diversité. $gamma^1$ et sont des hyperparamètres qui équilibrent les trois parties de la fonction de perte.Ramaravind K. Mothilal et al [5] ont donc comparé leur méthode de génération de contrefactuel au méthode de base suivant :

- SingleCF : Wachter et al.[**?**] utilise cette methode de base pour générer un seul Exemple CF, pour l'avoir on fait la difference entre la fonction d' optimisation et la proximité de perte y.
- MixedIntegerCF : Nous utilisons la méthode de programmation en nombres entiers mixtes proposé par Russell .[3] , pour générer diverses hypothèses contrefactuelles exemples. Cette méthode ne fonctionne que pour un modèle linéaire
- RandomInitCF : nous étendons ici SingleCF pour générer k exemples CF en initialisant l'optimiseur indépendamment avec k random .points de départ à partir de [0, 1] Depuis la fonction de perte d'optimisation est non convexe, on pourrait obtenir différents exemples de CF.
- NoDiversityCF : cette méthode utilise notre fonction de perte proposée qui optimise simultanément l'ensemble de k exemples,mais ignore le terme de diversité en définissant $\gamma^2 = 0$

En conclusion ils ont démontré que les différentes méthodes de génération des contrefactuels proposée par rapport aux approches passées en tant qu'approches enregistrés a plus d'avantage car les contrefactuelles trouvée sont utile et valable mais aussi le constat est que leurs méthodes proposé ne fonctionne pas sur toute les methode d'intélligence artificielle.

2.3.2 Générations des contrefactuelles avec les images

La solution proposée par vitor et al [1] consiste à introduire dans le processus de génération du contrefactuel un terme basé sur un auto-encodeur supervisé. Ce terme contraint les explications générées à être proches de la distribution des données et de leur classe cible.

En effet, le processus de génération d'explications contrefactuelles post-hoc peut conduire à une contrefactuelle peu réaliste, c'est ainsi qu'en 2020 l'utilisation des auto-encodeurs supervisée sera proposé pour la génération des contrefactuelles.

Les trois métriques pour évaluer la qualité des contrefactuels sont : le gain de prédiction, le réalisme et l'actionnabilité.

- Gain de prédiction : c'est la différence entre la probabilité prédite du contrefactuel et la probabilité prédite de l'exemple à expliquer pour la classe du contrefactuel.

$$Gain = [f_pred(x_cf)]_{y_i} - [f_{pred}(x_0)]_{y_i}$$

où y_i est la classe prédite pour le contrefactuel. Le gain de prédiction est borné dans [0;1], et une valeur plus élevée implique plus de confiance dans le changement de classe du contrefactuel.

- Réalisme : Le réalisme correspond à l'erreur de reconstruction du contrefactuel évaluée par un autoencodeur (AEevaluate)..

$$Réalisme = ||AE_{evaluate}(x_{cf}) - x_{cf}||_2$$

Une valeur faible indique que le contrefactuel est plus proche de la distribution des données.

- Actionnabilité : c'est la distance $L1$ entre l'exemple et le contrefactuel.

$$Actionnabilité = ||x_{cf} - x_0||_1 = ||p||_1$$

Une faible valeur indique une perturbation plus creuse, ce qui correspond au changement de moins de caractéristiques.

La Figure 2.1 présente les contrefactuels produits par les deux méthodes pour trois exemples différents. La colonne de gauche montre les exemples à expliquer, les deuxième et troisième colonnes représentent respectivement les contrefactuels obtenus avec un auto-encodeur supervisé et la méthode baseline.

Nous observons globalement plus de changements de pixels dans le cas de l'auto-encodeur supervisé. Par exemple, sur la première ligne d'images, un exemple prédit comme un 3 aura un contrefactuel prédit comme un 6 dans les deux cas. Cependant, on observe que plus de pixels ont été modifiés avec la méthode basée sur l'auto-encodeur supervisé (avec la création de la boucle qui permet d'obtenir un 6). De plus, le contrefactuel obtient une plus grande probabilité de prédiction pour la classe 6 (0 :99 vs 0 :41), ce qui conduit à un gain de prédiction plus important. Sur la deuxième ligne, l'exemple à expliquer est prédit comme un 9 et les contrefactuels comme un 5. Nous observons le même comportement que précédemment sur les contrefactuels. Dans le cas de l'auto-encodeur supervisé, des pixels sont modifiés dans la boucle du 9 afin d'obtenir un 5 alors que moins de pixels sont modifiés dans le cas de la baseline, ce qui induit moins de changements visuels (le contrefactuel est prédit comme un 5 mais ressemble visuellement à un 9). Dans la dernière ligne, l'exemple est prédit comme un 2 et le contrefactuel produit, est prédit comme un 1 dans le cas de l'auto-encodeur supervisé, et comme un 8 dans le cas de la baseline. Pour la baseline, la perturbation produit un contrefactuel qui apparaît comme étant visuellement similaire à l'exemple alors que la classe prédite est 8 (avec une probabilité de prédiction de 0,43). Avec l'auto-encodeur supervisé, le contrefactuel est

En conclusion, notre proposition consiste à introduire un terme basé sur un auto-encodeur supervisé dans la fonction de coût qui permet d'obtenir le contrefactuel à partir de l'exemple à expliquer. Ce travail est une extension de celui de Van Looveren et Klaise avec l'objectif d'améliorer la fidélité des contrefactuels à la distribution des données de la classe cible. La méthode a été évaluée sur un jeu de données d'images par le biais de différentes métriques. On a obtenu des contrefactuels avec une plus grande confiance dans leur classe de prédiction, mais au prix d'une modification d'un grand nombre de caractéristiques de l'exemple à expliquer. Étant donnée qu'ils ont modifié un grand nombre de caractéristiques pour avoir ces contrefactuels, nous nous posons les questions de savoir si la modification d'un grand nombre de caractéristiques de départ ne change pas le contenu de départ.

2.3.3 Auto-encodeur variationnels pour la génération des contrefactuels

xpells propose une solution de prodution des contrefactuels en utilisant comme données les textes courts. En effet, les textes courts sont envoyés a un encodeur, puis ceux-ci sont codés et envoyés à l'espace latent et dans cet espace latent de l'auto-encodeur variationnel, il y'aura production automatique des phrase proches de la phrase de départ appelée voisinage. Ces voisinages codés, sont envoyés au décodeur, puis à l' arbre de décision afin que celle-ci founir une explication à chaque phrase afin qu'on sache si c' est un exemplaire ou un contre-exemplaire. Parmir ces contreexemplaires et exemplaires seront selectionnés ceux les plus récurrent pour produire les factuelles et

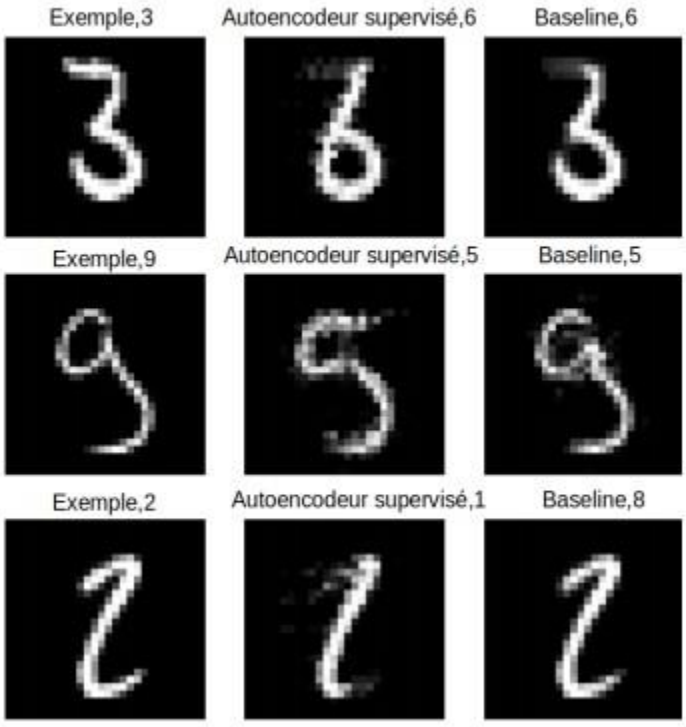

Figure 2.1 – Exemples à expliquer de MNIST (colonne de gauche) et les contrefactuels correspondants pour la méthode basée sur l'autoencodeur supervisé (au centre) et la méthode baseline (à droite). converti en un 1 avec une probabilité de 0 :99. Il a été obtenu par la modification des pixels aux extrémités.

les contrefactuelle. Notre intérêt portera sur les contrefactuelles. Apres avoir éxécuter, nous constatons que le systéme produit des contrefactuelles, ce qui est déja admirable, mais l'inconvénient est que la majorité des contrefactuels produisent un problème de validité et de sparcité.

2.4 Tableau récapitulatif

Le tableau présente un récapitulatif des approches étudiées concernant la génération des explications contrefactuelles. En ligne, nous avons les différentes auteurs et en colonne, nous avons la méthodologie employée dans chaque approche et les limites relevées.

AUTEURS	APPROCHES	METHODOLOGIE	INCONVENIENT
Ramaravind K. Mothilalet al	modifiifations des propriété des contrefac-tuels	combinaisons des proprité (la diversité,la proximité)	elle ne s'applique pas sur toutes les méthode de l' intélligence artificielle
Victor Guyomard et al	contraint les explications générées à être proches de la distribution des données et de leur classe cible.	introduire dans le processus de génération du contrefactuel un terme basé sur un auto-encodeur supervisé	modification d'un grand nombre de caracteristique
Orestis Lampridis et al	generation du voisinage	introduction des autoencodeurs variationnels pour produire le voisinage de maniere automa-tique	les voisinages générer pour la contruction des contrefactuels ne sont pas proches des phrases d' origine

Table 2.1 – Récapitulatif des méthodes de génération des explications contrefactuelles

Dans ce mémoire, nous traitons la question :**comment évité de générer aleatoire?** Car la génération aléatoire des voisinages ne nous donne aucun control sur les voisinages générer

2.5 Conclusion

Dans ce chapitre, nous avons présenté les différents concepts liés à la génération des explications contrefactuelles et passé en revue les différentes approches de génération des explications contrefactuelles, les approches basées sur les modifications de la propriété et les approches basées sur introduction de l'auto-encodeur supervisé et auto-encodeur variationnels. Dans le chapitre suivant, nous faisons une étude du problème de validité et sparcité des contrefactuelles et proposons notre solution à ce problème.

CHAPITRE 3

MÉTHODOLOGIE

3.1 Introduction

Dans le chapitre précédent, nous avons présenté les méthodes de génération des contrefactuelles sur les images et sur les textes. Nous nous sommes plus attardé sur celle de la génération des textes. D'après

Orestis Lampridis et al [2] la génération des contrefactuelles basée sur les auto-encodeurs variationnels est la meilleure façon de le faire. Toutefois, leurs travaux ne prennent pas en compte le fait que les contrefactuelles peuvent être insensées comparer à la phrase de départ. Dans ce chapitre, nous allons étudier ce problème et proposer une solution permettant de l'adresser.

3.2 définition

La "distance moyenne" est une mesure statistique qui calcule la valeur moyenne des distances entre tous les points d'un ensemble de données. Pour la calculer, il faut prendre la somme de toutes les distances entre les points, puis la diviser par le nombre total de paires de points.

La "distance médiane" est une mesure statistique qui représente la valeur au milieu de toutes les distances entre les points d'un ensemble de données, lorsque les distances sont triées par ordre croissant. Elle est utilisée pour représenter une mesure de centralité plus robuste que la moyenne, car elle n'est pas affectée par les valeurs extrêmes.

La "distance minimale" est la plus petite distance entre deux points, objets ou vecteurs dans un ensemble de données. Elle indique la proximité la plus proche entre les éléments et peut être utilisée pour déterminer la similarité ou la différence entre des éléments.

3.3 Model Xpell et Rappel du Problémé

3.3.1 XPELLS

xspells, une approche locale agnostique de modèle pour expliquer les décisions des modèles de boîte noire dans la classification de textes courts. Les explications fournies consistent en un ensemble de phrases exemplaires et un ensemble de phrases contre exemplaires. Les premières sont des exemples classifiés par la boîte noire avec la même étiquette que le texte à expliquer. Les secondes sont des exemples classés avec une étiquette différente (une forme de contrefactuel). Les deux sont proches du texte à expliquer, et les deux sont des phrases significatives : bien qu'elles soient générées synthétiquement. xspells génère des voisins du texte à expliquer dans un espace latent en utilisant des auto-codeurs variationnels pour encoder le texte et décoder les instances latentes. Un arbre de décision est appris à partir des voisins générés de manière aléatoire, et utilisé pour conduire la sélection des exemplaires et des contre-exemples. De plus, la diversité des contreexemples est modélisée comme un problème d'optimisation, résolu par un algorithme glouton avec une garantie théorique.

Auto-codeurs variationnels pour la génération des contrefactuelles avec Xspells

Nous abordons les VAE qui se sont avérés très efficaces pour générer des phrases diverses et bien formées (texte court). Un VAE est entraîné dans le but d'apprendre une représentation qui réduit la dimensionnalité de l'espace des mots à l'espace latent, capturant également les relations non linéaires. Un encodeur et un décodeur, sont appris simultanément avec l'objectif de minimiser la perte de reconstruction. À partir de l'encodage réduit, le VAE reconstruit une représentation aussi proche que possible de son entrée originale x. Après apprentissage, le décodeur peut être utilisé avec des objectifs génératifs pour reconstruire des instances jamais observées en générant des vecteurs en l'espace latent de dimension k. La différence avec les autoencodeurs est que les VAE sont entraînés en considérant une limitation supplémentaire de la fonction de perte, de sorte que l'espace latent soit dispersé et ne

contienne pas de "zones mortes". En effet, le nom "variationnel" vient du fait que les VAE fonctionnent en approchant la distribution postérieure avec une distribution variationnelle. L'encodeur émet les paramètres de cette distribution variationnelle, en termes d'une distribution gaussienne multifactorielle, et la représentation latente est prise par échantillonnage de cette distribution. Le décodeur prend en entrée la représentation latente et s'attache à reconstruire l'entrée originale à partir de celle-ci. L'évitement des zones mortes garantit que les instances reconstruites à partir des vecteurs dans l'espace latent.

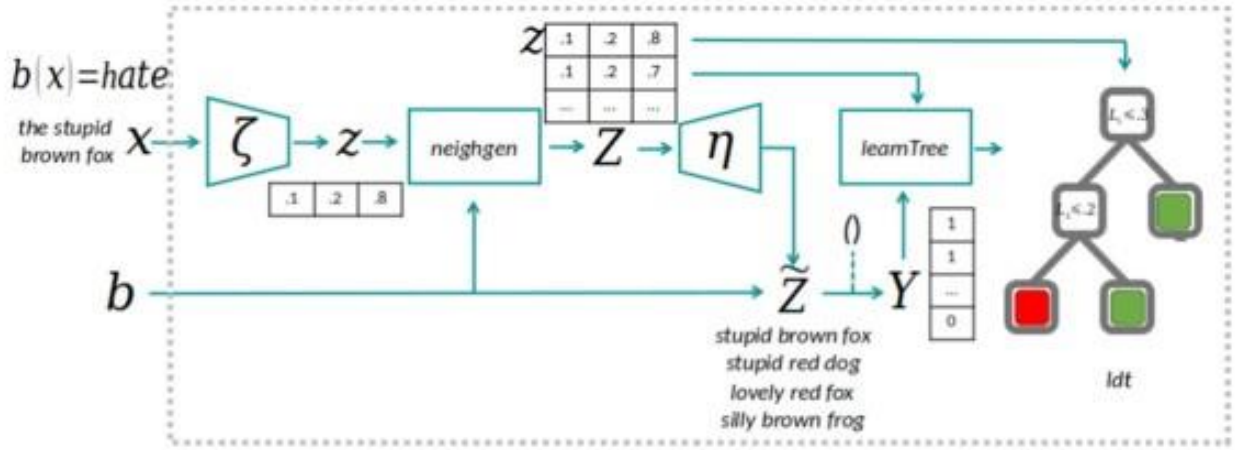

Figure 3.1 – Production des explication contrefactuelle

Étant donné une boîte noire b, un texte court x, par exemple un message haineux, et l'étiquette de classe $y = b(x)$ attribuée par la boîte noire, par exemple, haineux ou neutre, l'explication fournie par xspells est composée des éléments suivants :

(i) Un ensemble de textes exemplaires; (ii) un ensemble de textes contre exemplaires; (iii) l'ensemble des mots les plus courants dans les textes exemplaires et contre-exemplaire. Les textes, exemplaires et contre-exemplaire illustrent respectivement des instances classées avec la même étiquette et avec une étiquette différente de x. De tels textes sont proches de x, et ils permettent de comprendre ce qui fait que la boîte noire détermine l'étiquette des textes dans le voisinage de x. Les textes exemplaires aident à comprendre les raisons, par exemple, du sentiment attribué à x. Le contre exemplaire aide à comprendre les raisons qui inverseraient le sentiment attribué. Les mots les plus courants dans les exemples et les contre-exemples peuvent permettre de mettre en évidence les termes (qui n'apparaissent pas nécessairement dans x) qui font la différence entre l'étiquette attribuée et une autre.

Ces composants forment l'explication interprétable par l'homme et pour la classification $y = b(x)$. Après avoir expliqué la fonction de xpells, nous passons à l'algorithme utilisé par XSPELLS

Algorithme de production des explications contrefactuelles

Algorithm 1 XPELLS(x,b, ς , η)

Data : *x − texttoexplain,b − blackbox,ς − encoder,η − decodeur*

Result : *ξ − explaination*

1 $z \leftarrow \varsigma(x)$ ▷ *[r]le text est encodé dans l' espace latent $Z \leftarrow neighgen(x,b,\varsigma,\eta)$ ▷ *[r]dans l'espace latent le voisinage est généré $\tilde{Z} \leftarrow neighgen(\eta(z)$ ▷ *[r]apprendre l'abre de décision à comprendre ce qui se passe dans l'espace latent $y \leftarrow b(\tilde{Z})$ $Idt \leftarrow learTree(Y,Z)$ $r \leftarrow rule(Z,Idt)$ ▷ *[r]extrait les règle factuelles $E,C \leftarrow ExplCexpl(r,Z,Z,\tilde{Y})$ ▷ *[r]extraire les mots les plus courants $\xi = (E,C,w)$ ▷ *[r]retourne les explications

3.3.2 Rappel du probléme de Xppell

la similarité est faible (peut de mot en commun entre les voisinages produit et la phase original .
-J'ai inspecter de maniere visuel et utiliser aussi des calcul

inégalité entre les différentent classes produit (les phrases de la meme classe que la phrase originale sont plus produtes) Nous sommes particulierement artardé sur la faible similarité; voici dont quelques tableau pour montré la faible similarité

X(phrase orinale)	nbre voisinages générer	mot en commun	distance
not fun uk when you have braxton hicks and cramps regularly from moving around you try it prediction	0 : when people really go to say much predic- tion 1 : hey why dont u get involved in black 2 : it s stuff like this that makes black 3 : almost got to see a white boy get prediction 4 : hey go look at that video of the dolphins prediction	0	Distance minimale : 0 Distance moyenne : 0 Distance mediane : 0.0
the ending to that yankee game was so planned i can t even take it	0 : it s stuff like this that makes black 1 : hey do visiting teams whine about 2 : just because some moist bint lobbed 3 : a homeless dog living in a trash pile 4 : it s gon na be a long season for the	0	Distance minimale : 0 Distance moyenne : 0 Distance mediane : 0.0

getting off twitter jigg trynna stress me out more i don t need this type of heat	0 : just because about s go to jail bird 1 : no tv during the week rule has thrown 2 : just learn d dance nuh cuz u know 3 : no you ignorant cement headed hillbilly 4 : if you enjoy hunting for spo then	0	Distance minimale : 0 Distance moyenne : 0 Distance mediane : 0.0

Table 3.1 – tableau des voisinage générer pour xpell pour n = 5 voisin

Apres avoir présenter un tableau de 3 phrase avec 5 voisinage on contaste les 5 voisin enregistre 0 mot en commun avec chacune des voisinage .évoluons avec le cas de n = 10 voisin

X (phrase originale)	vosinage générer	mot commun
X_o	10	1
X_1	10	0
X_2	10	0
X_3	10	0
X_4	10	0

Table 3.2 – Présentation de 4 phrase ayant générer 10 voisin chacune

On constate que les quatre phrases, on génère chaque' un 10 voisin en commun et on a 1 mot en commun avec le voisinage de la première phrase et o pour le voisinage des autres phrases passons à 10 phrase en commun avec un voisinage de 20 phrase chacune

X (phrase originale)	vosinage générer	mot commun
X_o	15	o
X_1	15	1
X_2	15	0
X_3	15	0
X_4	15	0
X_4	15	0
X_6	15	0
X_7	15	03
X_8	15	0
X_9	15	3

Table 3.3 – Présentation de 10 phrase ayant générer 15 voisin chacune

Nous constatons après affichage du tableau que les 10 phrases ayant généré 15 voisins chacun, la 2 phrase enregistre deux mots en commun avec son voisinage, la 7 phrase qui enregistre trois mots en commun avec son voisinage , et la 9 phrase 3 mot en commun avec ses voisinage.et le reste des phrases enregistre zéro mot en commun avec leur voisinage, on peut donc conclu que la similarité entre les phrases et les voisinages est très faible au niveau des mots en commun avec la méthode de XPELLS d'où

le problème comment éviter de générer aléatoirement les voisinages dans la construction des voisinages?

3.4 solution proposé

L'idée etant d' éviter de génerer aléatoirement le voisinage comme avec l' auto-encodeur variationnel ,on a choisir rétirer le voisinage des donné d'apprentissage et pour le faire on proposé l'approche suivante : Une première étude a été d'analyser le code afin d'assimiler le fonctionnement de chaque module. Une remarque a été faite sur le module de génération des voisinages : dans ce module, les auteurs ont utilisé l'auto-encodeur variationnel pour la génération aléatoire des voisinages. L'idée de l'auto-encodeur variationnel était de générer les voisinages afin que partir ceux-ci soient choisis les contrefactuelles, mais on se rencontre que la majorité des voisinages générée sont loin des phrases originales, par conséquent loin des contrefactuelles attendues. **1 -Ranger les phrase de la classe désirer par ordre de proximiter par rapport à X**

2 -choisir les k plus de x.

Pour la première étape qui est celle de **-Ranger les phrase de la classe désiré par ordre de proximité par rapport à X** nous avons introduit le cosinus de similarité afin qu'on l'utilise cela pour calculer la similarité p entre le nombre de mots de chaque phrase des données d'apprentissage et la phrase originale.

Et par la suite pour pour la seconde étape qui de **— Choisir les k plus de x.** nous avons utilisé une fonction **nombre-en-commun.** qui permet de calculer et de données le nombre en commun entre la phrase originale et les différents voisins et ensuite, nous avons aussi introduit une mesure appelée distance qui permet d'avoir une vue globale sur le nombre de mots en commun

En effet, ces fonctions de distances permettent de grouper les mots en commun. Premièrement une calculée le plus petit nombre de mots commun entre la phrase et ses voisins, celle-ci est appelée **distance minimale.** et par la suite une autre calcule le nombre de mots commun moyen entre une phrase et son voisin, celle -ci est appelé **distance moyenne .**, enfin cette dernière qui calcule la médiane de nombre de mots en commun entre la phrase et et ses voisins, on lui a donné le nom de **distance médiane.**

```
Phrase 20 : still really confused re i think lucricus is a fag dagger prediction : 0
Phrases du voisinage
0 : you re responsible for the story inside  prediction : 1  Nombre de mots en commun : 1
1 : if you enjoy hunting for spo then  prediction : 0  Nombre de mots en commun : 0
2 : hey go look at that tanks can t wait  prediction : 0  Nombre de mots en commun : 0
3 : hey why dont u get involved in black  prediction : 1  Nombre de mots en commun : 0
4 : hey why dont u get mad cause black  prediction : 1  Nombre de mots en commun : 0
5 : no you ignorant cement headed hillbilly  prediction : 0  Nombre de mots en commun : 0
6 : if you like after romeo chances are  prediction : 0  Nombre de mots en commun : 0
7 : just learn d dance nuh cuz u know  prediction : 0  Nombre de mots en commun : 0
8 : it s so shady when you bitches niggers  prediction : 0  Nombre de mots en commun : 0
9 : hey go look at that video of the dolphins  prediction : 0  Nombre de mots en commun : 0
Distance minimale : 0
Distance moyenne : 0.1
Distance mediane : 0.0
```

Figure 3.2 – les distances entre la phrase 20 et ses différents voisins

Avec cette méthode proposée, nous enregistrons des avantages et quelques manquements/

Avantages

Cette solution nous permet de contrôler les données, ce qui était le contraire avec les autoencodeurs

De plus, elle est agnostique, c'est - à - dire s'applique à tous les modèles

Manquement

elle nous fait perdre beaucoup en temps, car il faut comparer la phase originale avec toutes les autres phrases des données d'apprentissage elle occupe beaucoup de mémoire parce qu'il faut calculer le nombre de mots en commun et

stocké à chaque fois

3.5 Conclusion

Ce chapitre a été consacré à la résolution du problème de sparcité et de validité, le problème que nous avons détaillé dans un premier temps afin de noter son importance. L'approche que nous avons proposée consiste à modifier la fonction de génération des contrefactuelles afin qu'elle ne soit plus produire de manière automatique mais qu'elles viennent plutôt de la donnée d'apprentissages.

Dans le chapitre suivant, nous présentons le protocole expérimental et les résultats obtenus.

CHAPITRE 4

EXPÉRIMENTATIONS ET RÉSULTATS

4.1 Introduction

Xspells est une approche locale agnostique pour expliquer les décisions prises par les modèles (boites noires) utilisés dans la classification de textes courts. Les explications fournies par xpells consistent en un ensemble de phrases exemples et un ensemble de phrases contre-exemples. Les premières sont des exemples classifiés par la boîte noire avec la même étiquette que le texte à expliquer. Les secondes sont des exemples classés avec une étiquette différente (une forme de contrefactuel). Les deux sont proches du texte à expliquer, et les deux sont des phrases significatives bien qu'elles soient générées synthétiquement.

4.2 Jeu de données

Les phrases sur lesquelles nous travaillons sont issues d'un dataset pris sur Github et accessible à l'adresse https://github.com/lstate/X-SPELLS-V2. Il s'agit d'un ensemble de 25296 phrases courtes collectées sur Tweeter , utilisées pour une tâche de détection de texte haineux.

4.3 Métriques d'évaluations

La génération des contrefactuels dans notre cas passe par la génération des voisnages. Cependant, il est nécessaire d'évaluer la qualité du voisinage car si notre voisinage n'a rien en commun avec le texte original, même le contrefactuel sera loin de celui attendu.

Dans ce memoire nous nous attardons sur la qualité du voisinage produit vis à vis de la phrase originale. Pour cette raison, nous avons introduit le consinus de similarité dans XPELLS ce qui nous à permis d'écrire une métrique qui va nous permettre de calculer le nombre de mot en commun de chaque voisinage et sa phrase originale. Par la suite, nous avons utilisé la metrique de distance pour recherché le voisinage qui à le moins de mot en commun avec la phrase orignale appelélée distance minimale. De plus nous avons utilisé une distance permettant de réchercher la moyenne de mot en commun des voisinages et la phrase orignale, que nous appelons distance moyenne. Enfin, nous avons calculé la médiane entre les voisinages et la phrase orignale et que nous avons appelé distance médiane. Nous avons appliqué ces différentes distances sur les phrases de XPELLS.

Soit deux vecteurs $A = (A_1, A_2, A_3, \cdots A_n)$ et B, A_n étant le vecteur de voisinage et B le vecteur de la phrase originale. Le cosinus de leur angle ϑ s'obtient en prenant leur produit scalaire divisé par le produit de leurs normes :

$$\cos\vartheta = \frac{A_n.B}{\|A\|.\|B\|}$$

où le cos étant comprise entre 0 et 1.

Nous avons également utilisé la distance pour faire les différentes comparaisons. Nous avons choisi la distance de Manhattan qui se définie comme suit :

$$DistManh = \sum_{i=1}^{n} |A_i - B_i|$$

avec A_i le nombre de mots commun du voisinage avec la phrase originale et B le nombre de mots de la phrase originale .

Avec cette formule nous allons pouvoir obtenir la distance minimale entre la phrase originale et tout ses voisinages qui sera egale à

$$\min(\cos\vartheta = \frac{A_n.B}{\|A\|.\|B\|})$$

En suite la distance moyenne

$$MeanDist = \frac{1}{n} \sum_{i=1}^{n} DistManh$$

Pour finir, nous avons la distance mediane, qui demande d'abord de trier les données dans l'ordre croissant ou décroissant, puis de trouver la valeur qui se situe exactement au milieu de l'ensemble de données. Si le nombre total de données est impair, la médiane sera la valeur centrale. Par exemple, dans le jeu de données $\{1,2,3,4,5\}$, la médiane serait 3. Si le nombre total de données est pair, la médiane

sera la moyenne des deux valeurs centrales. Par exemple, dans le jeu de données $\{1,2,3,4,5,6\}$, la médiane serait $(3 + 4)/2 = 3,5$.

4.4 Protocole expérimental

Le framework xpells a été développé par ses auteurs en python en utilisant plusieurs bibliothèques. Il s'agit de :

- Keras : pour l'entrainement et l'utilisation des modèles;
- Sklearn pour le prétraitement des données et les métriques;
- Nltk 3.4.4, pour les traitements sur les textes;
- Numpy 1.20.0 : pour les opérations sur les vecteurs.

La liste complète des bibliothèques utilisées se trouve dans le fichier « environment.yml » disponible dans le dossier du projet xpells. De ce fait, pour les expérimentations, nous avons utilisé le langage python et les bibliothèques précédemment citées. Nous avons utilisé un IDE anaconda en invite de commande et le logiciel Visual Studio Code version 1.67.2 sur un ordinateur tournant sur le système d'exploitation Windows 10 64 bits (build 19044).

Pour ce qui est du matériel utilisé, il s'agit d'un ordinateur classique DELL modèle Latitude E5440, 4 Go de RAM et 500 Go de ROM HDD.

4.5 Résultats

Nous avons mené plusieurs expérimentations sur les différents modèles présentés au chapitre 3 y compris celui de l'article; nous allons montrer tour à tour les expérimentations efféctué :

Nous avons souligné au chapitre précédent que le probléme majeur de xpells etait le fait que la majorité des contrefactuels produit n'avaient rien en commun avec la phrase orignale; nous pouvons donc constater à travers cette excécution :

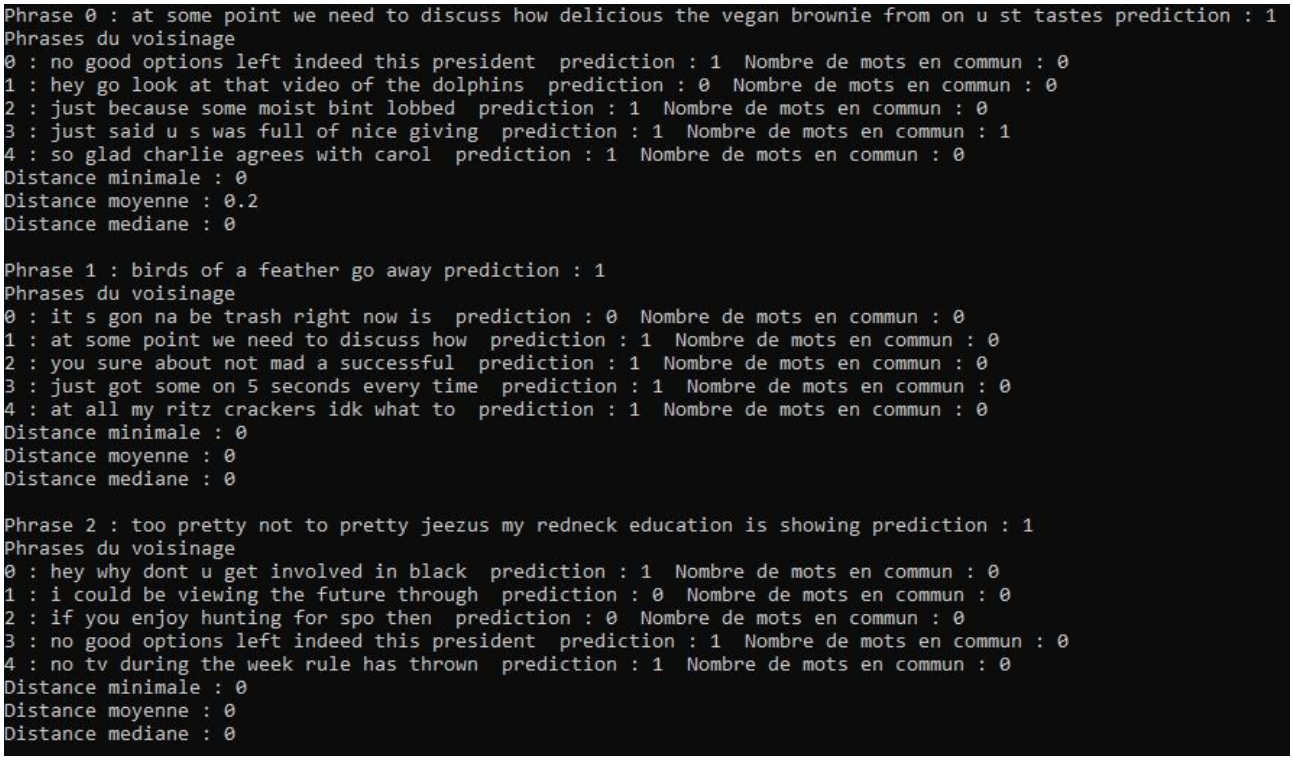

Figure 4.1 – voisinages produits avec la méthode de Xpells

Apres excécution nous constatons que la phrase 0 après prodution de 5 voisinages n'a qu'un seul mot en commun avec le quatrième voisinage et en plus un mot qui ne peut pas nous donner une grande signification. La phrase 1 n'a aucun mot en commun avec ses voisinages ainsi que la phrase 2. Et vu que les voisinages n'ont pas de mots en commun avec les voisinages alors les distances des valeurrs nulles sauf celui de la phrase 1. Donc la distance minimale est égale 0, la distance moyenne vaut 0,2, et la distance médiane vaut 0. Avec les voisinages qui n'ont rien à voir avec la phrase originale, nous constatons qu'on ne peut pas avoir les contrefactuels car ses contrefactuels sont tirés parmir ses voisinages. Ainsi, cette methode de production des voisinages de maniere automatique en utilisant les auto-encodeurs variationnels n'est pas une éfficace. Nous avons pensé qu'au lieu que les voisinages soient produits de maniere automatique dans l'espace latent de l'auto-encodeur vatiationnel en changeant un pétit parametré, il est préférable que les voisinages soient extraits des données brutes.

Après excécution, nous constatons que la phrase 0 après prodution de 5 voisinages possède désormais les mots en commun. Nous allons utiliser les tableau pour mieux expliquer.

La distance minimale est égale à 0, la distance moyenne est égale à 2,6 et la distance médiane vaut 2.

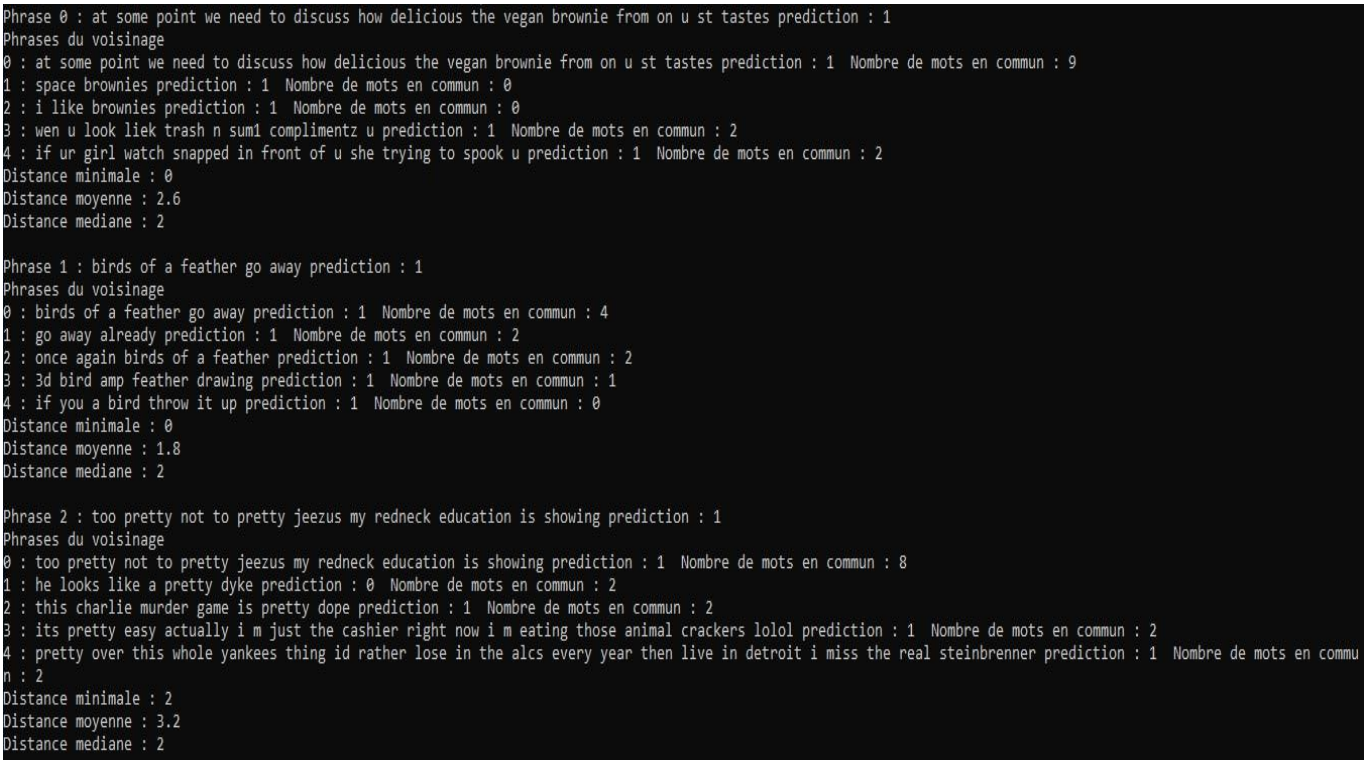

Figure 4.2 – génération des voisinages aprés utilisation de la méthode proposé

phrase	vosinage	mot commun
0	0	9
	1	0
	2	0
	3	2
	4	2

Table 4.1 – Présentation du nombre de mot en commun et la phrase 0 original

phrase	vosinage	mot commun
1	0	4
	1	0
	2	0
	3	2
	4	2

Table 4.2 – Présentation du nombre de mot en commun et la phrase 1 original

Et concernant la phrase 1 on a : la distance minimale vaut 0, la distance moyenne vaut 1.8 et la distance médiane vaut 2.

Et en fin la phrase 2 on a : la distance minimale vaut 2, la distance moyenne vaut 1.8 et la distance médiane 2.

Après l'évalution des trois phases, nous constatons que la majorité des voisinages ont des mots en commun avec leurs phrases originales. Nous constatons aussi que, dans ces voisinages nous trouverons des contrefactuels qui ne sont pas tres loin de la phrase originale,

Par ailleur nous avec aussi faire une comparaison global entre les voisinages produits avec

phrase	vosinage	mot commun
1	0	8
	1	2
	2	2
	3	2
	4	2

Table 4.3 – Présentation du nombre de mot en commun et la phrase 2 original

XPELLS et les voisinages apres changement de la methode de production .pour une comparaison globlale nous avons utilisé 99 phrases et chacune de ses phrase à 99 voisinages et nous avons obtenu les resultats suivants au niveau des distance pour chacun model.

les distances pour le model XPELLS

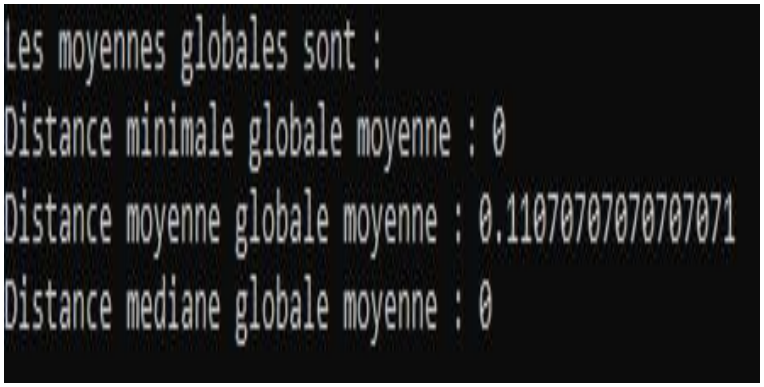

Figure 4.3 – les differents distances aprés excécusion du modél xpells

Aprés exécution nous constatons que les differentes distances ont les résultat suivant;la **Distance minimale globale moyenne : 0** Son objectif etant de quantifier le nombres moyen de mot commun du plus pétit nombre de mot en commun entre toutes les phrases du model xpells et leur voisinage .nous pouvons conclure que dans l'ensemple toutes les phrases et leurs voisinages n'ont pas vraiment de mot en commun car la **Distance mediane globale moyenne : 0** et celui de la **Distance moyenne globale moyenne : 0.11070707070707071** des valeurs nom sigiicatif pour le nombre de mot en commun **les distances pour le model apres changement de la methode de production**

Figure 4.4 – les differents distances aprés excécusion du modél apres changement de la methode de production des voisinage

aprés exécution nous constatons que **la distance minimale globale moyenne est de 0,12** Son objectif etant de quantifier le nombres moyen de mot commun du plus pétit nombre de mot en commun entre toutes les phrases du model et leur voisinage . Avec la distance minimale moyenne different de zero nous somme convaincu que entre chaque phrase et son vosinage on reNcontre au moins un mots en commun . Ensuite nous avons **la distance moyenne globale egale à 0,9567** nous constatons est plus élevé par rapport au modele de xpells donc avec le changement de la maniere de produire le voisinage on enregistre plus de mot en commun entre les les phrases et leur voisinage; Et enfin nous avons la **Distance mediane globale moyenne : 0,8** differenent 0 et avec cette distance nous avons la certitude que le nombre de mot en commun entre les phrase et leurs voisinages sont élévés

4.6 Conclusion

Dans ce chapitre, nous avons présenté le jeu de données utilisé ainsi que les métriques d'évaluations d'une part. Par la suite, nous avons présenté le protocole expérimental et les résultats de nos expérimentations d'autre part. Au vu de ces résultats, nous pouvons conclure que nos approches sont encourageantes pour la génération des explications contrefactuelles. Toutefois, nous avons remarqué que, les contrefactuelles ne sont pas autant générer que les factuelles.

CHAPITRE 5

CONCLUSION GÉNÉRALE

La génération des contrefactuelles vise à faire un examen de conscience aux humains. Elle a connu de grands succès dans la génération des articles. Elle se fait à l'aide des algorithmes d'apprentissage automatique qui se basent sur un ensemble de textes, images et d'autres. L'un des algorithmes utilisés est le celui de xpells. Il a été introduit pour la génération des contrefactuels avec les courtes phrases. Un travail dans la littérature a montré que la méthode présentant de meilleurs résultats était celle qui produit les contractuelles en utilisant les autoencodeurs variationnels, cependant Cette dernière souffre du problème de validité et de sparsité dans les sorties des contrefactuelles générées. Au vu de cette limite, nous avons proposé des approches de solution pour résoudre les dit problèmes. l'étude a été l'analyse du code où nous avons modifié la fonction génération automatique des voisinages avec les autoencodeurs variationnelles en introduisant la métric de similarité et les différentes métrics de distances afin que la contrefactuelle qui en découle soit proche de la phrase originale.

Après avoir eu des contrefactuelles valide avec la solution proposée, nous constatons que la génération des contrefactuelles produire est d'une fréquence très faible par rapport aux contrefactuelles qui en découle et dans la suite, nous concentrerons sur comment faire pour avoir un taux de contrefactuelle plus élevé.

BIBLIOGRAPHIE

[1] Victor Guyomard, Françoise Fessant, Tassadit Bouadi, and Thomas Guyet. Générer des explications contrefactuelles à l'aide d'un autoencodeur supervisé. In *Extraction et Gestion des Connaissances, EGC'2022*, 2022.

[2] Laura et Guidotti Riccardo et Ruggieri Salvatore Lampridis, Orestis et State. Expliquer la classification de textes courts avec divers exemples et contre-exemples synthétiques. *Apprentissage automatique*, pages 1–34.

[3] Audrey Moffett. *Biologie de Reproduction Comparée des Anguilles de L'atlantique en Déclin*. PhD thesis, Institut National de la Recherche Scientifique (Canada), 2019.

[4] Amit et Tan Chenhao Mothilal, Ramaravind K et Sharma. Expliquer les classificateurs d'apprentissage automatique à travers diverses explications contrefactuelles. In *Actes de la conférence 2020 sur l'équité, la responsabilité et la transparence*, pages 607–617.

[5] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 607–617, 2020.

[6] John et Hines Keegan Verma, Sahil et Dickerson. Explications contrefactuelles pour l'apprentissage automatique : les défis revisités. *préimpression arXiv arXiv :2106.07756*.

[7] Brent et Russell Chris Wachter, Sandra et Mittelstadt. Explications contrefactuelles sans ouvrir la boîte noire : Les décisions automatisées et le rgpd. *Harv. JL & Tech.*, 31 :841.