UNIVERSITÉ DU QUÉBEC À MONTRÉAL

APPLICATION DES FONCTIONS D'INFLUENCE EN TRAITEMENT AUTOMATIQUE DU LANGAGE

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN INFORMATIQUE

PAR

FANNY RANCOURT

JANVIER 2024

REMERCIEMENTS

Malgré les défis, ce retour aux études m'aura donné beaucoup et c'est grâce à toutes ces personnes si je peux enfin m'écrier « J'ai triomphé !! »

Tout d'abord, je tiens à remercier ma directrice de recherche, Marie-Jean Meurs, pour m'avoir donné la liberté d'explorer mes intérêts et de m'avoir appuyée dans mon développement. Son soutien académique et financier ont été d'une grande aide pour compléter ces travaux. De plus, je tiens à souligner le support de mon collègue Diego Maupomé qui aura été mon « *ride or die* » tout au long de mon parcours à l'UQAM. Sans lui, mes travaux auraient été bien drabes.

Ces travaux n'auraient probablement pas aboutis sans le soutien émotionnel de mon entourage, en particulier mon conjoint Frédéric Branchaud-Charron et les copain.ines du laboratoire. J'aimerais aussi souligner les magnifiques personnes impliquées dans le groupe étudiant ElleCode&STIM qui promeuvent la diversité en sciences. Vous m'avez donné une communauté dans laquelle m'impliquer et m'avez permis de développer des compétences générales qu'on est si peu amené à développer aux études supérieures. Enfin, je tiens à souligner les bourses d'excellence d'Hydro-Québec et de Louise-Laforest qui ont adouci mon parcours.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES ABRÉVIATIONS, DES SIGLES ET DES ACRONYMES

ACL *Association for Computational Linguistics*.

AI Artificial Intelligence.

Cos-E Commonsense Explanation.

FI fonctions d'influence.

GPT generative pre-trained Transformer.

GRU *gated recurent unit*.

IA intelligence artificielle.

IAX intelligence artificielle explicable.

IF Influence Functions.

IHM interaction humain-machine.

iid indépendant et identiquement distribué.

LLM Large Language Model.

LSTM *long short-term memory*.

MCO moindres carrés ordinaires.

NIST *National Institute of Standards and Technology*.

NLI Natural Language Inference.

NLP Natural Language Processing.

OCDE Organisation de coopération et de développement économiques.

RGPD Règlement général sur la protection des données.

RIT reconnaissance d'implications textuelles.

RN réseaux de neurones.

RNR réseaux de neurones récurrents.

T5 text-to-text transfer Transformer.

TALN traitement automatique du langage naturel.

UE Union européenne.

XAI explainable IA.

RÉSUMÉ

Les récents progrès en traitement automatique du langage naturel (TALN) auront permis à cette technologie de se répandre dans de nombreux domaines d'application tels que la traduction automatique et l'automatisation de processus organisationnels. Toutefois, cette adoption demande des clarifications quant à la responsabilité ainsi que la fiabilité des systèmes fondés sur l'intelligence artificielle (IA) afin que ces derniers soient dignes de confiance. Pour s'attaquer à ces enjeux, plusieurs organisations telles que le Parlement européen et le *National Institute of Standards and Technology* (NIST) préconisent l'analyse de ces systèmes à l'aide des principes pour une IA digne de confiance, tels la responsabilité, la transparence et l'explicabilité, puis la robustesse, la securité et la sûreté. Néanmoins, l'opérationalisation de ces principes et les exigences associées sont extrêmement difficiles à formuler. C'est particulièrement le cas pour l'explicabilité, la capacité d'expliquer le comportement d'un système fondé sur l'IA à un humain. L'IA étant une technologie guidée par les données, identifier les observations influençant une prédiction est l'une des principales approches en explicabilité. Ceci peut notamment être fait à l'aide des fonctions d'influence (FI), une technique issue des statistiques robustes.

Ce mémoire présente des travaux originaux traitant de l'application des FI pour des problèmes de classification en TALN. Premièrement, on analyse comment les FI peuvent informer sur la qualité de l'annotation. Bien que de moindre qualité que les annotations par des expert.es, celles impliquant la production participative d'un large public non-spécialiste sont précieuses, particulièrement lorsque peu de données sont disponibles. De plus, l'IA générative, qui peut produire des textes en langage naturel, permet de classifier ainsi que de justifier la prédiction. Toutefois, nos expériences ont révélé des limites importantes quant aux justifications lorsqu'on considère des architectures de moindres tailles. Notamment, les justifications textuelles non structurées tendent à suivre certains gabarits, même si ces derniers sont peu présents dans les données. Bien qu'il est intuitif que ces quelques exemples utilisant ces gabarits soit influents, les FI n'abondent pas en ce sens.

Mots clés : fonctions d'influence, explication post-hoc, explicabilité, interprétabilité, traitement automatique du langage naturel.

INTRODUCTION

Le traitement automatique du langage naturel (TALN) est un domaine de recherche interdisciplinaire qui s'intéresse au traitement et à la manipulation du langage humain par des ordinateurs. Les dernières années ont vu une hausse importante de l'intérêt pour cette discipline et ses applications. En effet, les conférences de premier plan telles que celles organisées par l'*Association for Computational Linguistics* (ACL) observent des records autant sur les plans de la quantité de soumissions reçues (Synced, 2019) que de la participation aux évènements (Rasmussen, 2021). Plus récemment, le maintenant célèbre agent conversationnel ChatGPT (Ouyang *et al.*, 2022), un assistant virtuel permettant les échanges en langage naturel et ayant le potentiel de résoudre de nombreuses tâches (telles que Choi *et al.*, 2023; Frieder *et al.*, 2023; Jiao *et al.*, 2023), a facilité la démocratisation des dernières avancées en TALN à l'ensemble de la population (De Angelis *et al.*, 2023).

Toutefois, l'utilisation de systèmes fondés sur l'intelligence artificielle (IA) en pratique est accompagnée d'enjeux de taille tels que le biais d'automatisation et un niveau de confiance inadéquat envers ces systèmes (Kocielnik *et al.*, 2019; Alon-Barkat et Busuioc, 2023), le besoin accru de transparence quant à l'origine et le fonctionnement du système (Arnold *et al.*, 2019; Mitchell *et al.*, 2019), ainsi que la nécessité de contester les décisions générées (Lyons *et al.*, 2021; Alfrink *et al.*, 2022). De plus, les cadres réglementaires sont mal adaptés pour assurer que les systèmes fondés sur l'IA sont dignes de confiance. Les travaux entamés par le parlement européen (Commission européenne, 2021) ainsi que le *National Institute of Standards and Technology* (NIST) (Tabassi, 2023) tentent d'apporter les ajustements nécessaires pour protéger le public des effets néfastes de cette technologie. Les principaux piliers sont des exigences relatives au niveau de risque d'un système ainsi que l'intégration des principes favorisant une IA « digne de confiance et qui respecte les droits de l'homme et les valeurs démocratiques » (Organisation de coopération et de développement économiques, 2019a). Parmi ces principes, on retrouve la robustesse, la transparence et l'explicabilité. Ce dernier cherche à expliquer le comportement de l'IA à un humain. Malheureusement, comprendre les solutions basées sur l'IA n'est pas une tâche simple. Premièrement, les systèmes sont géné-

ralement d'une complexité inouïe et coordonnent un ensemble de sous-systèmes eux-mêmes complexes. De plus, les solutions basées sur l'IA n'étant pas déterministes, c'est-à-dire que chaque nouvelle solution créée à l'aide du même protocole ne sera pas nécessairement la même, comprendre leurs comportements est une tâche extrêmement difficile (Wallace *et al.*, 2019; Ribeiro *et al.*, 2020; Gauthier-Melançon *et al.*, 2022).

Afin de rendre les solutions basées sur l'IA compréhensibles pour des humains, plusieurs approches ont été développées. On peut notamment penser aux cartes de saillance (Simonyan *et al.*, 2014; Ribeiro *et al.*, 2016; Lundberg et Lee, 2017), qui attribuent un poids à chacun des attributs d'une observation. Dans le cas du TALN, ces attributs peuvent être des mots composant le texte à traiter. Une autre approche est d'attribuer un poids aux exemples modélisés par le système. Ceci permet d'expliquer pourquoi on peut observer certains motifs dans le traitement des données (Han *et al.*, 2020). Pour cette approche, la méthode la plus populaire est les fonctions d'influence (FI) (Cook et Weisberg, 1982; Koh et Liang, 2017), une technique issue des statistiques robustes. Celle-ci permet d'évaluer efficacement l'effet de modifier les données à modéliser par le système. Les travaux présentés dans ce mémoire s'intéressent **aux FI et comment elles peuvent fournir de l'information pertinente sur le comportement d'un système en TALN**. En particulier, on étudie l'influence de la qualité de l'annotation à l'aide d'observation annotées par des expert.es du domaine d'application ainsi que des personnes non-spécialisées. Enfin, on analyse certains artéfacts des modèles autojustifiants, c'est-à-dire qui prédisent conjointement une décision ainsi que son explication, à l'aide des FI.

Ce manuscrit est structuré comme suit. Le chapitre 1 présente les bases en TALN et résume des travaux récents en intelligence artificielle explicable (IAX). Ces derniers visent à améliorer la compréhension des systèmes basés sur l'IA par des humains. Ensuite, le chapitre 2 présente la théorie sur laquelle reposent les FI dans le contexte classique ainsi qu'en IA. On conclut ce chapitre sur une discussion sur les limites des algorithmes actuellement disponibles pour le calcul des FI. Le travail de recherche présenté dans ce mémoire a été publié sous la forme de deux articles scientifiques revus par les pairs. Le premier article

Rancourt, F., Maupomé, D., Meurs, M.-J. (2022). *On the Influence of Annotation Quality in Suicidal Risk Assessment from Text.* Proceedings of the 35th Canadian Conference on Artificial Intelligence.

est présenté dans le chapitre 3, le second

Rancourt, F., Vondrlik, P., Maupomé, D., Meurs, M.-J. (2023). *Investigating Self-Rationalizing Models for Commonsense Reasoning.* Stats 6(3) : 907–919.

dans le chapitre 4.

En plus des travaux mentionnés ci-haut, mes travaux de maîtrise ont mené aux contributions scientifiques suivantes :

— Maupomé, D., **Rancourt, F.**, Belbahar, R., Meurs, M.-J. (2023). *MentalHealthBERT : A Pretrained Language Model for Mental Health Risk Detection.* Soumis.

— Maupomé, D., **Rancourt, F.**, Soulas, T., Lachance, A., Meurs, M.-J., Aleksandrova, D., Brochu Dufour, O., Pontes, I., Cardon, R., Simard, M., Vajjala, S. (2022). *Automatic Text Simplification of News Articles in the Context of Public Broadcasting.* arXiv preprint arXiv :2212.13317.

— Saravani, S. H. H., Normand, L., Maupomé, D., **Rancourt, F.**, Soulas, T., Besharati, S., Normand, A., Mosser, S., Meurs, M.-J. (2022). *Measuring the severity of the signs of eating disorders using similarity-based models.* CLEF 2022 Conference and Labs of the Evaluation Forum. eRisk Shared Task.

— Maupomé, D., Armstrong, M. D., Belbahar, R., Alezot. J., Balassiano, R., **Rancourt, F.**, Queudot, M., Mosser, S., Meurs, M.-J. (2022). *Automatically Estimating the Severity of Multiple Symptoms Associated with Depression.* Dans Early Detection of Mental Health Disorders by Social Media Monitoring. Studies in Computational Intelligence, vol 1018. Springer.

— Maupomé, D., Armstrong, M. D., **Rancourt, F.**, Soulas, T., Meurs, M.-J. (2021). *Early Detection of Signs of Pathological Gambling, Self-Harm and Depression through Topic Extraction and Neural Networks.* CLEF 2021 Conference and Labs of the Evaluation Forum. eRisk Shared Task.

— Maupomé, D., **Rancourt, F.**, Armstrong, M. D., Meurs, M.-J. (2021). *Position Encoding Schemes for Linear Aggregation of Word Sequences.* Proceedings of the 34th Canadian Conference on Artificial Intelligence.

— Maupomé, D., Armstrong, M. D., **Rancourt, F.**, Meurs, M.-J. (2021). *Leveraging Textual Similarity to Predict Beck Depression Inventory Answers.* Proceedings of the 34th Canadian Conference on Artificial Intelligence.

CHAPITRE 1
NOTIONS PRÉLIMINAIRES

À l'heure actuelle, les approches en TALN à l'état de l'art sont presque toutes basées sur des réseaux de neurones (RN). Afin de bien les saisir, on présente les bases de l'apprentissage automatique, la branche de l'IA s'intéressant aux approches permettant l'apprentissage de modèles à partir de données. Dans ce mémoire, on appelle « modèle » une fonction qui modélise un ensemble de données. En contrepartie, on parle d'« architecture » lorsqu'on se réfère à la famille de modèles Enfin, les RN étant difficile à comprendre pour les humains interagissant avec ceux-ci, on fait un survol des travaux récents visant à améliorer l'interprétabilité en apprentissage automatique.

Les principales sources sur lesquelles ce chapitre s'appuie sont citées ci-dessous. La section 1.1 présente des notions de base des RN et est inspirée de Goodfellow *et al.* (2016). Ensuite, on aborde le TALN moderne, des réseaux de neurones récurrents (RNR) aux transformeurs, dans la section 1.2. L'organisation de cette section est inspirée de Tunstall *et al.* (2022). Enfin, on conclut ce chapitre avec la présentation de diverses initiatives visant à rehausser l'interprétabilité des modèles neuronaux, un domaine appelé IAX. Cette discipline étant plus récente et moins bien définie (Krishnan, 2020), il n'existe pas à ce jour de référence aussi complète que pour les sujets précédents. Néanmoins, les travaux de Ras *et al.* (2021) et Molnar (2022) dressent un portait assez complets des techniques d'explicabilité.

1.1 Apprentissage automatique

Pour résoudre un problème à l'aide de l'apprentissage automatique, il y a quelques prérequis. Premièrement, il faut modéliser le problème étudier pour identifier la tâche qui est la mieux adaptée. Ensuite, il est nécessaire d'avoir accès à des données, qu'on appelle aussi observations, qu'on peut manipuler et desquelles on peut apprendre. En TALN, on utilise des contenus textuels tels que des articles de journaux et des textes de blog. Deux tâches classiques en apprentissage automatique sont la classification, la catégorisation d'observations

sachant un ensemble de classes prédéfinies, et la régression, l'apprentissage de la fonction qui représente le mieux un ensemble d'observations numériques. Celles-ci sont toutes deux des tâches dites supervisées comme elles tentent de prédire une cible donnée. Une approche populaire est de tirer parti des attributs. Ces derniers sont des propriétés mesurables ou non à utiliser notamment lors de l'exécution d'une tâche de apprentissage automatique. Par exemple, pour classifier un texte, on pourrait considérer chacun des mots comme des attributs. De plus, une tâche est dite non supervisée lorsqu'elle ne considère pas de cible lors de l'apprentissage. Par exemple, le partitionnement de données, qui tente de diviser les données en groupes homogènes, et les autoencodeurs, qui cherchent une représentation expressive en compressant puis reconstruisant les données, figurent parmi les tâches les plus communes de ce type d'apprentissage.

Une fois la tâche identifiée, il est nécessaire d'avoir des exemples desquels le modèle pourra apprendre. Pour que le modèle puisse apprendre dans des conditions optimales, il est typiquement nécessaire d'avoir une grande quantité d'exemples variés et de bonne qualité. La diversité est essentielle afin que le modèle puisse saisir les nuances qu'on retrouve dans la nature. La notion de qualité est spécifique à la tâche considérée, mais considère généralement la présence de bruit qui corrompt le signal que le modèle doit apprendre. Par exemple, en classification de textes, on dira qu'un ensemble de données est de qualité lorsque les étiquettes, c'est-à-dire les annotations identifiant la classe à laquelle une observation appartient, sont fiables. Ceci est mesuré avec l'accord de plusieurs annotations. La collecte de données de qualité étant coûteuse en temps et en argent, il est de coutume dans la littérature d'utiliser des ressources pré-existantes.

Fondamentalement, les modèles de apprentissage automatique sont des problèmes d'optimisation mathématique. En effet, ceux-ci tentent d'apprendre à l'aide d'un grand nombre d'exemples la meilleure combinaison de paramètres. Ceci est déterminé à l'aide de la fonction de perte qui mesure la performance sachant un ensemble de données et des paramètres $\theta \in \mathbb{R}^p$. Plus formellement, pour $\mathcal{L}(\cdot, \theta)$ une fonction de perte et un ensemble d'observations

(x_i, y_i) avec $i \in \{1, \ldots, n\}$, on appelle la fonction de perte empirique

$$\widehat{\mathcal{L}}(\theta) \overset{\text{déf}}{=} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(x_i, y_i, \theta).$$

C'est cette quantité que le solveur minimise. De plus, les paramètres optimaux sont $\hat{\theta} \overset{\text{déf}}{=} \arg\min_\theta \widehat{\mathcal{L}}(\theta)$. [1] Ainsi, le choix de la fonction de perte est critique : deux fonctions distinctes peuvent mener à des modèles très différents. Typiquement, la fonction de perte est une mesure liée à l'erreur de la prédiction. Par exemple, en classification, il est commun de considérer l'entropie croisée. Celle-ci cherche à maximiser la probabilité que la classe prédite par le modèle $\hat{y}$ soit la bonne classe.

Bien que les fonctions de perte puissent être utiles pour la comparaison de plusieurs modèles, il est rare qu'on l'utilise en pratique pour évaluer la qualité d'un modèle. De plus, il est commun de considérer un ensemble de mesures afin d'obtenir un portrait de chacun des modèles. Considérons une tâche de classification binaire, c'est-à-dire qu'on doit distinguer une classe de l'autre. Posons VP les vrais positifs, FP les faux positifs, VN les vrais négatifs, et FN les faux négatifs. On considère habituellement les métriques suivantes :

- l'exactitude mesure la proportion de prédictions correctes, c'est-à-dire $\frac{VP+VN}{VP+FP+VN+FN}$,
- la précision capte la proportion d'items pertinents parmi ceux qui ont été attribués à la classe j, c'est-à-dire $\frac{VP}{VP+FP}$,
- le rappel mesure la proportion d'éléments correctement attribués à la classe j parmi ceux appartenant à cette classe, c'est-à-dire $\frac{VP}{VP+FN}$,
- la F-mesure est la moyenne harmonique de la précision et du rappel, c'est-à-dire $\frac{2VP}{2VP_FP+FN}$.

Les métriques ci-haut peuvent aussi être considérées pour des tâches de classifications à plus de deux classes moyennant des ajustements mineures. De plus, selon le contexte d'application, certaines métriques peuvent être plus importantes que d'autres. Par exemple, en détection

1. Puisque $\hat{\theta}$ est minimal, on a $0 = \nabla_\theta \widehat{\mathcal{L}}(\theta)\Big|_{\theta=\hat{\theta}}$. De plus, la matrice hessienne empirique $\hat{H}_{\hat{\theta}} \overset{\text{déf}}{=} \nabla^2_\theta \widehat{\mathcal{L}}(\theta)\Big|_{\theta=\hat{\theta}}$ est définie positive par définition. Ainsi, $\hat{H}^{-1}_{\hat{\theta}}$ existe. L'existence de l'inverse est critique pour les travaux présentés au chapitre 2.

de risque de la santé mentale, le coût de ne pas détecter un cas est grand, alors un modèle avec un grand rappel est plus adapté.

Comme le but ultime de la plupart des solutions en apprentissage automatique est la généralisation, il est coutume de réserver un sous-ensemble des données à l'évaluation. On appelle ce sous-ensemble les données de test. Ainsi, on compare l'ensemble des modèles à l'étude en se basant sur les résultats obtenus sur les données de test. Pour que les résultats puissent être indicatifs de la capacité de généralisation, il est critique que la distribution des données de test représente celle de la population. Une autre approche est d'effectuer une validation croisée à k blocs. Après avoir partitionné les données en k groupes, un est réservé pour l'évaluation et le reste est utilisé lors de l'apprentissage. Cette procédure est répétée jusqu'à ce que tous les blocs aient servi à l'évaluation. Enfin, on agrège les résultats des k modèles avec la moyenne et la variance.

Examinons de plus près les RN à propagation avant, une architecture fondamentale en apprentissage automatique. Il s'agit de réseaux prenant la forme d'un graphe orienté acyclique dont les sommets sont organisés en groupes non connexes, qu'on appelle les couches. Les réseaux récurrents (sous-section 1.2.1) sont une architecture où le graphe n'est pas acyclique. On discute premièrement du réseau à une couche, le perceptron. Ensuite, on présente les réseaux à propagation avant ayant plusieurs couches.

1.1.1 Perceptron

Le perceptron (Rosenblatt, 1958) permet de modéliser les données binaires linéairement séparables. Il s'agit du modèle à propagation avant le plus simple comme il est composé d'exactement une couche. Plus formellement, pour une observation $x \in \mathbb{R}^m$ et $y \in \{0, 1\}$ sa classe, le perceptron f peut être décrit comme suit :

$$\eta(x) = \phi(w^T x + b) = \begin{cases} 1 & \text{si } \ w^T x + b > 0, \\ 0 & \text{sinon,} \end{cases} \tag{1.1}$$

où $w \in \mathbb{R}^m$ est le vecteur des poids appris et $b \in \mathbb{R}$ le biais. D'où, $\hat{\theta} = (w, b)$.

Ainsi, le vecteur des poids permet d'accorder plus d'importance à certains attributs de l'observation. Les poids et le biais sont appelés les paramètres du modèle. La fonction ϕ est appelée la fonction d'activation. On considère l'erreur quadratique $(y - \eta(x))^2$ comme fonction de perte.

En apprentissage automatique, il est coutume d'utiliser l'algorithme de la descente des gradients (Cauchy, 1847) pour effectuer l'apprentissage. [2] Sommairement, la mise à jour des paramètres se fait suivant l'opposé de la direction du gradient, la dérivée multidimensionnelle. C'est dans cette direction que la pente est la plus descendante. La longueur du pas utilisé pour la mise à jour est appelé le taux d'apprentissage. L'algorithme arrête lorsque la mise à jour des poids est négligeable, c'est-à-dire lorsque le gradient approche 0. Une limite importante de cet algorithme est qu'il n'est pas garanti d'obtenir un minimum global. Ainsi, les paramètres déduits ne sont pas nécessairement les bons.

Toutefois, la fonction de perte n'est pas dérivable comme la fonction d'activation en (1.1) ne l'est pas. C'est pourquoi il est courant d'utiliser l'approximation dérivable suivante :

$$\phi(t) = \frac{1}{1 + e^{-t}}$$

qu'on appelle la fonction logistique. On peut se référer à la figure 1.1 pour une comparaison des deux fonctions d'activation.

1.1.2 Modèles multicouches

En raison de la simplicité de son architecture, le perceptron est limité dans sa capacité de modélisation. Toutefois, en ajoutant des couches, il est possible de modéliser des données de plus en plus complexes.

2. On utilise « apprentissage » et « entraînement » de façon interchangeable dans ce mémoire.

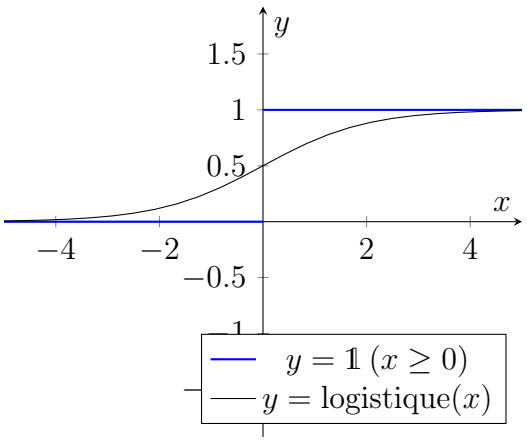


Figure 1.1 – Les fonctions indicatrice et logistique sont des fonctions d'activation

Considérons le modèle illustré à la figure 1.2. Celui-ci est doté de deux couches cachées, c'est-à-dire des couches dont la sortie est utilisée en entrée de la couche suivante. Ainsi, un modèle multicouche est une composition de fonctions. Plus formellement, la couche de sortie est $\eta(x) = \eta_3(\ell_2)$ où $\ell_2 = \eta_2(\eta_1(x))$ et $\eta_i(\cdot)$ donne la sortie ℓ_i de la i^{e} couche.

Supposons que la fonction de perte empirique est dérivable. Alors, on peut utiliser la descente de gradients (ou une de ses variantes) pour effectuer l'entraînement. La mise à jour des poids se fait avec la rétropropagation. Cet algorithme utilise la dérivation en chaîne pour propager le gradient à travers chacune des couches de la sortie jusqu'à l'entrée. Plus formellement, pour le modèle à 2 couches cachées, on trouve

$$\begin{aligned}
\nabla_{\ell_2} \mathcal{L} &= \nabla_y \mathcal{L} \frac{\partial y}{\partial \ell_2}, \\
\nabla_{\ell_1} \mathcal{L} &= \nabla_{\ell_2} \mathcal{L} \frac{\partial \ell_2}{\partial \ell_1}, \\
\nabla_x \mathcal{L} &= \nabla_{\ell_1} \mathcal{L} \frac{\partial \ell_1}{\partial x}.
\end{aligned}$$

Ceci est illustré à la figure 1.3.

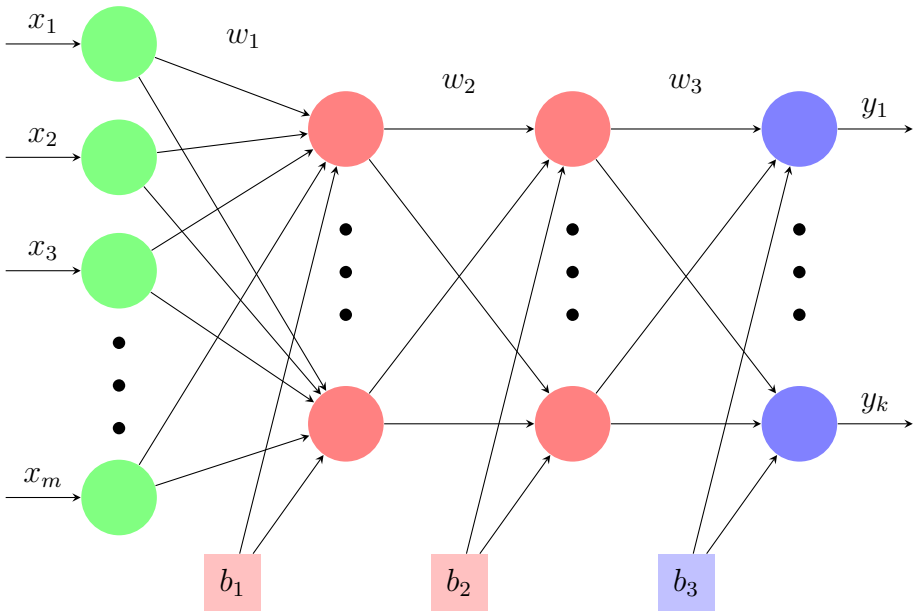


Figure 1.2 – Illustration d'un modèle à propagation avant avec 2 couches cachées (en rouge). La couche d'entrée est en vert et la couche de sortie est en bleu. Chaque couche est dotée d'un vecteur de poids w_i et d'un biais b_i.

1.2 Réseaux de neurones en traitement automatique du langage naturel

La langue suivant un certain nombre de règles, certaines considérations sont essentielles pour la traiter adéquatement. En effet, une phrase n'est pas une combinaison arbitraire de mots : leur ordre a de l'importance, un mot est dépendant des autres, et certaines combinaisons de mots ne vont pas ensemble. Par exemple, dans la phrase « le printemps est arrivé », le déterminant « le » dépend du nom « printemps » puis le verbe « est arrivé » dépend du sujet « le printemps ». De plus, « le printemps est arrivé » est une séquence différente de « arrivé est le printemps ». Ainsi, l'approche naïve codant la présence ou l'absence d'un mot pose d'importantes limites.

Le TALN statistique repose sur la notion de modèle de langue qui décrit une distribution de probabilité sur ses séquences de mots. Historiquement [3], on analysait les groupes de n mots consécutifs dans un texte, qu'on appelle n-grammes. Toutefois, il existe plus d'une formulation pour communiquer la même chose et cette approche n'injecte pas en amont de notion de

3. Se référer à Manning et Schutze (1999) et Rosenfeld (2000) pour un survol des fondements du TALN.

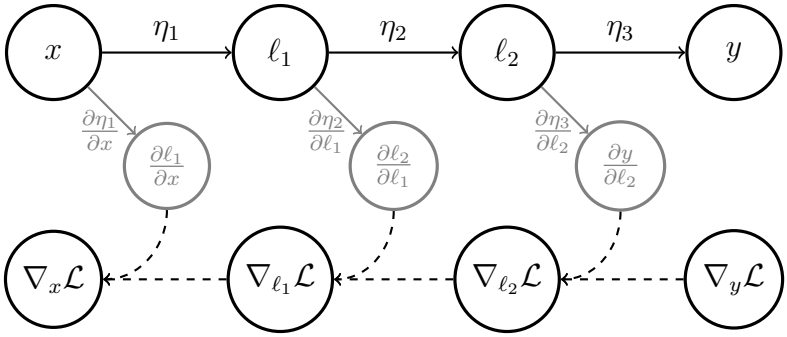


Figure 1.3 – Illustration simplifiée de la rétropropagation

proximité sémantique entre ces différents n-grammes. Plus récemment, les plongements de mots statiques (Mikolov *et al.*, 2013; Pennington *et al.*, 2014) ont permis d'associer un mot à une représentation dans l'espace apprise à partir de leurs co-occurences dans des corpora de documents. Ceci a pour effet de produire des vecteurs qui reflètent certaines propriétés sémantiques. Ainsi, pour un mot donné, son vecteur est construit de sorte qu'il soit proche de ses synonymes et des ses déclinaisons. Toutefois, il s'agit d'une représentation statique, c'est-à-dire que le vecteur associé à un mot sera le même pour toutes ses occurrences. Ceci est particulièrement problématique pour les homographes, qui peuvent avoir plus d'un sens mais qu'une représentation. En effet, c'est avec le contexte, les mots entourant l'homographe, qu'il est possible d'en saisir le sens réel.

Ce problème peut être abordé en s'appuyant sur des modèles produisant des plongements de mots qui prennent le contexte en considération. Les modèles récurrents (sous-section 1.2.1) et les transformeurs (sous-section 1.2.2) peuvent être déployés à ces fins.

1.2.1 Modèles récurrents

Les réseaux récurrents permettent le traitement de séquences, où les éléments sont dépendants les uns des autres et l'ordre des éléments n'est pas arbitraire. Plus formellement, pour

une séquence $x_1, \ldots, x_t, \ldots, x_T$, un réseau récurrent est défini comme suit :

$$h_0 = 0$$
$$h_t = \eta(h_{t-1}, x_t)$$

où h_{t-1} est l'état caché, c'est-à-dire l'information que le modèle conserve de la sous-séquence $x_1, \ldots, x_{t-1}$, et η est la fonction produisant le nouvel état caché. Ainsi, le modèle réutilise l'information qu'il a précédemment extraite pour le traitement du prochain élément. Par exemple, considérons la phrase « Le printemps est arrivé. » Lorsque le modèle traitera le mot « est », il le fera à l'aide de l'information extraite de la sous-séquence « Le printemps. » Ainsi, chaque mot est traité par le réseau dans l'ordre et sachant l'information précédemment traitée. On peut remarquer que le contexte conservé en mémoire ne considère que le « passé ». Pour s'attaquer à cet enjeu, on peut considérer les modèles bidirectionnels qui combinent les balayages avant et arrière de la séquence (tel que Graves et Schmidhuber, 2005).

Des enjeux notables des modèles récurrents sont la disparition et l'explosion du gradient, soit lorsque le gradient devient tellement petit (resp. grand) que la rétropropagation ne peut plus ajuster efficacement les poids du modèle (Goodfellow *et al.*, 2016). La disparition du gradient se produit lorsque ce dernier décroît continuellement jusqu'à devenir insignifiant et l'apprentissage est arrêté. À l'inverse, l'explosion du gradient produit des modèles instables où les poids changent drastiquement à chaque rétropropagation. En limitant la fenêtre de dépendance, les modèles *long short-term memory* (LSTM) (Hochreiter et Schmidhuber, 1997) et *gated recurent unit* (GRU) (Cho *et al.*, 2014) ont obtenu beaucoup de succès en TALN. Toutefois, limiter la mémoire du contexte pose aussi des contraintes.

1.2.2 Transformeurs

Les transformeurs ont été introduits par Vaswani *et al.* (2017) et permettent d'obtenir une représentation des mots sachant le contexte global. Lorsqu'utilisée avec des réseaux récur-

rents, la représentation interne est typiquement [4] un seul vecteur compressant l'ensemble du texte en entrée et les premiers mots d'une longue séquence sont « oubliés » (Tunstall *et al.*, 2022). Les transformeurs résolvent ce problème en effectuant un traitement qui n'est pas séquentiel et ne limitent pas la taille de la représentation.

Le succès de ce type de modèles repose sur 3 éléments essentiels : le couple encodeur-décodeur, l'attention et le pré-entraînement. Ces éléments ont fait leurs preuves avec les réseaux récurrents avant d'être combinés pour obtenir les transformeurs qu'on connaît aujourd'hui.

1.2.2.1 Encodeur et décodeur

L'encodeur transforme les mots du texte en entrée en vecteurs numériques d'attributs qu'on appelle la représentation interne. À partir de la représentation interne, le décodeur peut générer séquentiellement le texte en sortie. L'encodeur et le décodeur comportent chacun un certain nombre de couches permettant de travailler avec des attributs complexes.

1.2.2.2 Attention

Afin d'intégrer le rôle d'un mot dans une séquence dans sa représentation, les transformeurs utilisent le mécanisme d'attention. À tour de rôle, ce mécanisme «portera attention» sur tous les autres mots individuellement. Chaque paire de mots ainsi construite produira une valeur de saillance par l'intermédiaire de paramètres appris. Cette saillance cherche à modéliser la pertinence [5] du mot cible par rapport à celui qui lui porte attention. Enfin, puisque ces saillances sont calculées deux à deux sans tenir compte d'autres saillances, elles sont normalisées. Ce processus est répété plusieurs fois pour permettre plusieurs remises en contexte des représentations de mots. La figure 1.4 illustre une itération de ce processus. De plus, afin de permettre à une même paire de mots d'interagir de manière différente sur divers aspects, ce processus peut être réalisé de façon parallèle avec différents groupes de paramètres, les têtes

4. Le modèle ELMo (Peters *et al.*, 2018), qui utilise des LSTM bidirectionnels, est un contre-exemple notoire.

5. À noter que cette approche a d'importantes limites pour modéliser l'importance réelle des mots (Jain et Wallace, 2019).

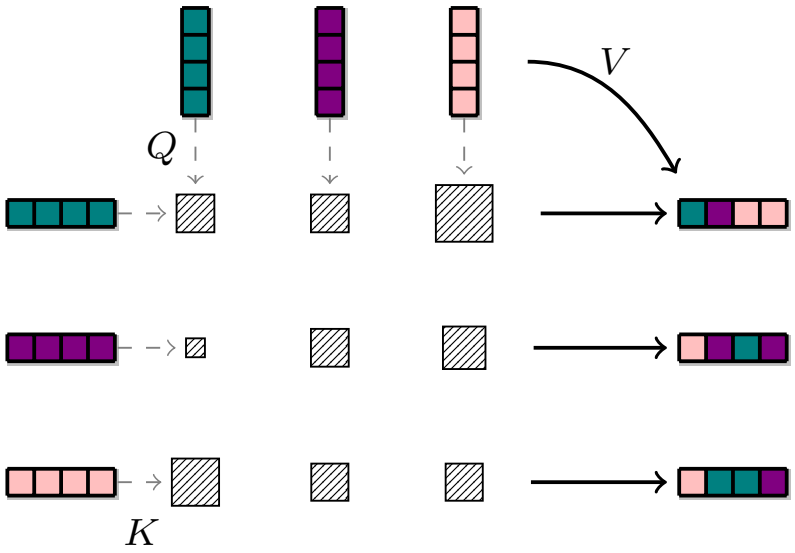


Figure 1.4 – Illustration du fonctionnement de l'attention. Chaque mot Q est comparé un à un aux mots K pour en déduire la saillance V. On obtient les plongements de mots en combinant la saillance et les mots K.

d'attention.

1.2.2.3 Pré-entraînement

En considérant plutôt une tâche fondamentale, le pré-entraînement permet au modèle de faire des apprentissages très généraux sur la langue à modéliser. On considère généralement des corpora textuels larges et non annotés tels que Chelba *et al.* (2013) et C4 (Raffel *et al.*, 2020) pour prédire le prochain mot (Howard et Ruder, 2018; Radford *et al.*, 2018). Cette tâche nécessite l'utilisation d'un encodeur et d'un décodeur. Ces apprentissages, ayant saisi le fonctionnement de la langue, peuvent ensuite être réinvestis pour la résolution de la tâche d'intérêt (Peters *et al.*, 2018). Comme on réutilise les poids du modèle pré-entraîné comme point de départ de l'entraînement (Howard et Ruder, 2018), peu de ressources (itérations d'entraînement et données) suffisent pour obtenir le modèle final. Ce second entraînement est communément appelé l'affinage. Sachant qu'il est très coûteux d'annoter des données, réduire la quantité nécessaires dans le domaine de la tâche à l'étude a permis de nombreuses

avancées. Enfin, un modèle pré-entraîné peut être réutilisé dans de multiples domaines et pour de multiples tâches de TALN moyennant des modifications mineures. Par exemple, les tâches de classification ne considèrent généralement [6] que l'encodeur.

Les travaux présentés au chapitre 3 considèrent le modèle RoBERTa (Liu *et al.*, 2019), une variante robuste de BERT (Devlin *et al.*, 2018). Il s'agit d'un transformeur où l'entraînement de l'attention est bidirectionnel : la tâche considérée lors du pré-entraînement est de prédire un mot masqué entouré de mots visibles. Pour effectuer de la classification de textes avec l'encodeur de RoBERTa, il est nécessaire d'ajouter une ou plusieurs couches neuronales permettant la classification. Les travaux du chapitre 3 utilisent un classifieur linéaire.

Les travaux présentés au chapitre 4 utilisent le modèle Text-to-text transfer Transformer (T5) (Raffel *et al.*, 2020), un modèle génératif composé d'un encodeur et d'un décodeur. En transformant en texte les données en entrée et en sortie, text-to-text transfer Transformer (T5) permet d'attaquer diverses tâches en TALN telles que la traduction, l'analyse de sentiment et la compréhension de texte, à l'aide d'un seul et même modèle. De plus, ce modèle peut être affiné pour résoudre une tâche spécifique telle que la génération d'explications pour une prédiction donnée (Narang *et al.*, 2020).

1.3 Interprétabilité et explications

L'IA est utilisée dans plusieurs domaines ayant d'importants effets sur la vie des gens tels que les recrutements professionnels (Mujtaba et Mahapatra, 2019; Wilson *et al.*, 2021), les admissions universitaires (Kleinberg *et al.*, 2018; Marcinkowski *et al.*, 2020) et les soins de santé (Char *et al.*, 2018; Hoffman, 2021; Maupomé *et al.*, 2021). La complexité des systèmes utilisant de l'IA et plusieurs scandales (tels que Angwin *et al.*, 2016; Isaak et Hanna, 2018), impliquant des systèmes opaques clamant utiliser cette technologie, ont mis de l'avant le besoin accru de transparence. L'Organisation de coopération et de développement économiques (OCDE) définit ce principe selon les axes suivants (Organisation de coopération et de développement économiques, 2019b) :

6. T5 (Raffel *et al.*, 2020) est un contre-exemple notoire.

— « divulguer l'utilisation de l'IA quand elle a lieu »,

— « permettre aux gens de comprendre comment un système d'IA est développé, formé, fonctionne et est déployé dans le domaine d'application concerné »,

— « fournir des informations significatives et clarifier quelles informations sont fournies et pourquoi »,

— « [faciliter] le discours public multipartite et la création d'entités dédiées, si nécessaire, à favoriser la sensibilisation et la compréhension générale des systèmes d'IA afin d'en favoriser l'acceptation et de générer de la confiance ».

Comme les systèmes reposant sur l'IA sont généralement très complexes, des cadres légaux tels que le Règlement général sur la protection des données (RGPD) (Commission européenne, 2016) et le *Blueprint for an Artificial Intelligence (AI) Bill of Rights* (White House Office of Science and Technology Policy, 2022) ont formalisé le *droit à une explication*. Bien que ce droit soit difficile à mettre en place (Selbst et Powles, 2017; Wachter *et al.*, 2017), il a motivé le développement de beaucoup d'approches techniques pour rendre les modèles plus compréhensibles pour les humains (tels que Ribeiro *et al.*, 2016; Lundberg et Lee, 2017; Kim *et al.*, 2018) ainsi que des recherches sur l'interaction IA-humain (tels que Yin *et al.*, 2019; Kocielnik *et al.*, 2019). Ces travaux sont issus de la communauté en IAX, qui s'intéresse aux différentes approches techniques ou socio-techniques visant à augmenter l'interprétabilité d'un modèle. On peut se référer à Jacovi *et al.* (2023) pour une revue des principales approches ainsi que la théorie sur les explications humaines dont elles découlent. Ce mémoire considère les aspects techniques de l'explicabilité.

Néanmoins, il n'y a pas à ce jour de consensus dans la communauté sur ce qu'est l'interprétabilité (Erasmus *et al.*, 2021; Prasetya, 2022; Erasmus et Brunet, 2022). De plus, cette ambiguïté existe tant au niveau conceptuel (non-technique) qu'opérationnel (technique) (Krishnan, 2020). Dans ce mémoire, on utilise la définition de l'OCDE (Organisation de coopération et de développement économiques, 2019b) :

> L'explicabilité signifie permettre aux personnes affectées par le résultat d'un système d'IA de comprendre comment ce résultat a été obtenu. Cela implique de fournir des informations faciles à comprendre pour les personnes affectées par le

résultat d'un système d'IA leur permettant de contester le résultat, notamment - dans la mesure du possible - les facteurs et la logique qui ont conduit à ce résultat.

De plus, un modèle sera dit *interprétable* lorsqu'une personne peut le comprendre intégralement (Lipton, 2018). Les modèles interprétables les plus populaires sont les arbres de décisions, les systèmes à base de règles ainsi que les régressions linéaires et logistiques. On peut se référer à Molnar (2022) pour une présentation plus exhaustive de ceux-ci. Un des principaux avantages de ces architectures est de faciliter la remise en question des décisions. Pour des applications où le risque de préjudice est élevé, Rudin (2019) plaide pour que tous les systèmes utilisant de l'IA soient interprétables.

On appelle *explication* l'action explicite d'expliquer une décision (explication locale) ou un modèle (explication globale) à une personne (Miller, 2019). Ainsi, une explication peut concerner une décision obtenue à l'aide d'un modèle interprétable ou non. Pour un survol des techniques d'explications globales, on peut se référer au Chapitre 8 de Molnar (2022). Une explication locale peut prendre plusieurs formes : il peut notamment s'agir de fournir des attributs, des exemples, des textes en langage naturel, etc. **Le terme explication référera uniquement aux explications locales dans la suite du manuscrit.**

Bien que plusieurs solutions techniques ont été proposées dans la littérature pour fournir des explications pour les modèles d'apprentissage profond (Ras *et al.*, 2021), l'évaluation de leur efficacité et de leur qualité demeure complexe (Doshi-Velez et Kim, 2017). En effet, l'interprétabilité étant mal définie (Krishnan, 2020), il est difficile de définir des critères et leur opérationalisation. Ceci est notamment lié aux collaborations limitées entre les expert.e.s en IAX et les communautés des sciences sociales et de l'interaction humain-machine (IHM) (Miller, 2019; Ehsan *et al.*, 2021). Ainsi, plusieurs cadres d'évaluation existent dans la littérature telles que présentées au Tableau 1.1. Comme les explications sont rarement une finalité (Krishnan, 2020), une évaluation basée sur l'application considérant la tâche à accomplir et les parties prenantes permet de réellement attester de l'utilité des explications. Malheureusement, ce type d'évaluation est peu commun dans la littérature comme il est coûteux. Une approche alternative est l'évaluation centrée sur l'humain où les personnes uti-

Tableau 1.1 – Catégorisation de l'évaluation de techniques d'explicabilité. Tiré de Doshi-Velez et Kim (2017).

Évaluation centrée sur	Humains impliqués ?	Types de tâches	Coût
l'application	Oui	Réelle	Élevé
l'humain	Oui	Simplifiée	Moyen
la fonctionnalité	Non	Substitut	Faible

lisatrices non spécialisées et la tâche considérée est simple. C'est notamment ce que Ribeiro *et al.* (2016) ont fait. Autrement, quelques protocoles d'évaluations tel que DeYoung *et al.* (2020) existent mais ne permettent pas d'assurer leur pertinence en pratique (Bilodeau *et al.*, 2022; Rozario et Čevora, 2023).

Or, pour une large adoption des solutions proposées par la communauté en IAX, il est nécessaire de répondre aux besoins d'explication des personnes utilisatrices afin que l'information partagée leur soit pertinente et utile. De plus, entre le développement d'un modèle et son utilisation pour une tâche finale, les besoins d'explication évoluent (Dhanorkar *et al.*, 2021). Ainsi, des travaux multi-disciplinaires alliant des expertises en IAX et IHM sont indispensables. Ceci est d'autant plus important lorsque les explications s'adressent à des personne utilisatrices avec des connaissances en IA très variables (Ehsan *et al.*, 2021). Malheureusement, les méthodes en IAX ont plutôt été développées par des praticien.nes de l'IA et pour des praticien.nes de l'IA (Miller, 2019).

Récemment, certains ensembles de données utilisés dans des tâches de classification ont été revus et augmentés avec des justifications concernant la classe assignée produites par des humains (Rajani *et al.*, 2019; Aggarwal *et al.*, 2021). Ainsi, il est maintenant possible d'évaluer les méthodes d'explicabilité avec un jugement humain et à faible coût. En effet, il n'est plus nécessaire d'exécuter une évaluation impliquant la production participative d'un large public non-spécialiste.

En TALN, Wiegreffe et Marasovic (2021) ont recensé trois types d'explications :

— Le *surlignage* : les mots du texte original justifiant la classe assignée sont surlignés. DeYoung *et al.* (2020) a établi un standard pour l'évaluation de ce type d'explications

regroupant plusieurs ensembles de données pour des tâches de classification de textes ayant un degré de complexité varié.

— L'*explication ouverte* : une prédiction est justifiée en langage naturel et sans structure particulière. Wiegreffe *et al.* (2021) note que ce type d'explication est plus approprié pour des tâches complexes telles que le raisonnement basé sur le sens commun se reposant sur des connaissances qui ne sont pas explicites dans le texte en entrée.

— Les *explications structurées* regroupent l'ensemble des justifications dont la forme est circonscrite à un quelconque balisage prédéterminé. Ainsi, il s'agit d'explications qui sont ni un surlignage, ni une explication ouverte. Le guide d'annotation peut notamment exiger l'utilisation de règles d'inférence (Lamm *et al.*, 2021) ou de chaînes de faits (Khot *et al.*, 2020). Pour certaines tâches telles que la reconnaissance d'implications textuelles (RIT), ce type d'explication est plus approprié (Camburu *et al.*, 2018).

On peut se référer à la figure 1.5 pour un exemple de chacun. L'avènement des ensembles de données avec des explications rédigées par des humains a permis l'entraînement de modèles autojustifiants tels que ceux présentés en section 1.3.2 et le développement de métriques pour l'évaluation des explications automatiques (DeYoung *et al.*, 2020). Les deux aspects les plus populaires pour évaluer la qualité d'une explication sont :

— la *fidélité* : l'explication est-elle conforme au modèle ?

— la *plausibilité* : l'explication est-elle vraisemblable ?

Selon le type d'explication considérée, la modélisation mathématique de ces aspects diffère (voir par exemple DeYoung *et al.* (2020) et Wiegreffe *et al.* (2021)).

Dans suite de cette section, l'accent est mis sur des techniques tentant d'améliorer la transparence et l'interprétabilité d'une prédiction donnée en classification de textes. Ces techniques se séparent en trois catégories (Lyu *et al.*, 2022) :

— les **modèles autojustifiants** qui obtiennent conjointement la prédiction ainsi que son explication,

— les **explications post-hoc** où un second module justifie les prédictions,

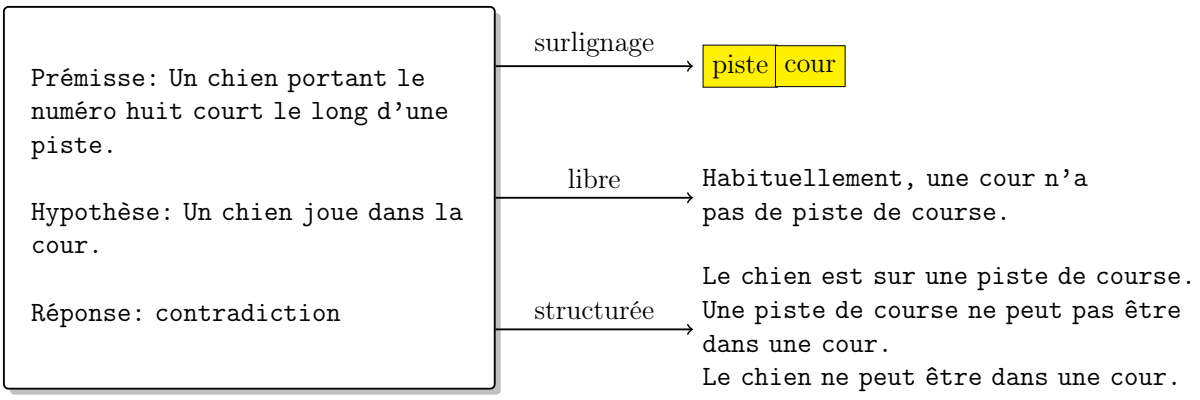


Figure 1.5 – Représentation graphique des différents types d'explications pour une tâche de RIT

— les modèles dits « **fidèles par construction** » [7] qui extraient des explications potentielles puis se basent sur celles-ci pour obtenir la prédiction.

Cette dernière approche n'est pas considérée dans ces travaux. On peut se référer à Wiegreffe *et al.* (2021) et Lyu *et al.* (2022) pour une présentation des limites des modèles fidèles par construction. Cette section est grandement inspirée de la revue de la littérature de Sun *et al.* (2021). Pour une présentation plus générale de l'explicabilité, on peut se référer à Ras *et al.* (2021) et Molnar (2022).

1.3.1 Explications post-hoc

Comme les approches à l'état de l'art en TALN sont très complexes et beaucoup basées sur l'architecture des Transformeurs (voir section 1.2), plusieurs méthodes d'explication post-hoc ont été développées. Pour les tâches de classification, ces dernières s'exécutent dans un pipeline après que le Transformeur ait prédit la classe. Ainsi, les explications post-hoc justifient les décisions mais sans affecter les performances du classifieur. Toutefois, il n'est pas possible de garantir que l'explication donnée soit fidèle (Jacovi et Goldberg, 2021). Comme illustré à la figure 1.6, cette opération peut être faite lorsque le modèle est dit *boîte noire*, soit lorsque seuls le texte original et la décision sont accessibles pour la justification, ou *boîte blanche*, si des composants internes tels que les gradients (voir section 1.1) ou les exemples

7. On peut se référer à Jacovi et Goldberg (2021) pour un contre-exemple.

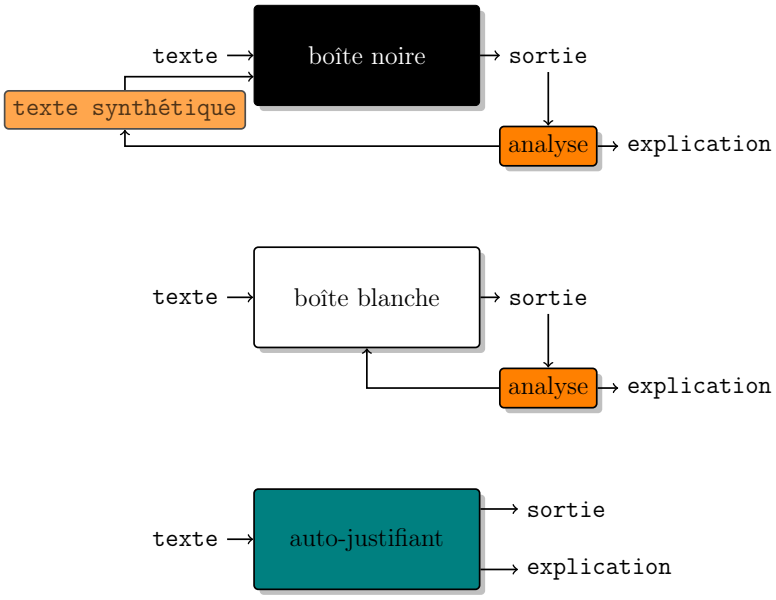


Figure 1.6 – Représentation graphique de différentes approches en explicabilité. Alors que la boîte blanche permet l'analyse des composantes internes du modèle, seules les sorties du modèles peuvent être analysées avec une boîte noire. Ainsi, plusieurs textes synthétiques sont générés et l'analyse considère leur sortie.

d'entraînement sont aussi disponibles. Une brève présentation de ces deux paradigmes suit. On peut se référer à Molnar (2022) pour une présentation plus exhaustive.

1.3.1.1 Boîte noire

Lorsque seuls le texte original et la décision sont accessibles pour extraire une explication, il est courant de générer de nouveaux exemples pour estimer la frontière de décision (Ribeiro *et al.*, 2016; Lundberg et Lee, 2017). Par exemple, en perturbant ou substituant un sous-ensemble du texte original, il est possible d'obtenir une décision différente. Ces perturbations incluent la négation de la phrase, la modification du genre du sujet de la phrase et de la ponctuation (voir Ribeiro *et al.*, 2020, pour plus de détails). Ainsi, les modifications apportées au texte original permettent de mettre en lumière comment elles contribuent à la prédiction : à l'aide d'un modèle entraîné sur les textes générés et leur prédiction, on peut estimer l'importance des mots (Ribeiro *et al.*, 2016; Lundberg et Lee, 2017). Il s'agit d'explications dites *contrastives* (Ross *et al.*, 2021). Néanmoins, comme les perturbations sont plutôt arbitraires, la fiabilité de ces explications est discutable (Slack *et al.*, 2020), particulièrement en TALN

où un contenu textuel plausible doit vérifier certaines règles de la langue (Morris *et al.*, 2020). En effet, même une perturbation minime peut peut générer un texte hors distribution.

1.3.1.2 Boîte blanche

Lorsque des composantes internes du modèle sont accessibles, des explications plus fiables et variées deviennent possibles. En particulier, la disponibilité des gradients informe du comportement local de la frontière de décision, autrement dit, des directions pour lesquelles les variations de la fonction de perte sont les plus brusques et la décision moins « stable ». Cette information est utile pour identifier les exemples dit *contre-factuels* (Wachter *et al.*, 2018), soit les exemples les plus « proches » menant à une décision différente ou les exemples à attaquer pour tromper le modèle (Goodfellow *et al.*, 2015). De plus, à l'aide de la règle de dérivation en chaîne (voir section 1.1), il est possible d'estimer l'importance des mots du texte original pour une prédiction donnée avec des cartes de saillance (Simonyan *et al.*, 2014), ou encore de déterminer quels sont les exemples d'entraînement associés aux plus grandes variations à l'aide des fonctions d'influence (Koh et Liang, 2017). Comme ces explications requièrent de l'information qui n'est pas disponible pour le grand public, elles sont plutôt utilisées par les practicien.nes ainsi que les développeur.ses de modèles pour mieux saisir le comportement du modèle ou le déboguer (Bhatt *et al.*, 2020b; Dhanorkar *et al.*, 2021).

1.3.2 Modèles autojustifiants

Il demeure difficile d'utiliser des modèles interprétables et d'obtenir des résultats à l'état de l'art pour certaines tâches, notamment en TALN (Wiegreffe *et al.*, 2021). Pour améliorer l'interprétabilité des modèles complexes utilisés en TALN, Zhang *et al.* (2016) ont proposé une approche multi-tâches incluant la classification ainsi que l'explication de cette dernière. Pour ce faire, les données annotées incluent la classe assignée ainsi que la justification tel que présenté au début de section 1.3. De plus, des résultats prometteurs en vision par ordinateur suggèrent que les modèles neuronaux apprenant des attributs similaires à ceux des humains sont plus robustes (Zhang *et al.*, 2019). Dans le même ordre d'idées, certaines approches en TALN utilisent, lors de l'entraînement, la justification du choix de la classe pour améliorer les

performances de leurs modèles (Zaidan *et al.*, 2007). Comme la prédiction et son explication sont générées à l'aide de la même représentation interne, la fidélité de l'explication est plus plausible que dans le contexte post-hoc, mais pas garantie. En effet, un modèle peut donner une explication sans toutefois l'utiliser pour obtenir la classe prédite. On présente par la suite les modèles utilisant le surlignage et les explications ouvertes lors de l'entraînement.

1.3.2.1 Surlignage

L'intuition derrière le surlignage ayant des similarités avec l'attention, il fut naturel de superviser l'attention avec ce type d'annotation en transformant les classifieurs en modèles multi-tâches (Bao *et al.*, 2018). En effet, la classe ainsi que l'attention constituent les sorties du modèle. Pour ce faire, il est commun d'ajouter un terme à la fonction de perte pénalisant l'attention (se référer à la section 1.2) divergeant des annotations humaines telle que l'entropie croisée (Mathew *et al.*, 2021). À noter qu'il est nécessaire d'appliquer une transformation au surlignage, qui est une représentation binaire des composantes importantes du texte original, pour la comparaison avec l'attention du modèle, qui est une représentation continue. Ce type d'architecture a mené à des gains de performances en classification de textes (Bao *et al.*, 2018; Mathew *et al.*, 2021) ainsi qu'en extraction d'explications des prédictions (Strout *et al.*, 2019). Suite aux travaux précurseurs de Zaidan *et al.* (2007) et Zhang *et al.* (2016), plusieurs ensembles de données pour d'autres tâches de classification de textes ont vu le jour afin d'encourager la recherche sur les modèles autojustifiants (Wiegreffe et Marasovic, 2021). Toutefois, cette pratique étant récente, les directives d'annotations entre les divers ensembles de données ne sont pas constantes. Ainsi, il est difficile d'interpréter les résultats obtenus sur des standards d'évaluation tels que DeYoung *et al.* (2020). De plus, un débat subsiste à savoir si l'attention peut être considérée comme une explication (se référer à Bibal *et al.*, 2022, pour un survol).

1.3.2.2 Explications ouvertes

Comme ces explications sont en langage naturel, les modèles séquence-à-séquence (voir section 1.2) permettent de générer la classe ainsi que l'explication à partir d'une même représen-

tation interne (Camburu *et al.*, 2018; Rajani *et al.*, 2019; Narang *et al.*, 2020). Ainsi, la sortie du modèle contient la classe prédite suivie d'une explication. L'énorme quantité de contenus textuels considérés lors du pré-entraînement permet à de gros modèles tels que T5 (Raffel *et al.*, 2020) et InstructGPT (Ouyang *et al.*, 2022) de capter une grande quantité de connaissances du monde et de les utiliser pour justifier leur prédiction (Dasgupta *et al.*, 2022). En particulier, Narang *et al.* (2020) obtiennent des résultats à l'état de l'art en plus de valider les conditions nécessaires à la fidélité élaborées par Wiegreffe *et al.* (2021). Premièrement, l'injection de bruit au modèle autojustifiant doit affecter la classe prédite et son explication, indiquant que celles-ci sont liées. Deuxièmement, « les mots du texte original contribuant à la prédiction de la classe doivent aussi contribuer à la génération de l'explication et vice versa » [notre traduction].

CHAPITRE 2
FONCTIONS D'INFLUENCE

La robustesse favorise le bon fonctionnement d'un système dans des conditions qui ne sont pas idéales. En apprentissage automatique, il est important de se préoccuper de cette propriété comme on cherche à produire des modèles qui peuvent généraliser à plusieurs sortes de contexte. Un modèle est dit robuste lorsque des petites modifications sur la formulation du problème impliquent au plus des petites variations entre le modèle original et le modèle résultant (Huber, 1981). Ces modifications peuvent être au niveau des hypothèses du modèle, du poids relatif des données, etc. (Cook et Weisberg, 1982). En apprentissage automatique, on cherche typiquement à répondre à la question suivante : « Si on avait des données différentes, quel effet cela aurait-il sur le modèle résultant ? » [Notre traduction] (Koh et Liang, 2017). Les modèles d'une même architecture étant définis par leurs paramètres, il suffit de mesurer l'écart entre les paramètres du modèle orignal et ceux du modèle résultant pour analyser les variations.

On définit l'*influence d'une observation* comme les variations que les modifications qu'elle subit causent au modèle. Ce type d'analyse est originaire des statistiques robustes. Dans la littérature, la principale modification considérée est la variation du poids relatif d'une observation dans la fonction de perte (Cook et Weisberg, 1982), allant de la surreprésentation à l'exclusion. Il s'agit d'une approche similaire au Jackknife (Quenouille, 1956), une technique de validation croisée un-contre-tous.

Examiner l'influence des données d'entraînement peut être effectué dans un but de diagnostic, notamment pour identifier des faiblesses du modèle (Koh et Liang, 2017; Guo *et al.*, 2021), ou proactif, pour exclure les observations qui sont vraisemblablement mal étiquetées (Koh et Liang, 2017; Teso *et al.*, 2021) ou augmenter efficacement les données (Yang *et al.*, 2020; Lee *et al.*, 2020; Zhang *et al.*, 2022). L'augmentation fait référence à l'ajout de nouvelles observations à l'ensemble d'entraînement afin d'obtenir un modèle plus performant. Parmi les applications classiques de l'analyse de l'influence, on retrouve la détection d'observations

hors de la distribution et l'évaluation de la couverture d'un ensemble de données et sa suffisance pour la prédiction (Huber, 1981; Cook et Weisberg, 1982). De plus, un ajustement du poids relatif des données d'entraînement peut être effectué à l'aide de l'influence (Cook et Weisberg, 1982; Ting et Brochu, 2018; Liu *et al.*, 2021). En explicabilité, les fonctions d'influence (FI) sont principalement utilisées dans un but descriptif : selon le signe de sa valeur d'influence, une observation sera considérée comme bénéfique ou néfaste (Koh et Liang, 2017; Han *et al.*, 2020; Rancourt *et al.*, 2022). Les autres applications des FI incluent l'empoisonnement des données (Koh et Liang, 2017), qui cherche à malicieusement causer des erreurs de classification, le désapprentissage automatique (Wu *et al.*, 2022), qui permet d'« oublier » des observations des données d'entraînement, et l'identification de corrélations fallacieuses appris par le modèle (Han *et al.*, 2020).

Ce chapitre permet de motiver et d'introduire la théorie de l'influence. C'est sur ces notions que s'appuient les travaux présentés aux chapitres 3 et 4. Ce chapitre est structuré comme suit. Les fondements de l'influence en statistiques robustes présentant le contexte classique de l'analyse de l'influence, la régression, sont discutés dans la section 2.1. Cette présentation est inspirée des travaux de Huber (1981) et de Cook et Weisberg (1982). Les travaux de Koh et Liang (2017), qui ont formalisé l'analyse de l'influence en apprentissage automatique où les régressions non linéaire et logistique sont courantes, sont grandement inspirés de la section 5.2 de Cook et Weisberg (1982). La section 2.2 présente cette formulation et ces limites. Ensuite, une courte présentation des méthodes d'évaluation des FI en IA et des améliorations visant à accélérer leurs calculs conclut ce chapitre.

2.1 L'influence pour la régression linéaire

Les problèmes de régression tentent d'approcher une fonction, une droite ou une courbe, à l'aide d'un ensemble d'observations. Plus particulièrement, on cherche la fonction η liant une variable dépendante y dite *réponse* à un ensemble de prédicteurs X appelés covariables : soit $y \approx \eta(X)$. Pour la régression linéaire (Galton, 1886), on cherche à estimer le vecteur des coefficients de régression θ de sorte que $y \approx X\theta$.

Encore aujourd'hui, les modèles de régression linéaire demeurent centraux en statistiques et dans ces domaines d'application. Un modèle linéaire peut être considéré pour ses capacités prédictives ou pour expliquer les variations de la variable réponse attribuées à chacune des covariables (Breiman, 2001). Cette dernière utilisation est particulièrement importante lorsqu'on cherche à quantifier l'effet d'une covariable sur la variable réponse.

La régression linéaire étant un modèle populaire et facile à appréhender, elle est un excellent exemple pour motiver et présenter les FI. Cette section est structurée comme suit. Premièrement, la sous-section 2.1.1 présente la formulation du problème de la régression linéaire. Deuxièmement, la sous-section 2.1.2 discute des limites de l'analyse des résidus et du Jackknife pour l'estimation de l'influence des données d'entraînement sur le modèle. Enfin, la sous-section 2.1.3 présente la théorie des FI pour la régression linéaire.

2.1.1 Formulation du problème

Soit (X_i, y_i), la i^{e} observation d'un échantillon de taille n avec $y_i \in \mathbb{R}$ et $X_i \in \mathbb{R}^{m+1}$ pour $m < n$. On cherche à exprimer $y = (y_1, \ldots, y_n)^T$ en fonction de

$$X = \begin{pmatrix} 1 & x_{11} & \ldots & x_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \ldots & x_{nm} \end{pmatrix}. \tag{2.1}$$

On suppose que $X_1, \ldots, X_n$ sont statistiquement indépendantes, [1] c'est-à-dire que

$$\Pr(X_1, \ldots, X_n) = \prod_{i=1}^{n} \Pr(X_i).$$

Ainsi, la matrice X est inversible comme ses lignes sont linéairement indépendantes. [2]

1. Comme cette hypothèse est rarement réalisable en pratique, il est possible de relaxer cette contrainte (Hoaglin et Welsch, 1978). Pour les besoins de cette présentation, seul le problème classique est considéré.

2. Dans le cas contraire, simplement conserver les lignes linéairement indépendantes comme les autres n'apportent pas de nouvelle information.

S'il existe une relation linéaire entre X et y, c'est-à-dire que les coefficients à estimer sont de degré 1, alors il est approprié d'utiliser un modèle de régression linéaire de la forme :

$$y = X\theta + \epsilon$$

où $\epsilon \sim \mathcal{N}(0, \sigma^2 I_n)$ est l'erreur observée, qu'on appelle le résidu, et θ est appelé le vecteur des coefficients de régression. Comme il n'est pas possible d'observer θ et ϵ, on les estime à l'aide de $\hat{\theta}$ et $\hat{\epsilon}$ respectivement. Plus particulièrement, on pose $\hat{y} = X\hat{\theta}$ puis on déduit $\hat{\epsilon} = y - \hat{y}$. Cette dernière quantité est essentielle pour l'estimation du vecteur des coefficients de régression avec la méthode des moindres carrés ordinaires (MCO). Celle-ci, pouvant être attribuée à Gauss (Stigler, 1981), cherche l'estimateur $\hat{\theta}$ minimisant la fonction de perte $\mathcal{L}$ définie par :

$$\mathcal{L}(X, y, \hat{\theta}) = \|\hat{\epsilon}\|^2 = \|y - X\hat{\theta}\|^2. \tag{2.2}$$

Théorème 1. *Avec la méthode des MCO, on obtient*

$$\hat{\theta} = \left(X^T X\right)^{-1} X^T y. \tag{2.3}$$

On peut se référer à l'appendice A.1 pour la démonstration.

2.1.2 Motivations des fonctions d'influence

L'architecture de la régression linéaire étant définie, on peut maintenant analyser sa robustesse. Ceci est fait en modifiant les données considérées lors de l'entraînement. Par exemple, en présence de données aberrantes, c'est-à-dire des observations très différentes des autres, les coefficients de la régression linéaire peuvent subir d'importantes variations. Lorsque le modèle est robuste, les modifications faites aux données d'entraînement affectent peu les paramètres. Ainsi, le modèle original est fiable. Les approches suivantes peuvent être considérées :

— L'analyse des résidus avec la matrice chapeau, qui relie la variable réponse aux prédictions, permet d'identifier les *points à fort effet de levier*, des observations qui seraient mal prédites si elles étaient exclues des données d'entraînement (Huber, 1981).

— La méthode du Jackknife (Quenouille, 1956) qui consiste à exclure séquentiellement chacune des observations une-à-une puis d'entraîner un modèle. Il s'agit d'un cas spécifique de la validation croisée (sous-section 1.1).

Cette sous-section présente le fonctionnement des méthodes ci-haut.

2.1.2.1 Analyse de la matrice chapeau

Il est possible d'exprimer y en fonction de $\hat{y}$ à l'aide de la matrice des projections P, dite matrice chapeau. Celle-ci permet de relier la variable réponse aux prédictions. En développant l'équation (2.3), on a

$$\hat{y} = X\hat{\theta} = \underbrace{X\left(X^TX\right)^{-1}X^T}_{P}y.$$

De même, on peut observer que $\hat{\epsilon} = (I_n - P)\epsilon$. Bien que P ne dépende que de la matrice X, on peut interpréter p_{ij}, la valeur de la matrice P aux coordonnées (i, j), comme l'influence de l'observation y_i sur la prédiction $\hat{y}_j$ (Hoaglin et Welsch, 1978) puisque

$$\hat{y_j} = p_{ij}y_i + \sum_{k\neq i} p_{kj}y_k.$$

Définition 1. On dit que l'observation y_i est *utile* (resp. *nuisible*) à la prédiction de $\hat{y}_j$ lorsque p_{ij} est positif (resp. négatif).

Les termes diagonaux de P sont particulièrement intéressants comme ils indiquent qu'il s'agit de points à fort effet de levier. En effet, lorsque p_{ii} est grand, la connaissance de y_i est vitale à la prédiction de $\hat{y}_i$ comme

$$\hat{y_i} = p_{ii}y_i + \sum_{j\neq i} p_{ij}y_j \approx p_{ii}y_i.$$

Dans l'optique où il est souhaitable d'avoir un modèle robuste, les points à fort effet de levier sont à éviter. En effet, observer des valeurs s'approchant de 0 sur la diagonale de P indique que la prédiction $\hat{y}_i$ est influencée par plusieurs autres observations. Ainsi, même si (X_i, y_i)

n'était pas connu à priori, sa prédiction demeurerait plutôt stable. Lorsque p_{ii} s'approche de 1, la i^{e} observation est un point à fort effet de levier.

Exemple 1. Soit les observations suivantes :

i	1	2	3	4	5	6	7
X_i	2,3	3,1	1,5	3,4	3,9	0,7	6
y_i	0,8	1,05	0,65	1,25	1	0,45	-10

On remarque que la dernière observation $(6, -10)$ est aberrante, mais ignorons ce fait pour le moment. En reprenant la notation de l'équation (2.1), on a

$$X^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2,3 & 3,1 & 1,5 & 3,4 & 3,9 & 0,7 & 6 \end{pmatrix}$$

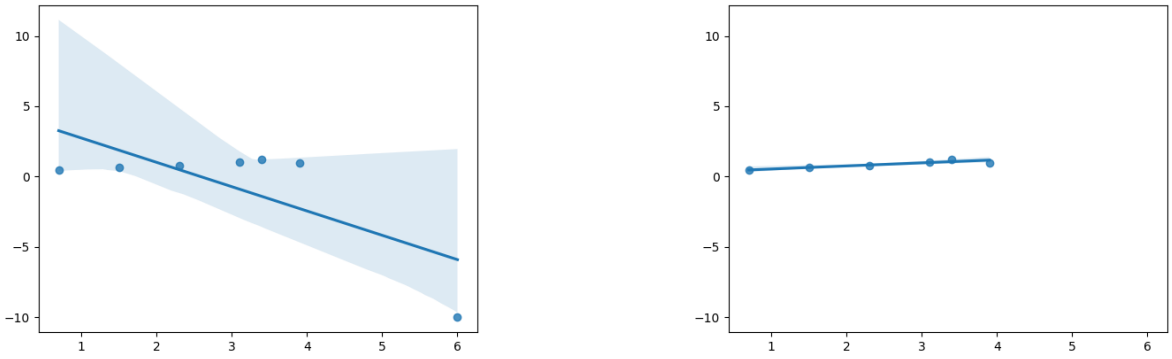
et on trouve, avec les MCO,

$$\hat{\theta}^T = (4,476 \quad -1,729)$$

$$P = \begin{pmatrix} 0,169 & & & & & & \\ 0,139 & 0,144 & & & & & \\ 0,199 & 0,133 & 0,265 & & & & \\ 0,127 & 0,145 & 0,109 & 0,152 & & & \\ 0,108 & 0,149 & 0,067 & 0,164 & 0,189 & & \\ 0,230 & 0,128 & 0,331 & 0,090 & 0,027 & 0,433 & \\ 0,028 & 0,162 & -0,106 & 0,212 & 0,296 & -0,240 & 0,647 \end{pmatrix}.$$

Pour faciliter la lisibilité, les chiffres ont été arrondis au millième près et seul le triangle inférieur de P est présenté.

On peut remarquer que p_{66} et p_{77} sont substantiellement plus grands que les autres diagonaux. De plus, on observe à la figure 2.1a que le point $(6, -10)$ « tire » la pente négativement. Ce dernier est donc un point à fort effet de levier.



(a) Avec le point aberrant

(b) En excluant le point aberrant

Figure 2.1 – L'exclusion du point aberrant en (6, -10) a un profond impact sur le modèle obtenu. Les intervalles de confiance sont à 95%.

De plus, le point aberrant est nuisible à la prédiction de $\hat{y}_3$ et $\hat{y}_6$ puisque p_{73} et p_{76} sont négatifs. Il est intéressant de noter que $p_{ij} > 0$ pour $i, j \in \{1, \ldots, 6\}$.

Excluons maintenant le point $(6, -10)$ du problème de régression. On trouve

$$\hat{\theta}^T = (0,323 \;\; 0,219)$$

$$P = \begin{pmatrix} 0,171 & & & & & \\ 0,151 & 0,218 & & & & \\ 0,191 & 0,085 & 0,297 & & & \\ 0,144 & 0,243 & 0,045 & 0,280 & & \\ 0,132 & 0,285 & -0,021 & 0,342 & 0,438 & \\ 0,211 & 0,018 & 0,403 & -0,054 & -0,174 & 0,596 \end{pmatrix}$$

et la droite de régression présentée à la figure 2.1b. L'exclusion du point aberrant a eu un effet substantiel sur le modèle de régression. En effet, la pente est maintenant positive. De plus, p_{55} et p_{66} ont des valeurs substantiellement plus grandes que les autres termes diagonaux. On peut remarquer que ceux-ci ont des résidus $\hat{\epsilon}$ plus plus grands que les autres observations, ce qui explique cette variation.

Comme illustré à l'exemple 1, l'analyse des termes diagonaux de la matrice chapeau est utile, mais aussi un art. En effet, bien que la présence du point aberrant change beaucoup la droite de régression, les valeurs de la matrice chapeau avec et sans cette observation sont plutôt similaires. Ainsi, il n'existe pas de critère universel pour déterminer ce que sont des petites et des grandes valeurs pour les termes de la matrice chapeau.

2.1.2.2 Jackknife

Le Jackknife consiste à comparer les variations observées en excluant chacune des observations, exactement une fois. Ainsi, il s'agit d'une validation croisée avec $n-1$ blocs de taille 1. En vertu de la grande quantité de blocs, il est nécessaire d'entraîner $N+1$ modèles : N pour la validation croisée et 1 pour l'ensemble des données. Pour un estimateur θ, le Jackknife permet d'estimer empiriquement le biais, $\mathcal{B}(\hat{\theta}) = \mathbb{E}(\hat{\theta}) - \theta$, comme suit :

$$\hat{\mathcal{B}}(\hat{\theta}) = (n-1)\left(\frac{1}{n}\sum_{i=1}^{n}\hat{\theta}_{(i)} - \hat{\theta}\right)$$

où $\hat{\theta}_{(i)}$ est l'estimateur obtenu à l'aide des observations $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$.

Exemple 2 (Suite de l'exemple 1)**.** Soit les observations suivantes :

i	1	2	3	4	5	6	7
X_i	2,3	3,1	1,5	3,4	3,9	0,7	6
y_i	0,8	1,05	0,65	1,25	1	0,45	-10

On sait que $\hat{\theta}^T = (4,476 \quad -1,729)$ et $\hat{\theta}^T_{(7)} = (0,323 \;\; 0,219)$. Calculons les $\hat{\theta}_{(i)}$ manquants :

$$\begin{aligned}
\hat{\theta}^T_{(1)} &= (4,383 \quad -1,715) \\
\hat{\theta}^T_{(2)} &= (4,196 \quad -1,743) \\
\hat{\theta}^T_{(3)} &= (5,129 \quad -1,867) \\
\hat{\theta}^T_{(4)} &= (4,244 \quad -1,801) \\
\hat{\theta}^T_{(5)} &= (4,511 \quad -1,933) \\
\hat{\theta}^T_{(6)} &= (7,067 \quad -2,359).
\end{aligned}$$

De plus, $\hat{\mathcal{B}}(\hat{\theta}^T) = (-1,266 \;\; 0,774)$.

$\hat{\theta}^T_{(7)}$ étant substantiellement différent des autres $\hat{\theta}^T_{(i)}$, il est évident que la 7$^{\text{e}}$ observation a une importante influence sur la régression obtenue. Comme il s'agit d'une observation abberante, ce résultat n'est pas étonnant.

Bien qu'utiles, ces méthodes ont d'importantes limites. L'exemple 1 a mis en lumière que l'analyse de la matrice chapeau n'est pas suffisante pour évaluer la contribution d'une observation. Tandis que l'exemple 2 a nécessité l'entraînement de 8 modèles linéaires, ce qui ne passe pas à l'échelle pour les grands ensembles de données utilisés pour l'entraînement des modèles à l'état de l'art en apprentissage automatique.

2.1.3 Courbe de l'influence

Comme exclure une observation et entraîner de nouveau le modèle nécessite trop de calculs, on considère plutôt les FI. Celles-ci modifient le poids relatif d'une observation dans la fonction de perte. Tel que formulé à la section 2.1.1, toutes les observations contribuent également à la fonction de perte définie à l'équation (2.2) avec un poids de $\frac{1}{n}$.

Reprenons les exemples 1 et 2 de la section précédente. Pour déterminer l'influence de l'observation aberrante, les approches suivantes sont possibles :

Q1 Pour $\varepsilon > 0$, quelles variations seraient observées si l'observation 7 avait plutôt une pondération de $1 + \varepsilon$?

Q2 Quelles variations seraient observées si l'observation 7 était exclue ?

À l'exemple 1, nous avons répondu à **Q2** en entraînant de nouveau le modèle linéaire en excluant l'observation 7. De même, on peut répondre à **Q1** en entraînant de nouveau un modèle linéaire avec la fonction de perte $\|\hat{\epsilon}\|^2 + \varepsilon\, \hat{\epsilon}_7{}^2$ où $\hat{\epsilon}_7$ est l'erreur de prédiction pour l'observation 7. Néanmoins, il est possible, à l'aide des FI, de répondre à cette question sans résoudre de problème d'optimisation.

Pour définir les FI telles qu'on les utilise en apprentissage automatique, il est nécessaire de définir au préalable la *courbe de l'influence*. Cette dernière considère des variables fonctionnelles, des variables aléatoires prenant des valeurs dans un espace de dimension infinie $\mathcal{F}$, et des statistiques fonctionnelles. Ces dernières sont définies à l'aide d'une statistique, une suite d'opérations effectuées sur un échantillon, et d'une fonctionnelle, une fonction agissant sur des fonctions.

Définition 2 (Fernholz (1983))**.** Soit $x_1, \ldots, x_n$, un échantillon issu de la distribution F et T_n une statistique. S'il existe une fonctionnelle T ne dépendant pas de n telle que $T_n(x_1, \ldots, x_n) = T(\hat{F})$ où

$$\hat{F}(s) = \frac{1}{n} \sum_{1 \leq i \leq n} \mathbb{1}\left(x_i \leq s\right)$$

est la fonction de répartition empirique, alors T est une statistique fonctionnelle.

Exemple 3. Soit F une fonction de répartition d'une variable continue définie sur $\mathbb{R}$ de moyenne μ. Alors, $\mu = \int x \, \mathrm{d}F(x) = T(F)$ où T est une statistique fonctionnelle.

Pour plus d'exemples de statistiques fonctionnelles, on peut se référer à Wang *et al.* (2016).

Dans ce chapitre, on considère uniquement des espaces munis de la norme euclidienne [3] et des statistiques fonctionnelles sur $\mathbb{R}^k$ avec $k \in \mathbb{N}$. On pose $T(F) = \theta$ où θ est le paramètre

3. Ainsi, nous travaillons dans un espace de Banach, car il est de dimension finie et normé.

à estimer. De plus, on suppose que le domaine de T comprend l'ensemble des fonctions de répartition empiriques $\{\hat{F} \in \mathbb{R}^{m+1} : m \geq 1\}$ et la fonction de répartition de la population $F \in \mathbb{R}^{m+1}$.

Définition 3 (Huber (1981))**.** Soit F une fonction de répartition, T une statistique fonctionnelle et δ_z la fonction de répartition attribuant toute sa masse à $z \in \mathbb{R}^{m+1}$. On appelle $\mathcal{IC}_{F,T}$ la courbe de l'influence lorsque

$$\mathcal{IC}_{F,T}(z) = \lim_{\varepsilon \to 0} \frac{T\left((1-\varepsilon)F + \varepsilon\delta_z\right) - T(F)}{\varepsilon}. \tag{2.4}$$

Ainsi, la courbe de l'influence « représente le taux de changement d'une statistique fonctionnelle advenant une légère contamination par [l'observation z] » [notre traduction] (Breheny, 2012). La définition 3 utilise ce qu'on appelle la dérivée de Gâteaux (Gâteaux, 1922) qui généralise la dérivée directionnelle. Ainsi, on peut généralement [4] réécrire l'équation (2.4) comme suit

$$\mathcal{IC}_{F,T}(z) = \left.\frac{\mathrm{d}T(F + \varepsilon D)}{\mathrm{d}\varepsilon}\right|_{\varepsilon=0} \tag{2.5}$$

où $D = \delta_z - F$. Le terme de droite de l'équation (2.5) est appelé la dérivée de von Mises. Pour une présentation plus détaillée de la dérivation de statistiques fonctionnelles, on peut se référer à Fernholz (1983) et Ren et Sen (1995).

Pour se familiariser avec la définition de la courbe de l'influence, considérons premièrement une statistique simple : la moyenne.

Exemple 4 (Exemple 3.2.1 de Cook et Weisberg (1982), suite de l'exemple 3)**.** On considère la moyenne $\mu = \int x \, \mathrm{d}F(x) = T(F)$. Posons δ_z la fonction de répartition attribuant toute sa

4. Voir l'exemple 2.2.2 de Fernholz (1983) pour un contre-exemple.

masse à l'observation z. La courbe de l'influence est

$$\begin{aligned}\mathcal{IC}_{F,T}(z) &= \lim_{\varepsilon \to 0} \frac{\int x \, \mathrm{d}\left((1-\varepsilon)F(x) + \varepsilon\delta_z(x)\right) - \int x \, \mathrm{d}F(x)}{\varepsilon} \\ &= \lim_{\varepsilon \to 0} \frac{(1-\varepsilon)\mu + \varepsilon z - \mu}{\varepsilon} \\ &= z - \mu.\end{aligned}$$

Ce calcul repose sur le fait que la fonctionnelle considérée est linéaire, c'est-à-dire, $T(aF + bG) = aT(F) + bT(G)$ pour toutes constantes $a, b \in \mathbb{R}$ et toutes fonctions F, G.

Ainsi, plus z s'éloigne de la moyenne, plus son influence sur la moyenne sera forte et ce sans borne ($\mathcal{IC}_{F,T}$ n'est pas bornée). Ceci est cohérent avec le fait que la moyenne n'est pas robuste aux valeurs aberrantes.

Les FI sont appelées courbes de l'influence empirique dans la communauté des statistiques robustes. Il s'agit d'un estimateur plug-in de la courbe de l'influence (Hampel, 1974), c'est-à-dire qu'on l'estime avec la distribution empirique $\hat{F}$.

Définition 4. Soit $x_1, \ldots, x_n$, un échantillon indépendant et identiquement distribué (iid) issu de la distribution F et $\hat{F}$ la fonction de répartition empirique. La courbe de l'influence empirique est donc

$$\widehat{\mathcal{IC}}_{\hat{F},T}(z) = \lim_{\varepsilon \to 0} \frac{T\left((1-\varepsilon)\hat{F} + \varepsilon\delta_z\right) - T(\hat{F})}{\varepsilon} \tag{2.6}$$

où $\hat{F}$ est la fonction de répartition empirique.

En pratique, on utilise souvent $\epsilon = \frac{1}{n+1}$ et $z = x_i$ dans l'équation (2.6). Ceci revient à augmenter le poids de l'observation x_i. Pour retirer la i^{e} observation, on considère $\epsilon = -\frac{1}{n}$ et $z = x_i$.

De plus, les variations entraînées par l'augmentation du poids relatif de la i^{e} observation est $\widehat{\mathcal{IC}}_{\hat{F},T}(x_i)$, ce qui répond à **Q1**. Pour l'exclusion de la i^{e} observation (**Q2**), on calcule

$\widehat{\mathcal{IC}}_{\hat{F}_{(i)},T}(x_i)$ où $\hat{F}_{(i)}$ est la fonction de répartition empirique excluant la i^{e} observation.

Exemple 5 (Suite de l'exemple 4)**.** Soit $x_1, \ldots, x_n$ un échantillon iid et $\hat{F}$ la fonction de répartition associée. On a

$$T(\hat{F}) = \int x \, \mathrm{d}\hat{F}(x) = \frac{1}{n} \sum_{1 \leq i \leq n} x_i = \hat{\mu}.$$

Avec $\epsilon = \frac{1}{n+1}$, on trouve

$$\begin{aligned} T\left(\frac{n}{n+1}\hat{F} + \frac{1}{n+1}\delta_z\right) - T(\hat{F}) &= \frac{n}{n+1}T(\hat{F}) + \frac{1}{n+1}T(\delta_z) - T(\hat{F}) \\ &= \frac{1}{n+1}(z - \hat{\mu}). \end{aligned}$$

D'où,

$$\widehat{\mathcal{IC}}_{\hat{F},T}(z) = \lim_{n \to \infty} \frac{\frac{1}{n+1}(z - \hat{\mu})}{\frac{1}{n+1}} = z - \hat{\mu}.$$

Calculons maintenant la courbe de l'influence en excluant la i^{e} observation, c'est-à-dire $\widehat{\mathcal{IC}}_{\hat{F}_{(i)},T}(z)$. On a

$$T\left(\frac{n-1}{n}\hat{F}_{(i)} + \frac{1}{n}\delta_z\right) - T(\hat{F}_{(i)}) = \frac{1}{n}\left(z - \hat{\mu}_{(i)}\right)$$

où $\hat{\mu}_{(i)}$ est la moyenne excluant l'observation i. On observe que $\hat{\mu} = \frac{n-1}{n}\hat{\mu}_{(i)} + \frac{1}{n}x_i$, d'où

$$\widehat{\mathcal{IC}}_{\hat{F}_{(i)},T}(x_i) = \lim_{n \to \infty} \frac{n-1}{n}\left(x_i - \frac{n}{n-1}\hat{\mu} + \frac{1}{n-1}x_i\right) = x_i - \hat{\mu}.$$

Maintenant qu'on comprend mieux comment manipuler la courbe de l'influence empirique, il est temps d'analyser si celle-ci est un bon estimateur de la courbe de l'influence. Pour évaluer la qualité d'un estimateur, plusieurs critères peuvent être utilisés, dont la convergence. Conceptuellement, « [un] estimateur convergent est un estimateur avec la propriété que la probabilité de la différence entre la valeur estimée et la vraie valeur du paramètre de la population [...] approche zéro lorsque la taille de l'échantillon tend vers l'infini. » [Notre traduction] (Statistics, 2023). Notamment, la moyenne échantillonale $\hat{\mu}$ est un estimateur

convergent puisqu'elle converge asymptotiquement vers μ. Lorsqu'on travaille avec des statistiques fonctionnelles, on peut utiliser la définition suivante.

Définition 5 (p.8 de Huber (1981))**.** Soit $\epsilon > 0$. Soit F une fonction de répartition et T une statistique fonctionnelle. Lorsque $\lim_{n\to\infty} \Pr(|T(\hat{F}) - T(F)| < \epsilon) = 0$, T est dite convergente en F. On note $T(\hat{F}) \xrightarrow{\text{P}} T(F)$.

On peut observer que dans le cas de la moyenne empirique, $\widehat{\mathcal{IC}}_{\hat{F},T}(z) \xrightarrow{\text{P}} \mathcal{IC}_{F,T}(z)$. Ce comportement est dû au fait que $\hat{\mu}$ est un estimateur convergent et que la statistique fonctionnelle T est linéaire. [5]

Proposition 1. *Soit F une fonction de répartition et $\hat{F}$ sa fonction de répartition empirique. Soit $\hat{\theta} = T(\hat{F})$ un estimateur convergent de $\theta = T(F)$. Si T est une statistique fonctionnelle linéaire alors, $\widehat{\mathcal{IC}}_{\hat{F},T}(z) \xrightarrow{\text{P}} \mathcal{IC}_{F,T}(z)$.*

Esquisse de la preuve. Simplifier $\widehat{\mathcal{IC}}_{\hat{F},T}(z)$ et $\mathcal{IC}_{F,T}(z)$ puis appliquer la définition 5. □

Toutefois, toutes les fonctionnelles ne sont pas linéaires. Ainsi, on peut se tourner vers la dérivée pour obtenir des conditions suffiantes pour que courbe de l'influence empirique soit un estimateur convergent. Comme la dérivée de Gâteaux n'est pas toujours unique (Huber, 1981), elle ne suffit pas à garantir la convergence de $\widehat{\mathcal{IC}}_{\hat{F},T}$. On utilise plutôt la dérivée au sens d'Hadamard (Fernholz, 1983).

Définition 6 (Breheny (2012))**.** Soit D une fonctionnelle. La statistique fonctionnelle T est dérivable au sens de Hadamard si, pour toutes séquences ε_n telles que $\lim_{n\to\infty} \varepsilon_n = 0$ et D_n est tel que $\lim_{n\to\infty} \sup_x |D_n(X) - D(x)| = 0$, alors

$$\frac{\mathrm{d}T}{\mathrm{d}F} = \lim_{n\to\infty} \frac{T(F + \varepsilon_n D_n) - T(F)}{\varepsilon_n} = \lim_{\varepsilon\to 0} \frac{T(F + \varepsilon D) - T(F)}{\varepsilon}.$$

5. Il s'agit de la dérivée au sens de Fréchet (voir définition 2.3.1 de Fernholz, 1983).

Puisque $\hat{F} \xrightarrow{\mathrm{P}} F$, seul un changement de variable est nécessaire pour utiliser la définition ci-dessus dans l'équation (2.4). Ainsi, on pose $D_n = \delta_z - \hat{F}$ et $D = \delta_z - F$.

De plus, lorsque qu'une fonctionnelle est dérivable au sens d'Hadamard, elle l'est aussi au sens de Gâteaux (Fernholz, 1983) et les deux dérivées coïncident (Shapiro, 1991). Ainsi, on obtient le résultat suivant.

Théorème 2. *Lorsque T est dérivable au sens d'Hadamard, $\widehat{\mathcal{IC}}_{\hat{F},T}$ est un estimateur convergent de $\mathcal{IC}_{F,T}$.*

On peut se référer au théorème 2.1 de Shapiro (1991) pour la démonstration.

Ainsi, sous certaines conditions, la courbe de l'influence empirique est un « bon » estimateur. Ce n'est toutefois pas quelque chose qui peut être pris pour acquis.

2.1.3.1 Pour la régression linéaire

Lorsqu'on travaille avec la régression linéaire, qui modélise les données à l'aide d'un plan, il est plus complexe de déterminer la statistique fonctionnelle T à utiliser. On présente ci-dessous la formulation proposée par Hinkley (1977). Cette présentation reprend les sections 3.3 et 3.4 de Cook et Weisberg (1982).

Posons F la fonction de répartition du vecteur (x, y) où $x \in \mathbb{R}^{m+1}$ est le vecteur aléatoire des covariables et $y \in \mathbb{R}$ la variable réponse. On cherche à exprimer le vecteur des coefficients de régression $\theta \in \mathbb{R}^{m+1}$ obtenu par MCO à l'aide d'une variable fonctionnelle. Avec la notation de matrices par blocs, on trouve

$$\mathbb{E}_F \left[\begin{pmatrix} x \\ y \end{pmatrix} (x^T, y) \right] = \begin{pmatrix} \boldsymbol{\Sigma}(F) & \gamma(F) \\ \gamma^T(F) & \tau(F) \end{pmatrix}. \tag{2.7}$$

Ici, $\boldsymbol{\Sigma}(F) = \int xx^T \, \mathrm{d}F(x, y)$ est la statistique de la matrice des variances-covariances de X, $\gamma(F) = \int xy \, \mathrm{d}F(x, y)$ correspond aux covariances entre Y et chacun des indices de X puis

$\tau(F) = \int y^2 \, \mathrm{d}F(x,y)$ correspond à la variance de Y. On suppose $\mathbf{\Sigma}$ non singulière, c'est-à-dire que son déterminant [6] est non nul. Ainsi, $\mathbf{\Sigma}^{-1}$ existe. Enfin, la statistique fonctionnelle T est définie comme suit :

$$T(F) = \mathbf{\Sigma}^{-1}(F)\gamma(F) = \theta \tag{2.8}$$

correspond à l'estimateur des MCO de θ définit à l'équation (2.3).

Pour obtenir la courbe de l'influence telle que spécifiée à la définition 3, il est nécessaire d'évaluer $T((1-\epsilon)F + \epsilon\delta_{(x,y)})$. Ici, $\delta_{(x,y)}$ est la fonction de répartition qui attribue toute sa masse au point (x,y). Comme T est définie en fonction de $\mathbf{\Sigma}$ et γ, calculons $\mathbf{\Sigma}((1-\varepsilon)F + \varepsilon\delta_{(x,y)})$ et $\gamma((1-\varepsilon)F + \varepsilon\delta_{(x,y)})$.

Lemme 1 (Équation (3.3.3) de Cook et Weisberg (1982))**.** *Soit la statistique fonctionnelle T telle que définie à l'équation* (2.8) *et $\delta_{(x,y)}$ la fonction de répartition qui attribue toute sa masse au point (x,y). Alors,*

$$\mathbf{\Sigma}((1-\varepsilon)F + \varepsilon\delta_{(x,y)}) = (1-\varepsilon)\left(\mathbf{\Sigma}(F) + \frac{\varepsilon}{1-\varepsilon}xx^T\right), \tag{2.9}$$

$$\gamma((1-\varepsilon)F + \varepsilon\delta_{(x,y)}) = (1-\varepsilon)\gamma(F) + \varepsilon yx. \tag{2.10}$$

Esquisse de la preuve. Traduire les espérances en intégrales et distribuer l'opérateur de la dérivée. On peut se référer à la section 3.3 de Cook et Weisberg (1982) pour les détails. □

Ainsi, $T((1-\epsilon)F + \epsilon\delta_{(x,y)})$ est le produit de l'inverse de $\mathbf{\Sigma}((1-\varepsilon)F + \varepsilon\delta_{(x,y)})$ avec $\gamma((1-\varepsilon)F + \varepsilon\delta_{(x,y)})$. Il n'est toutefois pas évident d'inverser la quantité décrite à l'équation (2.9). Le lemme suivant nous fournit une formule pour obtenir le résultat.

Lemme 2 (Annexe A.2 de Cook et Weisberg (1982))**.** *Soit A une matrice symétrique $p \times p$ inversible et a, b deux matrices $q \times p$ de rang q. Si l'inverse de $A + a^T b$ existe, alors*

$$\left(A + a^T b\right)^{-1} = A^{-1} - A^{-1}a^T\left(I_p + bA^{-1}a^T\right)^{-1} bA^{-1} \tag{2.11}$$

6. Noter qu'il s'agit ici du déterminant fonctionnel. Cette notion dépasse largement la portée de ce mémoire, mais l'intuition suffit pour suivre la présentation. On peut se référer à Branson (1993) pour connaître les détails.

où I_p est la matrice identité de dimensions $p \times p$.

Esquisse de la preuve. Calculer $\left(A + a^T b\right)\left(A + a^T b\right)^{-1}$ et $\left(A + a^T b\right)^{-1}\left(A + a^T b\right)$ puis observer que les deux quantités sont égales à I_p. □

Ainsi, il est maintenant possible de calculer la courbe de l'influence des coefficients de régression θ.

Théorème 3. *Soit $\delta_{(x,y)}$ la fonction de répartition qui attribue toute sa masse au point (x, y). La courbe de l'influence du vecteur des coefficients de régression θ obtenu avec les MCO est*

$$\mathcal{IC}_{F,T}(x, y) = \mathbf{\Sigma}^{-1}(F)x\left(y - x^T\theta\right) \tag{2.12}$$

où T est tel que défini à l'équation (2.8).

Esquisse de la preuve. Pour trouver $\mathbf{\Sigma}^{-1}((1-\varepsilon)F + \varepsilon\delta_{(x,y)})$, appliquer le lemme 2 à l'équation (2.9). Multiplier le résultat obtenu à l'équation (2.10) puis simplifier. Enfin, passer à la limite pour obtenir l'équation (2.12). Pour la preuve complète, se référer à l'annexe A.2. □

On observe que $\mathcal{IC}_{F,T}$ est un vecteur de même dimension que $\hat{\theta}$. Puisque $\hat{\theta}$ est un estimateur convergent (voir pp. 134–135 de Fahrmeir *et al.*, 2021, pour l'intuition), on obtient le résultat suivant.

Théorème 4 (Équations (3.4.3) et (3.4.4) de Cook et Weisberg (1982))**.** *Pour $\hat{\theta} = T(\hat{F})$ obtenu à l'aide des MCO,*

$$\begin{aligned} \widehat{\mathcal{IC}}_{\hat{F},T}(x, y) &= n\left(\mathbf{X}^T\mathbf{X}\right)^{-1} x(y - x^T\hat{\theta}) \\ \widehat{\mathcal{IC}}_{\hat{F}_{(i)},T}(x_i, y_i) &= (n-1)\left(\mathbf{X}_{(i)}^T\mathbf{X}_{(i)}\right)^{-1} x_i(y_i - x_i^T\hat{\theta}_{(i)}) \end{aligned} \tag{2.13}$$

où n est la taille d'échantillon, $\frac{1}{n}\mathbf{X}^T\mathbf{X} = \int xx^T\, d\hat{F}$ et $\frac{1}{n-1}\mathbf{X}_{(i)}^T\mathbf{X}_{(i)} = \int xx^T\, d\hat{F}_{(i)}$.

Exemple 6 (Suite des exemples 1 et 2)**.** Pour chacune des observations, on calcule 2.13 pour $\widehat{\mathcal{IC}}_{\hat{F},T}(x_i, y_i)$ et on trouve :

i	1	2	3	4	5	6	7
$\widehat{\mathcal{IC}}_{\hat{F},T}(x_i, y_i)$	0,539	1,677	-3,358	1,377	-0,200	-10,283	10,245
	-0,080	0,086	0,712	0,427	1,161	2,501	-4,805

On observe que le signe de $\widehat{\mathcal{IC}}_{\hat{F},T}(x_i, y_i)$ est le même que l'écart $\hat{\theta} - \hat{\theta}_{(i)}$.

Une autre approche pour estimer l'influence à l'aide d'un échantillon utilise la *courbe de l'influence échantillonnale* SIC_i. Alors que $\widehat{\mathcal{IC}}_{\hat{F},T}$ suppose une infinité d'observations, la SIC_i considère plutôt un échantillon de taille n.

Définition 7. Soit $x_1, \ldots, x_n$ un échantillon. On pose $X^t = (x_1^T, \ldots, x_n^T)$ et $\epsilon = -\frac{1}{n-1}$. La courbe de l'influence échantillonnale est

$$SIC_i = (n-1)(\hat{\theta} - \hat{\theta}_{(i)}) = \frac{(n-1)\left(X^T X\right)^{-1} x_i \hat{\epsilon}_i}{1 - p_{ii}}$$

où p_{ii} est le i^{e} terme diagonal de la matrice chapeau.

2.1.3.2 Influence pour la prédiction

Jusqu'à maintenant, seules les variations concernant des paramètres du modèle étaient étudiées dans nos exemples. En effet, nous avons principalement discuté des changements observables du côté de la pente de la régression. Mais qu'en est-il des prédictions ? Cette préoccupation se rapproche des explications locales discutées à la section 1.3 qui cherchent à justifier une prédiction donnée. Les développements ci-dessous étudient l'effet de l'observation (x_i, y_i) sur la prédiction d'une observation quelconque. Pour une présentation détaillée, on peut se référer à la section 4.3 de Cook et Weisberg (1982).

Posons l'observation $(x', y') \in \mathbb{R}^{m+2}$. En vertu de l'équation (2.3), sa prédiction est $\hat{y}' = x'^T \hat{\theta}$.

Comme nous connaissons déjà la dérivée $\frac{\mathrm{d}\hat{\theta}}{\mathrm{d}F}$, il est judicieux d'utiliser la dérivation en chaîne pour estimer l'influence de la i^{e} observation sur $\hat{y}'$.

Lemme 3 (Proposition 3.1.2 de Fernholz (1983))**.** *Soit F une fonction de répartition et soit S et T deux statistiques fonctionnelles dérivables au sens d'Hadamard. Alors,*

$$\frac{\mathrm{d}S \circ T}{\mathrm{d}F} = \frac{\mathrm{d}S}{\mathrm{d}T(F)} \frac{\mathrm{d}T}{\mathrm{d}F}.$$

Théorème 5. *Soit $\delta_{(x,y)}$ la fonction de répartition qui attribue toute sa masse au point (x, y). La courbe de l'influence pour la prédiction y' obtenue à l'aide de x' est*

$$\mathcal{IC}^{pred}_{F,T}(x, y, x') = x'^T \mathcal{IC}_{F,T}(x, y) \tag{2.14}$$

où T est tel que défini à l'équation (2.8)*.*

Esquisse de la preuve. Utiliser la dérivation en chaîne avec $T(F) = \hat{\theta}$ pour calculer $\frac{\mathrm{d}\hat{y'}}{\mathrm{d}F}$. On peut se référer à la section 3.2 de Ting et Brochu (2018) pour les détails. □

Ainsi, l'influence telle que calculée en (2.14) est un concept connexe au

$$DFFIT_i = \hat{y}_i - \hat{y}_{i(i)}$$

tel que défini par Belsley *et al.* (2005). Celui-ci peut être estimé à l'aide des valeurs de la matrice chapeau définie en section 2.1.2.

Ensuite, on déduit l'estimateur plug-in $\widehat{\mathcal{IC}}^{pred}_{\hat{F},T}$. On l'obtient en considérant la distribution empirique $\hat{F}$ plutôt que la distribution théorique.

Corollaire 1 (Ting et Brochu (2018))**.** *Pour une observation $(x', y') \in \mathbb{R}^{m+2}$ à prédire, on a*

$$\widehat{\mathcal{IC}}^{pred}_{\hat{F},T}(x, y, x') = x'^T \widehat{\mathcal{IC}}_{\hat{F},T}(x, y)$$
$$\widehat{\mathcal{IC}}^{pred}_{\hat{F}_{(i)},T}(x_i, y_i, x') = x'^T \widehat{\mathcal{IC}}_{\hat{F}_{(i)},T}(x_i, y_i)$$

Exemple 7 (Suite des exemples 1, 2 et 6)**.** Posons $x'^T = (1 \;\; 2,3)$ et $y' = 0,8$ Pour chacune des observations, on applique l'équation (2.14) pour $\widehat{\mathcal{IC}}^{pred}_{\hat{F},T}(x_i, y_i, x')$ et on trouve

i	1	2	3	4	5	6	7
$\widehat{\mathcal{IC}}^{pred}_{\hat{F},T}(x_i, y_i, x')$	3,355	1,875	-1,720	11,198	2,470	-5,660	-0,8056

Il est intéressant de remarquer que l'information sur l'influence obtenue avec la matrice chapeau et celle obtenue avec $\widehat{\mathcal{IC}}^{pred}_{\hat{F},T}(x_i, y_i, x')$ divergent. En effet, selon la matrice chapeau, toutes les observations sont utiles à la prédiction de $\hat{y_1}$ et l'observation 6 est la plus influente. Avec $\widehat{\mathcal{IC}}^{pred}_{\hat{F},T}(x_i, y_i, x')$, cette observation demeure influente mais dans une moindre mesure que l'observation 4.

2.2 Les fonctions d'influence en apprentissage automatique

L'introduction des FI à la communauté en IA a été faite par Koh et Liang (2017) qui reprennent la formulation générale de la section 5.2 de Cook et Weisberg (1982). Plutôt que de considérer $\widehat{\mathcal{IC}}_{\hat{F},T}$ comme nous l'avons fait jusqu'à présent, on estime plutôt la courbe de l'influence échantillonnale.[7] Cette méthode repose sur l'approximation quadratique de la fonction de perte. Pour assurer la convergence de cette approximation, on utilise les propriétés asymptotiques de l'expansion de von Mises (von Mises, 1947) qui assurent la convergence en probabilité (Fernholz, 1983). La plus importante contribution de Koh et Liang (2017) est d'avoir montré que les FI peuvent être utilisées en apprentissage automatique et qu'elles apportent de l'information pertinente concernant le comportement du modèle. On présente ces résultats à la sous-section 2.2.1. Ensuite, la sous-section 2.2.3 discute de comment les FI sont évaluées en pratique. On conclut cette section avec une brève présentation d'approches visant à améliorer l'efficacité de l'estimation des FI à la sous-section 2.2.4.

7. Pour la régression linéaire, se référer à la définition 7.

2.2.1 Variation de la fonction de perte

Soit $\mathcal{L}$ une fonction de perte strictement convexe et dérivable au sens d'Hadamard 2 fois en θ, c'est-à-dire que la matrice hessienne $\nabla^2_\theta \mathcal{L}(\theta)$ existe et est définie positive. On utilise l'expansion de von Mises (von Mises, 1947), le pendant fonctionnel de l'expansion de Taylor [8], pour l'estimer.

Définition 8 (Fernholz (1983))**.** Soit F, une fonction de répartition, et δ_z fonction de répartition attribuant toute sa masse à z. Soit T, une statistique fonctionnelle, dérivable k fois au sens d'Hadamard. Alors, l'expansion de von Mises est

$$T(F+\epsilon\delta_z) = T(F) + \left.\frac{\mathrm{d}T(F+\epsilon\delta_z)}{\mathrm{d}\epsilon}\right|_{\epsilon=0} \epsilon + \ldots + \left.\frac{\mathrm{d}^k T(F+\epsilon\delta_z)}{\mathrm{d}\epsilon^k}\right|_{\epsilon=0} \epsilon^k + R_k$$

où R_k est le reste de l'expansion de degré k.

Remarque 1. Lorsque

$$\mathbb{E}\left(\left.\frac{\mathrm{d}T(F+\epsilon\delta_z)}{\mathrm{d}\epsilon}\right|_{\epsilon=0}\right)^2 < \infty$$

et $\sqrt{n}R_k \xrightarrow{\mathrm{P}} 0$, on peut montrer, avec le théorème de Slutsky, que le reste converge vers une loi normale centrée en 0. Selon Fernholz (1983), lorsque la statistique T est dérivable k fois au sens d'Hadamard, on a $\lim_{n\to\infty} R_k = 0$. Cette propriété est essentielle pour attester de la qualité de l'approximation avec l'expansion.

Ainsi, on obtient l'approximation suivante :

$$\mathcal{L}_{(i)}(\theta) \approx \mathcal{L}_{(i)}(\hat{\theta}) + \nabla_\theta \mathcal{L}_{(i)}(\hat{\theta})^T(\theta-\hat{\theta}) + (\theta-\hat{\theta})^T \nabla^2_\theta \mathcal{L}_{(i)}(\hat{\theta})(\theta-\hat{\theta}) \tag{2.15}$$

avec $R_2 \xrightarrow{\mathrm{P}} 0$. C'est avec cette approximation que nous travaillons par la suite.

Considérons $(x_1, y_1), \ldots, (x_n, y_n)$, un échantillon issu de la distribution F, et posons $\widehat{\mathcal{L}}(\theta)$ sa

8. À noter qu'il n'est pas possible d'appliquer le théorème de Taylor sans considérations particulières aux statistiques fonctionnelles tel qu'illustré dans ce contre-exemple (whuber, 2018) où l'approximation de degré 1 a un reste divergent. Si l'on avait plutôt considéré des fonctions réelles, le reste aurait été borné par un polynôme d'ordre 2.

fonction de répartition empirique. Pour simplifier la notation, on pose $z_i = (x_i, y_i)$. Posons $\widehat{\mathcal{L}}(\theta) \overset{\text{déf}}{=} \frac{1}{n}\sum_i \mathcal{L}(\theta, x_i, y_i)$, la fonction de perte empirique, et $\hat{\theta} = \arg\min_\theta \widehat{\mathcal{L}}$, l'estimateur du maximum de vraisemblance.

Soit z une observation de l'échantillon dont on souhaite augmenter le poids infinitésimalement. Alors,

$$\hat{\theta}_{\epsilon,z} \overset{\text{déf}}{=} \arg\min_\theta \widehat{\mathcal{L}}(\theta) + \epsilon\,\mathcal{L}(\theta, z) = T(\hat{F} + \epsilon\delta_z)$$

où δ_z est la fonction de répartition de la distribution attribuant toute sa masse en z. On peut observer que $\hat{\theta}_{0,z} = \arg\min_\theta \widehat{\mathcal{L}}(\theta) = \hat{\theta}$. Puisque $\mathcal{L}$ est dérivable et strictement convexe, $\hat{\theta}_{0,z}$ est bien défini. On suppose $\hat{\theta}_{\epsilon,z}$ dérivable au sens d'Hadamard pour la suite des développements [9].

De plus, l'effet sur les paramètres d'augmenter le poids de l'observation z est

$$T(\hat{F} + \epsilon\delta_z) - T(\hat{F}) = \hat{\theta}_{\epsilon,z} - \hat{\theta}.$$

Avec l'expansion de von Mises, on a

$$\hat{\theta}_{\epsilon,z} \approx \hat{\theta} + \epsilon \left.\frac{\mathrm{d}\hat{\theta}_{\epsilon,z}}{\mathrm{d}\epsilon}\right|_{\epsilon=0} \iff \hat{\theta}_{\epsilon,z} - \hat{\theta} \approx \epsilon \left.\frac{\mathrm{d}\hat{\theta}_{\epsilon,z}}{\mathrm{d}\epsilon}\right|_{\epsilon=0}. \tag{2.16}$$

Le théorème suivant permet l'obtention d'une forme close pour la dérivée de $\hat{\theta}_{\epsilon,z}$ évaluée en $\epsilon = 0$.

Théorème 6. *Soit $\hat{H}_{\hat{\theta}} = \frac{1}{n}\sum_i \nabla^2_\theta \mathcal{L}(\theta, z_i)$, la matrice hessienne empirique, alors*

$$\left.\frac{\mathrm{d}\hat{\theta}_{\epsilon,z}}{\mathrm{d}\epsilon}\right|_{\epsilon=0} = -\hat{H}^{-1}_{\hat{\theta}}\,\nabla_\theta\,\mathcal{L}(\theta, z)|_{\theta=\hat{\theta}}\,. \tag{2.17}$$

Esquisse de la preuve. Utiliser que $\hat{\theta}_{\epsilon,z}$ est un point critique du problème d'optimisation $\min_\theta \widehat{\mathcal{L}}(\theta) + \epsilon\,\mathcal{L}(\theta, z)$ afin d'obtenir une égalité. Puis, dériver par rapport à ϵ et évaluer en 0. La preuve complète est présentée à l'annexe A.4. □

9. Cette hypothèse n'apparaît pas explicitement dans Koh et Liang (2017) ou Cook et Weisberg (1982), mais elle est implicite.

On a présenté à la section précédente que les FI peuvent estimer un Jackknife avec $\epsilon = -\frac{1}{n}$. Ceci demeure le cas dans un contexte général, d'où,

$$\hat{\theta}_{-\frac{1}{n},z} = \arg\min_{\theta} \widehat{\mathcal{L}(\theta)} - \frac{1}{n}\mathcal{L}(\theta, z) = \arg\min_{\theta} \frac{1}{n}\sum_{z_i \neq z} \mathcal{L}(\theta, z_i) = \hat{\theta}_{(z)}. \tag{2.18}$$

On déduit le résultat suivant :

Corollaire 2. *Une approximation de l'influence d'éliminer l'observation z est*

$$\hat{\theta}_{(z)} - \hat{\theta} \approx \frac{1}{n}\hat{H}_{\hat{\theta}}^{-1} \nabla_\theta \mathcal{L}(\theta, z)|_{\theta=\hat{\theta}}\,.$$

Analyser les variations des paramètres peut être utilisé pour estimer l'incertitude des paramètres (Schulam et Saria, 2019; Alaa et van der Schaar, 2020). Néanmoins, la grande quantité de paramètres des modèles en apprentissage automatique rend l'interprétation des variations brutes quasi impossible. Pour obtenir une explication locale, on reprend l'influence pour la prédiction telle que développée à la section précédente.

Théorème 7. *Soit z' une observation à prédire. L'influence de l'observation z sur la prédiction est*

$$\left.\frac{\mathrm{d}\,\mathcal{L}(z', \hat{\theta}_{\epsilon,z})}{\mathrm{d}\epsilon}\right|_{\epsilon=0} = -\nabla_\theta \mathcal{L}(z', \theta)^\top\big|_{\theta=\hat{\theta}}\ \hat{H}_{\hat{\theta}}^{-1}\ \nabla_\theta \mathcal{L}(\theta, z)|_{\theta=\hat{\theta}}\,.$$

Esquisse de la preuve. Utiliser la dérivation en chaîne puis appliquer l'équation (2.18). □

2.2.2 Robustesse des fonctions d'influence en apprentissage automatique

Pour que les résultats ci-haut soient applicables en apprentissage automatique, il est nécessaire d'évaluer s'il est possible de relâcher les contraintes suivantes :

C1 $\hat{\theta} \neq \arg\min_\theta \mathcal{L}$,

C2 $\mathcal{L}$ n'est ni convexe ni dérivable au sens d'Hadamard,

C3 θ n'est pas dérivable au sens d'Hadamard et $\hat{\theta}$ n'est pas un estimateur convergent.

L'analyse la plus complète de la robustesse des FI est présentée par Bae *et al.* (2022). Empiriquement, on peut se référer à Basu *et al.* (2020).

2.2.2.1 Contrainte **C1**

Une pratique courante lors de l'entraînement de RN est l'arrêt prématuré de l'entraînement, qui met fin à l'apprentissage lorsque que peu d'améliorations sont observée sur les données de validation. Cette pratique limite les risques de surapprentissage (Goodfellow *et al.*, 2016) mais entraîne que $\hat{\theta}$ est rarement optimal. Ainsi, il n'est plus garanti que la matrice hessienne soit définie positive. C'est pourquoi, plutôt que de considérer l'équation (2.15), on travaille avec la fonction de perte modifiée

$$\mathcal{L}_{(i)}(\theta) \approx \mathcal{L}_{(i)}(\hat{\theta}) + \nabla_\theta \mathcal{L}_{(i)}(\hat{\theta})^T(\theta - \hat{\theta}) + (\theta - \hat{\theta})^T(\nabla^2_\theta \mathcal{L}_{(i)}(\hat{\theta}) + \lambda I)(\theta - \hat{\theta}) \tag{2.19}$$

où I est la matrice identité et $\lambda > 0$ est un hyperparamètre. Ainsi, $\nabla^2_\theta \mathcal{L}_{(i)}(\hat{\theta}) + \lambda I$ est définie positive et les paramètres obtenus à l'aide de l'apprentissage $\hat{\theta}$ sont un minimum de la fonction de perte modifiée. L'argument de Koh et Liang (2017, appendice B) pour justifier théoriquement les résultats présentés à la sous-section 2.2.1 est basée sur l'algorithme de Newton [10]. De plus, les garanties sur le reste de von Mises permettent d'assurer la qualité de l'estimation. En pratique, Teso *et al.* (2021) ont observé que la hessienne est rarement définie positive malgré l'équation (2.19) et l'approximation de $\hat{\theta}$ n'est pas précise. Ainsi, il est fort probable que l'estimation des FI soit elle aussi imprécise comme elle repose sur les quantités mentionnées ci-haut.

2.2.2.2 Contrainte **C2**

Lorsque la fonction de perte n'est pas dérivable, il est coutume de l'estimer à l'aide d'une fonction dérivable. Une telle fonction non dérivable peut être une fonction tronquée telle qu'utilisée par Koh et Liang (2017, section 4.3) et conserver des bonnes performances. Fern-

10. À noter que leur développement ne considère pas l'équation (2.19) mais l'équation (2.15).

holz (1983, section 7.5) a présenté des garanties théoriques pour la troncature des densités et des estimateurs, ce qui peut expliquer les bons résultats obtenus par Koh et Liang (2017).

2.2.2.3 Contrainte **C3**

Il n'existe pas, à notre connaissance, de travaux ayant étudié la violation de cette contrainte. Ceci apparaît comme un manque dans la littérature comme la dérivabilité de la statistique T est essentielle pour garantir un estimateur convergent (Théorème 2). La notion de convergence est néanmoins abordée dans les travaux de Bae *et al.* (2022). En effet, plutôt que de supposer que $\hat{\theta}$ est un estimateur convergent de θ, leurs travaux suggèrent de changer notre interprétation des FI. Ainsi, les FI « estiment l'effet d'exclure une observation en tentant de conserver les prédictions du modèle (partiellement) entraîné. » [Notre traduction]

Bien que les résultats de Koh et Liang (2017) semblent très forts, ceux-ci ne suffisent pas à garantir la robustesse des FI pour des gros modèles. En effet, la fragilité de l'estimation des FI (Basu *et al.*, 2020) et plus particulièrement de l'inverse de la hessienne (Teso *et al.*, 2021; Zhang et Zhang, 2022) est bien documentée. Ainsi, bien que les FI demeurent utiles pour déboguer un modèle (Schioppa *et al.*, 2023), il est critique de faire preuve de vigilance lorsqu'on travaille avec ces explications locales.

2.2.3 Évaluation des fonctions d'influence

L'explicabilité étant difficile à évaluer, il n'existe pas de méthodes d'évaluation pour les FI faisant consensus dans la communauté (Zhang *et al.*, 2021). Deux méthodes semblent cependant ressortir dans la littérature. Premièrement, on peut comparer les résultats des FI à ceux obtenus en entraînant à nouveau le modèle. Cette approche suppose des liens forts entre la définition des FI (voir la définition 3) et les variations observées. Alternativement, certains travaux utilisent les FI de Koh et Liang (2017) comme point de comparaison. Ces derniers ayant introduit la méthode en IA, il s'agit d'un point de référence naturel pour les travaux tels que Yeh *et al.* (2018) et Barshan *et al.* (2020) proposant des variantes des FI. Néanmoins, les FI étant reconnues pour leur fragilité, cette approche a d'importantes

limitations.

Nous avons motivé l'influence à l'aide de la méthode Jackknife qui exclut des observations avant d'entraîner de nouveau le modèle et d'étudier les variations des paramètres ou des performances. C'est cette approche que Koh et Liang (2017), Koh *et al.* (2019) et Basu *et al.* (2020) ont utilisée. Toutefois, cette méthode est moins utilisée en pratique comme l'entraînement des modèles en apprentissage profond peut être très long. De plus, la fiabilité de cette approche pour les RN est discutable comme étudié dans la sous-section 2.2.2 et dans les travaux de Bae *et al.* (2022). C'est pourquoi Guo *et al.* (2021) ont plutôt affiné à l'aide des exemples influents puis analysé les variations de la fonction de perte. Ainsi, les modèles qui sont affinés avec les exemples néfastes enregistrent des hausses de la fonction de perte tandis que des baisses sont observées lorsque les modèles sont affinés avec les exemples utiles.

Dans un autre ordre d'idées, on peut remarquer que les commentaires posés sur l'influence aux exemples 6 et 7 étaient toujours en comparaison avec d'autres mesures de l'influence. C'est l'approche utilisée par Pezeshkpour *et al.* (2021) pour évaluer différentes méthodes estimant l'influence. En effet, une approche sera considérée comme une bonne approximation lorsque sa corrélation de rang avec l'influence de Koh et Liang (2017) est forte.

2.2.4 Accélérer les calculs

La grande quantité de calculs nécessaires à l'estimation des FI est le principal frein à leur adoption en explicabilité. En effet, l'estimation de l'inverse de la hessienne est très coûteuse pour les modèles en apprentissage automatique en raison de la grande quantité de paramètres (Yeh *et al.*, 2018; Zhang et Zhang, 2022). Par exemple, avec la formulation de Koh et Liang (2017), évaluer les FI d'un modèle T5-base (Raffel *et al.*, 2020) nécessite l'inversion d'un matrice de dimensions $2,2 \times 10^8$ par $2,2 \times 10^8$. De plus, cette inversion doit être effectuée pour chacune des prédictions dont on cherche à connaître les exemples d'entraînement influents.

FastIF (Guo *et al.*, 2021) propose des accélérations substantielles grâce à un travail d'ingénie-

rie combinant des algorithmes d'optimisation (Agarwal *et al.*, 2017) et de recherche (Johnson *et al.*, 2019). En plus d'obtenir des bonnes approximations des FI, cette solution peut réduire les temps de calcul par un facteur allant jusqu'à 80. Ces accélérations sont notamment obtenues en estimant itérativement le produit en l'équation (2.17) à l'aide de mini lots (Agarwal *et al.*, 2017). De ce fait, l'inverse de la hessienne n'est jamais explicitement calculée. De plus, Guo *et al.* (2021) suggèrent de seulement estimer l'influence des exemples d'entraînement à proximité de l'observation à expliquer dans l'espace de représentation. Cette sélection permet d'extraire les données d'entraînement propices à être influentes et comprendre le comportement local du modèle. Cette recherche est effectuée à l'aide de l'algorithme des K plus proches voisins (Fix et Hodges Jr, 1952). Schioppa *et al.* (2022) ont obtenu des accélérations supérieures en considérant plutôt les itérations d'Arnoldi (1951) pour l'estimation implicite de la hessienne.

Pour accélérer les calculs, plusieurs approches telle que les *representer points* (Yeh *et al.*, 2018; Sui *et al.*, 2021) ont été proposées. Cette méthode estime linéairement la prédiction à l'aide des attributs dans l'espace de représentation de chacun des exemples d'entraînement. Enfin, les coefficents de la combinaison linéaire constituent l'approximation de l'influence. Lorsque les attributs considérés sont complexes, cette méthode inspirée du *representer theorem* (Schölkopf *et al.*, 2001) est très efficace. Néanmoins, Pezeshkpour *et al.* (2021) ont observé que les approximations de l'influence avec les *representer points* sont peu corrélées avec les FI de Koh et Liang (2017).

En somme, l'influence est la dérivée d'une statistique fonctionnelle telle que la fonction de perte, les paramètres et la prédiction d'une observation. Elle permet d'estimer les variations de cette dernière causées par des changements infinitésimaux concernant les données d'entraînement. L'analyse des variations de la fonction de perte causées par les données d'entraînement permet d'améliorer les performances du modèles en désapprenant (Wu *et al.*, 2022) ou en excluant les observations néfastes. Ces dernières comprennent les observations mal étiquetées (Koh et Liang, 2017). De plus, l'information obtenue à l'aide des FI permet notamment d'informer sur l'importance des observations lors de l'apprentissage du modèle (Koh

et Liang, 2017; Guo *et al.*, 2021; Rancourt *et al.*, 2022), l'incertitude des paramètres (Schulam et Saria, 2019; Alaa et van der Schaar, 2020) et les artefacts appris par le modèle (Han *et al.*, 2020). Ainsi, les FI apportent de l'information pertinente pour comprendre le comportement du modèle (Ghorbani *et al.*, 2019; Meng *et al.*, 2020; Bhatt *et al.*, 2020b).

Toutefois, l'adoption des FI est limitée en raison de ses importants besoins en calcul. En effet, l'attribution des exemples doit considérer, pour chacune des prédictions, l'ensemble des données d'entraînement. De plus, il est très coûteux d'estimer l'inverse de la matrice hessienne, une composante essentielle des FI.

CHAPITRE 3
SUR L'INFLUENCE DE LA QUALITÉ D'ANNOTATION POUR L'ÉVALUATION DU RISQUE DE SUICIDE

3.1 Contexte et références

Ces travaux consistent en une application des FI afin d'explorer la relation d'une variable latente des données d'entraînement, l'expertise en santé mentale de la personne annotatrice, sur la prédiction du risque de suicide à l'aide d'un modèle de type RoBERTa (Liu *et al.*, 2019). Les données considérées pour ces travaux sont issues de la tâche partagée CLPsych 2019 (Zirikly *et al.*, 2019) et sont notamment constituées de publications extraites de la communauté r/SuicideWatch du réseau social Reddit. Comme chaque usager.ère a exactement une estimation de leur risque de suicide, cette classe est attribuée à chacune de leurs publications. En plus de simplifier l'architecture du modèle en réduisant la longueur des contenus textuels considérés, ce changement n'affecte qu'une faible proportion des usager.ères : seulement 27% comptent plus d'une publication. Pour chacune des prédictions de l'ensemble de test, on estime l'influence de l'ensemble des données d'entraînement avec la librairie FastIF (Guo *et al.*, 2021), qui permet d'accélérer grandement ces calculs tout en conservant une bonne exactitude. Les résultats obtenus suggèrent que les données obtenues en impliquant la production participative d'un large public non-spécialiste ont une valeur substantielle bien qu'inférieure à celles annotées par des expert.es. Ceci est cohérent avec les conclusions de Shing *et al.* (2018).

L'article joint [1] en section 3.2 a été présenté par l'autrice de ce mémoire à la conférence *35th Canadian Conference on Artificial Intelligence* qui a eu lieu du 30 mai au 3 juin 2022 à Toronto, Canada. Le manuscrit est disponible en libre accès dans les actes de la conférence (Rancourt *et al.*, 2022). Le code source associé aux travaux est disponible publiquement via un dépôt gitlab (Rancourt et Maupomé, 2022). L'accès aux données doit être approuvé par le comité organisateur de la tâche partagée CLPsych.

1. Le format a été adapté pour se conformer à celui de ce mémoire.

L'autrice de ce mémoire a formulé les hypothèses, conçu le protocole expérimental et a participé à son exécution en collaboration avec le deuxième auteur. Enfin, elle a contribué à la majeure partie de la rédaction de la publication ci-dessous.

3.2 Article

On the Influence of Annotation Quality in Suicidal Risk Assessment from Text

Fanny Rancourt, Diego Maupomé, Marie-Jean Meurs
Université du Québec à Montréal

Abstract

In applying Natural Language Processing to support mental health care, gathering annotated data is difficult. Recent work has pointed to lapses in approximative annotation schemes. While studying gaps in prediction accuracy can offer some information about these lapses, a more careful look is needed. Through the use of Influence Functions, quantification of the relevance of training examples according to their type of annotation is possible. Using a corpus aimed at suicidal risk assessment containing both crowdsourced and expert annotations, we examine the effects that these annotations have on model training at test time. Our results indicate that, while expert annotations are more helpful, the difference with respect to crowdsourced annotations is slight. Moreover, most globally helpful observations are crowdsourced, pointing to their potential.

3.2.1 Introduction

Recently, there has been increased research interest in utilizing Natural Language Processing (NLP) techniques in the service of mental health care. Mental health is a major public health issue, accounting for 13% of the global burden of disease (James *et al.*, 2018), leading to higher rates of morbity and mortality (Saxena, 2018). Thus, early intervention in mental health has become a key issue in service reform (Schotanus-Dijkstra *et al.*, 2017; McGorry et Mei, 2018).

NLP can play an important part in this aspect of care by analyzing textual content from social media, widely used by at-risk persons to discuss their mental state (Shing *et al.*, 2020). NLP research efforts will attempt to infer models that predict the mental health status of a person given their writings on various online social media. This assessment will usually pertain to specific mental health disorders, symptoms or harmful behaviors. However, there is large variation in the nature of these assessments. Clinically grounded assessments are costly. Where annotation in other NLP tasks can be carried out on the documents to be analyzed, in the context of mental health, annotation pertains to the author, and clinically grounded annotation requires access to this person. This makes the data collection process difficult. To boot, sophisticated prediction models often require large amounts of training data. This has given rise to a variety of what could be understood as approximations to clinical truth. These vary from the use of affiliation to specific fora or groups (Chancellor *et al.*, 2016) and mentions of diagnosis (Coppersmith *et al.*, 2014) to the use of clinical self-report tools (Losada *et al.*, 2020).

However, there is concern about the validity of these approaches. That is, while models can be developed to accurately predict these assessments on unseen data, it is possible that these assessments capture a different construct than the desired ones, *i.e.* the clinically actionable aspects of mental health of interest (Chancellor et De Choudhury, 2020). Evaluating the predictive performance of models issuing from these annotation schemes on clinically grounded data can offer some insight into this phenomenon (Ernala *et al.*, 2019). Nonetheless, a closer examination of the inner workings of these models may offer richer information as to whether these annotations remain disjointed from clinically sounder ones.

In addition to explanations aiming at external stakeholders (Bhatt *et al.*, 2020a), explainability techniques can provide useful insights for the development and deployment of machine learning models (Bhatt *et al.*, 2020b; Guo *et al.*, 2021). In NLP, this can take the form of feature importance (Ribeiro *et al.*, 2016; Lundberg et Lee, 2017) and saliency maps (Simonyan *et al.*, 2014) techniques, which can give insights regarding words or spans from the input text. Alternatively, methods explaining with examples (Lipton, 2018) such as Influence

Functions (IF) can give valuable insights on such complex tasks. In fact, evidence suggests that IF are more appropriate explanations than saliency maps for non trivial NLP tasks such as language understanding (Han *et al.*, 2020). Risk assessment of mental health issues from social media content is such a complex classification task as the labels usually require additional information about the authors. The work presented in this paper seeks to study the use of IF in mental health assessment by NLP. Specifically, its goals are to :

RQ1 Examine the impact of crowdsourced annotations on model predictions.

RQ2 Assess whether crowdsourced annotations are globally more influential than expert annotations.

The paper is structured as follows. Section 3.2.2 defines influence functions and Section 3.2.3 discusses the methodology used to conduct our experiments. Finally, Section 3.2.4 discusses the results, and Section 3.2.5 concludes this paper with potential future work.

3.2.2 Influence Functions

Influence Functions (IF) – a classic notion from robust statistics – monitor the changes occurring after small modifications to the problem formulation (Cook et Weisberg, 1982). This analysis rely on the hypothesis that slight perturbations should cause at most small variations on the results of a model or a statistical test (Huber, 1981). The most popular of said perturbations regards the distribution of the observations.

Given a machine learning model, IF aim to answer the following question : *what if we had a different training set ?* By infinitesimally up-weighting an observation z, IF allow practitioners to assess the influence of this data point on a given prediction : a large loss variation indicates that z is *influential*. From an explainability standpoint, the sign of the loss variation indicates whether z is helpful or not for the prediction. Moreover, one can also probe local robustness by monitoring parameter change (Koh et Liang, 2017). Other alternatives, such as assessing the stability of the predictions between both sets of parameter values, can also be considered. Thus, IF can improve algorithmic transparency as defined by Lipton (2018) and can indicate the presence of predictive uncertainty (Schulam et Saria, 2019) More

formally, let $\mathcal{D} = \{(\mathbf{x}_k, y_k) : 1 \leq k \leq n\}$ be the training data of a classic supervised task. The learned parameters $\hat{\theta}$ are obtained by resolving optimization problem $\arg\min_\theta \sum_k \mathcal{L}(z_k, \theta)$, where $\mathcal{L}$ is the loss function combined with regularization factors if applicable. To assess the influence of $z_i \in \mathcal{D}$, we consider the parameter change occurred when up-weighting it by ϵ, *i.e.* $\widehat{\theta}_{\epsilon,i} := \arg\min_\theta \sum_k \mathcal{L}(z_k, \theta) + \epsilon\, \mathcal{L}(z_i, \theta)$. The parameter change is approximately

$$\widehat{\theta}_{\epsilon,i} - \hat{\theta} \approx \epsilon \left.\frac{\mathrm{d}\,\widehat{\theta}_{\epsilon,i}}{\mathrm{d}\epsilon}\right|_{\epsilon=0} = -\epsilon H_{\hat{\theta}}^{-1}\, \nabla_\theta\, \mathcal{L}(z_i, \hat{\theta}) \tag{3.1}$$

where $H_{\hat{\theta}}^{-1} := \frac{1}{n} \sum_k \nabla_\theta^2\, \mathcal{L}(\hat{\theta}, z_k)$. Calculation details of the right term in (3.1) are provided in Section 5.2 of (Cook et Weisberg, 1982). Some properties of this estimator are discussed in (Fernholz, 1983). To assess whether an observation z_i is helpful or not to predict another observation $\tilde{z}$, we refer to the variation of the loss function

$$\mathcal{I}(\tilde{z}, i) := \left.\frac{\mathrm{d}\,\mathcal{L}(\tilde{z}, \widehat{\theta}_{\epsilon,i})}{\mathrm{d}\epsilon}\right|_{\epsilon=0} = -\nabla_\theta\, \mathcal{L}(\tilde{z}, \hat{\theta})^\top H_{\hat{\theta}}^{-1}\, \nabla_\theta\, \mathcal{L}(z_i, \hat{\theta}) \tag{3.2}$$

obtained with the chain rule (Koh et Liang, 2017). For (3.1) and (3.2) to hold, the loss function must be twice differentiable and convex, assumptions that most machine learning models do not uphold. Nonetheless, this approximation can remain fairly accurate for Transformer models (Han *et al.*, 2020; Guo *et al.*, 2021) and smaller neural architectures (Koh et Liang, 2017; Basu *et al.*, 2020).

3.2.3 Methodology and Resources

3.2.3.1 Influence Functions

IF are particularly useful for error analysis as they bring forth "insights about how models rely on and extrapolate from the training data" (Koh et Liang, 2017). Hence, we use IF to examine prediction errors and assess whether there is a correlation with the annotator of those influential (helpful or harmful) examples with respect to (3.2). Additionally, IF can give insights regarding latent structures from the training set (Guo *et al.*, 2021). To examine **RQ2**, the average influence as well as globally influential examples are considered. A data

Tableau 3.1 – Distribution of risk levels of each post according to their annotator

	training set		validation set		test set	
Risk level	crowdsourced	expert	crowdsourced	expert	crowdsourced	expert
None	130	28	32	8	34	9
Low	48	47	11	15	13	15
Moderate	124	99	30	31	41	32
Severe	436	57	108	18	98	18

point is *globally influential* when the absolute value of its influence ranks among the highest 25% for at least half the test examples. As the dominant class comprises 44% of the test set (see Table 3.1), this guarantees that these examples are influential for multiple risk levels. FastIF (Guo *et al.*, 2021) is used to compute influence functions. A damping parameter of 5E-3 is added to the diagonal of $H_{\hat{\theta}}$ to ensure that all the eigenvalues are positive.

3.2.3.2 Data

The experiments conducted concern suicidal risk assessment using data from CLPsych (Shing *et al.*, 2018; Zirikly *et al.*, 2019). The data consists of posts written on the Reddit forum r/SuicideWatch, which aims to support peers struggling with suicidal ideation [2]. The suicidal risk estimated for each author is placed on a four-point scale – None, Low, Moderate, and Severe. The task is modified slightly for the purposes of this investigation : each post is treated as a single observation, using the label of its author as its own. The task then becomes one of document classification, simplifying the architecture. While some authors span several posts, 73% count exactly one post. Further, given that annotation was performed on the basis of these documents, this simplification remains sound.

Lastly, the original dataset is divided between observations annotated by crowdsource workers and experts. As such one additional modification was made to it : whereas expert-annotated examples were used only in testing by Shing *et al.* (2018), expert-annotated examples were used in training in the present work so as to be assess their influence. Specifically, expert annotations are spread into the training, validation and test sets at rates of 60%, 20% and

2. This corresponds to CLPsych 2019 Task A v2.

20%, respecting label proportions. Table 3.1 presents the distribution of risk levels among different annotation sources. For further details regarding the annotation guidelines and process, see Shing *et al.* (2018). Word counts for each set are presented in Table 3.5 (see Appendix 3.2.7).

3.2.3.3 Model

The classifier is an adaptation of the RoBERTa model (Liu *et al.*, 2019). In order to keep the model small, the parameters of all but the topmost Self-Attention layers are left fixed. To mitigate class imbalance, observations are weighted in inverse proportion to the weight of their class. The model is trained over 30 epochs with mini-batches of 32 observations, using the Adam optimizer ($\beta_1 = .9, \beta_2 = .999$) (Kingma et Ba, 2014) with a learning rate of 5E-5 and a weight decay of 1E-5. At the end of each epoch, the model is evaluated in order to select the best-performing point. This evaluation is done on a randomly selected validation set with equal proportions of each of the four classes. The selection is based on the macro-aggregated f1-score.

3.2.4 Results

Models are evaluated using macro-averaged precision, recall and f1-score, counteracting label imbalance. Classification results are presented in Table 3.2. Furthermore, to assess the relevance of the classification approach, a second model was trained with the same configurations using the training and test sets from Zirikly *et al.* (2019). At test time, the highest-risk label predicted for a document becomes the predicted label for its author. Results are presented in Table 3.4.

3.2.4.1 **RQ1**

As our results show, the Moderate risk category appears to be poorly captured by our model. This label is seldom predicted, and most Moderate risk observations are assigned Severe risk instead (see Table 3.3). Furthermore, throughout misclassified Moderate risk examples, the

Tableau 3.2 – Test results on CLPsych data combining expert and crowdsourced annotations for training and test sets

Risk level	precision	recall	f1-score
None	.59	.77	.67
Low	.32	.43	.36
Moderate	.30	.18	.22
Severe	.51	.53	.52
Macro-avg	.43	.48	.45

Tableau 3.3 – Confusion matrix of test set predictions with respect to annotation source

	crowdsourced				expert			
	None	Low	Moderate	Severe	None	Low	Moderate	Severe
None	25	1	2	6	8	0	1	0
Low	1	6	1	5	3	6	2	4
Moderate	4	9	9	19	0	2	4	26
Severe	15	12	18	53	0	2	7	9

majority of highly helpful examples are from the Severe class. Of those, 88% were labeled by crowdsourced annotators. This could indicate the presence of mislabeled data : Moderate risk being classified as Severe by crowdsourced annotators. This is reflected by the higher frequency of Severe risk in crowdsourced annotation, as compared to expert annotation (see Table 3.1), and is consistent with previous findings (Shing *et al.*, 2018).

While error rates are similar between annotators, the model is more likely to underestimate the risk level of crowdsourced annotated data. In particular, 15% of Severe risk crowdsourced-annotated data are classified as No risk while none of the expert-annotated ones are. Further, the prediction of Moderate risk data follows a similar pattern. This is problematic for triage-based applications, given that a high recall for high risk documents is of utmost importance.

3.2.4.2 **RQ2**

Among training examples found to be helpful to classification, expert-annotated ones had higher influence on average than examples issuing from crowdsourced annotation (0.673 vs 0.561). In contrast, harmful examples from each annotation source had comparable influence (0.441 vs 0.443). This further suggests that expert annotations are of greater quality as fine-

Tableau 3.4 – Results obtained training and testing only on crowdsourced annotations against best results at CLPsych2019

	RoBERTa			CLaC (Mohammadi *et al.*, 2019)
Risk level	precision	recall	f1-score	f1-score
None	.86	.75	.80	.74
Low	.38	.38	.38	.24
Moderate	.37	.46	.41	.40
Severe	.63	.60	.61	.54
Macro-avg	.56	.55	.55	.48

tuning on them would likely improve the model performances (Guo *et al.*, 2021). Nonetheless, the gap between annotation types is modest, which indicates that crowdsourced annotations have value. Additionally, most globally helpful examples are labeled as Severe risk, 14% of which were expert-annotated. Thus, crowdsourced annotation actively contributes to improve our model. In contrast, the examples most harmful to classification are labeled as Moderate risk, with the balance falling slightly to crowdsourced annotation.

3.2.5 Conclusion and Future work

Our experiments demonstrate the potential of IF in analyzing the effects of annotation quality on model predictions in suicidal risk assessment. Given, the importance of establishing the soundness of annotation in mental health applications, this area warrants further study. Error correction could be applied in order to improve on our results, particularly for low- and moderate- risk examples. Additional improvements could be made by considering pretrained language models trained on domain-specific data or using domain adaptation.

3.2.6 Acknowledgements

FR would like to thank Frédéric Branchaud-Charron for insightful discussions, and we also thank Dr. Leila Kosseim for her support. This research was enabled by Calcul Québec resources. MJM acknowledges the support of the Natural Sciences and Engineering Research Council of Canada [NSERC Grant number 06487-2017] and the Government of Canada's New Frontiers in Research Fund (NFRF), [NFRFE-2018-00484].

3.2.7 Appendix : Data statistics

Tableau 3.5 – Mean and standard deviation of word counts

	training set		validation set		test set	
Risk level	mean	std	mean	std	mean	std
None	146	172	160	217	154	183
Low	268	306	361	415	211	184
Moderate	235	255	272	273	210	190
Severe	213	242	174	203	251	211

CHAPITRE 4
INVESTIGATION DES MODÈLES AUTO-JUSTIFIANT

4.1 Contexte et références

Ces travaux consistent en une analyse de modèles autojustifiants qui classifient et génèrent une explication. Ce type d'approche figure parmi les plus prometteuses en NLP explicable à ce jour (Wiegreffe *et al.*, 2021). Pour nos travaux, deux types d'explications sont considérées : le surlignage et les explications ouvertes [1]. Nos analyses considèrent deux ensembles de données populaires en TALN explicable : Cos-E (Rajani *et al.*, 2019), une tâche de raisonnement basé sur le sens commun, et e-SNLI (Camburu *et al.*, 2018), une tâche de RIT. Toutes les explications ont été produites par des humains. Les expériences ont été exécutées à l'aide du modèle génératif T5 (Raffel *et al.*, 2020). Ce modèle pré-entraîné est très flexible comme il peut être affiné pour la classification ainsi que pour la génération des explications mentionnées ci-haut (Narang *et al.*, 2020). Ainsi, il est possible d'analyser des modèles autojustifiants reposant sur la même architecture. Nos résultats mettent en doute la validité des explications générées par le modèle. En effet, il est fréquent que les surlignages générés ne soient pas un sous-ensemble du contenu en entrée, ce qui contredit la littérature (Narang *et al.*, 2020). De plus, beaucoup d'explications ouvertes générées par le modèle suivent un gabarit, indiquant un manque d'expressivité. Une cause possible de ce comportement est la façon dont ces dernières sont générées par le modèle (Vijayakumar *et al.*, 2020). Comme les gabarits identifiés sont peu fréquents dans les données d'entraînement, nous avions supposé que ces exemples seraient très influents. Un constat similaire a pu être observé pour la RIT (Han *et al.*, 2020). Néanmoins, notre analyse des FI n'a pas décelé d'association entre l'influence et la présence du gabarit dans l'annotation.

Le manuscrit joint [2] en section 4.2 a été accepté pour publication dans le journal scientifique *Stats*. Le manuscrit est disponible en libre accès (Rancourt *et al.*, 2023). Le code source

1. Se référer à la section 1.3 pour une présentation détaillée.
2. Le format a été adapté pour se conformer à celui de ce mémoire.

associé aux travaux est disponible publiquement via un dépôt gitlab (Rancourt et Maupomé, 2023). Tous les ensembles de données sont disponibles via la librairie `datasets` de HuggingFace (Lhoest *et al.*, 2021).

L'autrice de ce mémoire a contribué à la formulation des hypothèses, à la conception du protocole expérimental, à la programmation et à l'analyse des résultats. Enfin, l'autrice de ce mémoire a rédigé la publication ci-dessous en collaboration avec les co-auteur.trices.

4.2 Article

Investigating Self-Rationalizing Models for Commonsense Reasoning

Fanny Rancourt, Paula Vondrlik, Diego Maupomé, Marie-Jean Meurs
Université du Québec à Montréal

Abstract

The rise of explainable natural language processing spurred a bulk of work on datasets augmented with human explanations, as well as technical approaches to leverage them. Notably, generative large language models offer new possibilities, as they can output a prediction as well as an explanation in natural language. This work investigates the capabilities of fine-tuned text-to-text transfer Transformer (T5) models for commonsense reasoning and explanation generation. Our experiments suggest that while self-rationalizing models achieve interesting results, a significant gap remains : classifiers consistently outperformed self-rationalizing models, and a substantial fraction of model-generated explanations are not valid. Furthermore, training with expressive free-text explanations substantially altered the inner representation of the model, suggesting that they supplied additional information and may bridge the knowledge gap. Our code is publicly available, and the experiments were run on open-access datasets, hence allowing full reproducibility.

4.2.1 Introduction

Over recent years, the increased predictive power of Artificial Intelligence (AI) methods has heralded the practical deployment of AI systems in various sectors of human activity. In turn, this has renewed research interest in the relationship between these systems and their human operators. A key component of this relationship is trust. Inadequate trust dynamics between automated systems and their operator can result in suboptimal performance. This is evidenced by the fact that overt human reliance on automated systems can result in poor performance. Trust in AI systems, however, is not merely a matter of reliability, *i.e.* decreased error. Instead, favorable trust dynamics are best served by frameworks allowing the appropriate calibration of trust between human and autonomous agents (Lyons *et al.*, 2017; Nor *et al.*, 2022). Improving the ability of humans to calibrate trust in automated tools is a matter of transparency : human operators should understand the inner workings of the tool in question either in general or for a given prediction (Dzindolet *et al.*, 2003; Mercado *et al.*, 2016).

Exercising transparency is a key concern in explainable IA (XAI) research, which seeks to make AI methods and their models more interpretable by human observers in order to maintain intellectual oversight on them (Héder, 2023; Hulsen, 2023). Although XAI research concerns a variety of types of data and fields of application, the present work focuses on explainability in Natural Language Processing (NLP) methods. Modern NLP methods leverage heavy representation learning in order to grapple with the irregularities of human language, making explainability in NLP a burgeoning field of research, with several avenues for development. One such avenue is the use of models explicitly trained to provide textual explanations for their predictions (Wiegreffe et Marasović, 2021; Rajani *et al.*, 2019; Camburu *et al.*, 2018). Indeed, models can be trained to replicate explanations provided by human annotators for individual data points, much in the same manner they are trained for the task of interest. Thus, models need not be structured in an inherently interpretable manner but may extract patterns of explanation from the data. There are several limitations to such a framework. A primary concern is whether model-generated explanations indeed reflect the decision-making process of the model—are *faithful*—or shallowly replicate explanation

formulae. It has been argued that this can be achieved by imposing structural constraints on models (Wiegreffe *et al.*, 2021). For example, models can be structured to make predictions based on explanations, thus becoming faithful by construction (Jain *et al.*, 2020). Another concern is how these explanations might alter the learning of the primary task they intend to explain. Continuing the previous example, an inaccurate explanation may hinder the ability of a model to produce an accurate prediction. To the contrary, training with explanations has been suggested to help task acquisition in some instances by providing supplementary information. This could be the case for tasks such as commonsense reasoning and natural language inference, where the input is not sufficient to complete the task and world knowledge is needed to bridge the gap (Wiegreffe *et al.*, 2021).

The present work concerns itself with *self-rationalizing* models, which jointly produce predictions and their associated explanations, achieving promising results (Narang *et al.*, 2020). In doing so, self-rationalizing models do not constrain their prediction and explanation components to rely directly on the output of one another. However, it is unclear how considering explanations in the training process affects a model. For example, gaps in prediction quality between self-rationalizing and prediction-only models have been reported (Wiegreffe *et al.*, 2021). The present work seeks to further study this issue.

We begin by measuring the effects of explanations and their type on performance in commonsense reasoning and natural language inference tasks. In order to make an equitable comparison between approaches with different explanation configurations, we unite them under a flexible common base, the text-to-text transfer Transformer (T5) model (Raffel *et al.*, 2020). This allows the treatment of the task target and accompanying explanation as a single target sequence. Our results indicate, in agreement with previous work, that explanations both in excerpts and free-text hurt prediction performance. This stands in contrast with human learning, where self-explanation *helps* learning (Hoffman *et al.*, 2018).

Subsequently, we examine whether the presence of explanations causes models to rely on data artifacts, as has been observed in Natural Language Inference (NLI) tasks (Han *et al.*, 2020).

Model-generated free-text explanations tend to be structured (Wiegreffe et Marasović, 2021) and appear to rely on a handful of rare formulae from the training data. Lastly, we investigate whether the presence and type of explanations considered in training alter the influence that individual examples may have on the training process, specifically, which examples become more influential. We observe that some types of explanations have significant effect on the inner representation of the model.

This paper is structured as follows. Section 4.2.2 discusses human explanations and their usage in explainable NLP. Section 4.2.3 provides technical information on our experiments. Section 4.2.4 presents our results and Section 4.2.5 concludes this paper.

4.2.2 Background

4.2.2.1 Explainable NLP

Many approaches and models in modern NLP are not built with interpretability in mind. Therefore, much work has been devoted to developing means of explaining model behavior *a posteriori*, by examining the models themselves or their predictions. For example, one can employ feature attribution (Ribeiro *et al.*, 2016; Simonyan *et al.*, 2014) or instance attribution methods (Koh et Liang, 2017) to probe models for the significance of specific features or observations. Nonetheless, the faithfulness of these *post-hoc* explanations can vary (Jacovi et Goldberg, 2020; Pezeshkpour *et al.*, 2021). Similarly, a widespread approach in NLP is to interrogate the values produced by the attention mechanism (Bahdanau *et al.*, 2015)—a ubiquitous content-based weighting of concurrent parts, *e.g.* the words in a sentence. Simply put, the distribution of attention can be attributed an explanatory value : parts receiving larger attention values are deemed to play a larger role in a prediction. However, the soundness of this attribution is debated (Bibal *et al.*, 2022; Bastings et Filippova, 2020; Wiegreffe et Pinter, 2019; Jain et Wallace, 2019).

Rather than probing models *a posteriori* in order to elucidate their inner workings, one can constrain models to produce explanations for their predictions. That is, models are built to accompany their output with a human-interpretable justification. These explanations

justify an individual occurrence rather than teach generalized theories, the latter being out of scope for most of the explainability literature. In NLP, these explanations—or *rationales*—are text sequences. We use the terms *rationales* and *explanations* interchangeably in this work, referring the reader to Jacovi et Goldberg (2020) for a discussion on the nuances of both terms. There exist several competing notions regarding the use of rationales, namely, surrounding the constraints on explanations and their relationship to the output. Indeed, the explanations to be produced can be constrained in their structure in different manners. Similarly, different dependencies can be established between the output and accompanying rationale. For example, a model can be trained to produce its output as a function of the rationale or vice versa. These choices in model and explanation constraints articulate different considerations regarding the role of explanations in prediction.

In any case, models are *trained* to produce these explanations based on supervision by human-generated explanations. This has the advantage of favoring model explanations that are adequate by construction. Of course, training models on human-generated explanations exposes the framework to the quality of explanation data, which may be difficult to verify. Further, it requires an understanding of the nature of human explanations in order to gain perspective on model behavior. Lastly, such a framework assumes that explanations are intended to justify a prediction to a human observer rather than to teach them what the model has inferred. In other words, the model is the explainee, rather than the user. The latter paradigm is out of the scope of the present work, which focuses on commonsense reasoning.

4.2.2.2 Human Explanations

Human explanations tend to be contrastive in order to fit the decision border (Miller, 2019). Thus, it is natural for an explanation of a classification prediction to not be self-contained, referring instead to other classes. This is particularly true for commonsense reasoning tasks with *distractor* choices : explanations tend to highlight why these distractors are unreasonable alternatives (Rajani *et al.*, 2019). Further, explanations are also a *co-adaptive* process in which the explainer and the explainee collaborate to obtain a satisfactory explanation (Hoffman

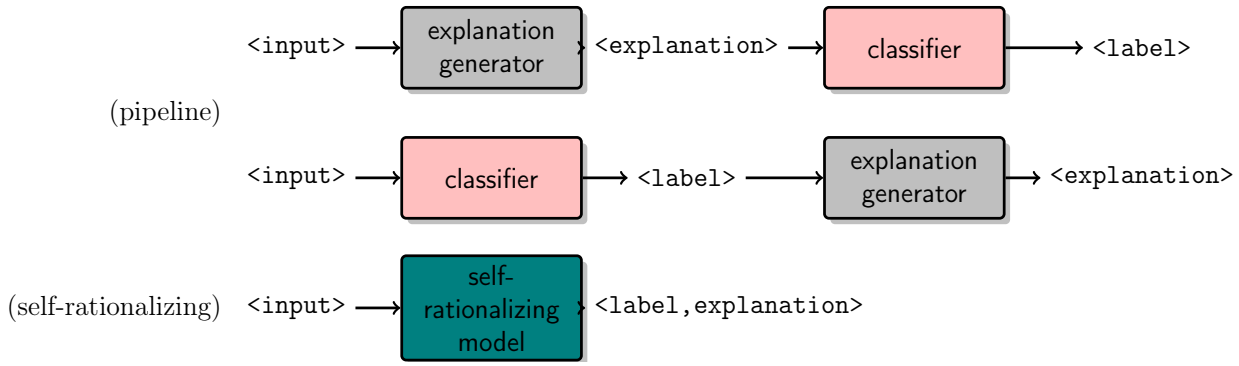


Figure 4.1 – Two alternative paradigms of explainable NLP models in a classification example. Pipeline models produce explanations and outputs through separate, sequentially dependent components. Self-Rationalizing models produce outputs and explanations jointly.

et al., 2018). Since humans have a limited capacity to process information, they tend to select simple explanations that are less specific and cite fewer causes over plausible ones (Miller, 2019). As explanations are seldom self-contained, they are social : the explainer states what is necessary and the explainee leverages their knowledge of the situation to contextualize the explanation (Miller, 2019). While earlier work in explainability focused on delivering one explanation, advances in reinforcement learning with human feedback is a promising avenue for Large Language Model (LLM) to align with user preferences (Ouyang *et al.*, 2022) thus approaching a co-adaptive process.

4.2.2.3 Rationales

It is well established that humans can improve their learning and understanding of a given context through self-explanation (Hoffman *et al.*, 2018). However, it is unclear if this holds for machine learning models. Differences can arise from the format constraints of rationales. The two most prominent choices are highlights and free-text rationales. A *highlight* is a (non-contiguous) subsequence of the input text. These excerpts are the key portions of the input supporting the prediction. As such, they constitute extractive explanations. In contrast, abstractive explanations produce new text intended to explain a decision. Namely, a *free-text explanation* is one written in natural language without restricting its format. This terminology is borrowed from Wiegreffe et Marasović (2021). Previous results suggest that models considering human-generated highlights achieve higher accuracy (Mathew *et al.*, 2021), re-

quire fewer observations to achieve similar performance (Zaidan *et al.*, 2007), and improve model explanations (Strout *et al.*, 2019). For free-text explanations, strong results can be achieved through a generative pre-trained Transformer (GPT) (Radford *et al.*, 2018) model fine-tuned on human-annotated explanations (Rajani *et al.*, 2019) while a gap subsists for T5 (Wiegreffe *et al.*, 2021).

Explanations also benefit data quality : the need to explain a decision tends to improve its accuracy (McDonnell *et al.*, 2016). Although the time required for annotation with both label and explanation is greater at first, it largely decreases over time to nearly equal the time required to annotate labels only (Kutlu *et al.*, 2020). Furthermore, when restricting explanation to highlights, strong inter-annotator agreements can be observed (Zaidan *et al.*, 2007) which can in turn be easily monitored to insure explanation quality.

4.2.2.4 Self-Rationalizing Models

The production of datasets augmented with human-generated explanations gained prominence recently (McDonnell *et al.*, 2016; Kutlu *et al.*, 2020), enabling explainable NLP work. Indeed, two paradigms emerged : *pipeline models* (see DeYoung *et al.*, 2020)—one model generates the decision and another the explanation—and *self-rationalizing models* (Rajani *et al.*, 2019; Narang *et al.*, 2020)—which jointly output both. This definition is adapted from the work of Wiegreffe *et al.* (2021). These approaches are illustrated in Figure 4.1. Multiple configurations can be used for pipeline models : *input* $\rightarrow$ *output* $\rightarrow$ *explanation*—any *post-hoc* explainability method (Ribeiro *et al.*, 2016; Simonyan *et al.*, 2014; Koh et Liang, 2017) can provide a plausible explanation, but its faithfulness is not guaranteed (Jacovi et Goldberg, 2020)—and *input* $\rightarrow$ *explanation* $\rightarrow$ *output*—deemed faithful by construction as the first model acts as an "evidence extractor" and the prediction model can only rely on this evidence. The main limitation of this latter approach is that the quality of the first model may limit the accuracy of the whole (Jacovi et Goldberg, 2021).

In contrast, self-rationalizing models produce labels and explanations jointly. Because they generate the label and an explanation from the same representation, this explanation is dee-

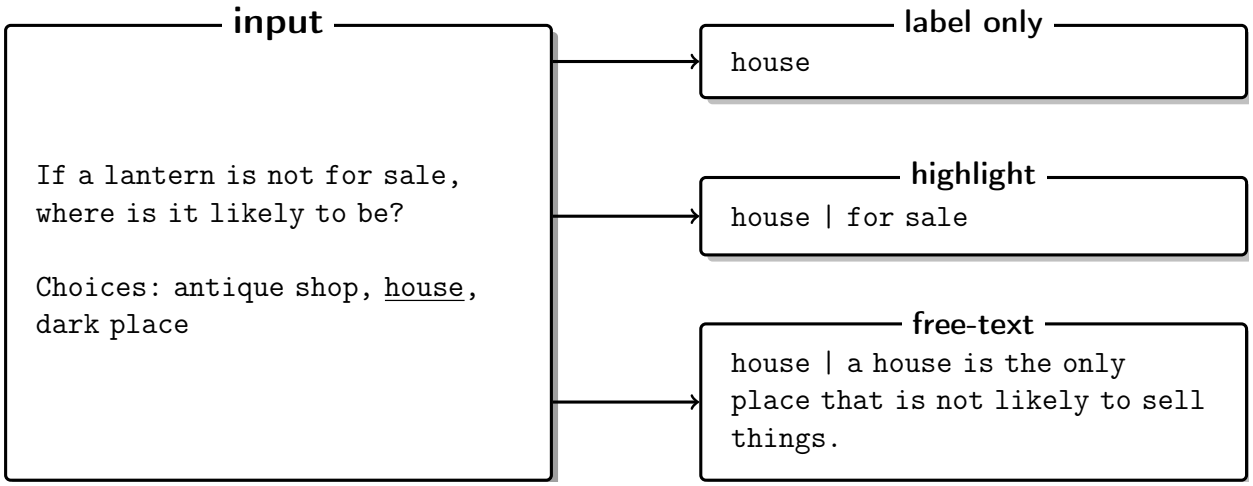


Figure 4.2 – Framing different types of rationale as sequence-to-sequence tasks. Highlight explanations extract sequences from the input. Free-text explanations have free form. Example from Cos-E v1.0, correct label is `house`.

med *introspective* (Sheh et Monteath, 2018) and can be faithful (Wiegreffe *et al.*, 2021). Early work revolved around attention supervision with human-generated highlights (see Bibal *et al.*, 2022). While sufficient highlights (DeYoung *et al.*, 2020) are favored in annotation guidelines rather than comprehensive ones (Wiegreffe et Marasović, 2021), it may be too sparse to understand model behaviour (Meister *et al.*, 2021). With the rise of generative LLM, self-rationalizing models now have the ability to output the label as well as an explanation in natural language (Rajani *et al.*, 2019; Narang *et al.*, 2020). Moreover, LLM such as GPT (Brown *et al.*, 2020) and T5 (Raffel *et al.*, 2020) benefit from their large pre-training corpora to distill world knowledge, which is helpful when explaining commonsense reasoning decisions.

4.2.3 Experimental Setup

The CommonsenseQA (Talmor *et al.*, 2019) dataset supports a multiple-choice question-answering task which seeks to evince the capacity of language models to leverage real-world knowledge to make inferences. The first version of the dataset, v1.0, comprises 7.5k examples, with three options per question, only one of which is correct. The v1.11 incorporates two additional so-called distractor choices and provides additional observations, for a total of

11k. The Commonsense Explanation (Cos-E) datasets (Rajani *et al.*, 2019) are augmented versions of these datasets introducing human-generated rationales, highlights and free-text alike. Although the authors had quality checks during their collection, free-text explanations of v1.11 tend to be of lesser quality (Narang *et al.*, 2020). Our experiments are carried out on both versions. All of our analyses were conducted on the validation set to access ground-truth annotation. Unanswered test sets are available on `https://github.com/salesforce/cos-e` (accessed on 1 August 2023) for v1.0 and `https://www.tau-nlp.sites.tau.ac.il/commonsenseqa` (accessed on 1 August 2023) for v1.11.

Similarly, the e-SNLI dataset (Camburu *et al.*, 2018) is an extension of the Stanford NLI dataset (Bowman *et al.*, 2015). It comprises 570k pairs of sentences, a premise and a hypothesis, together with a label categorizing their logical relationship. This label designates whether the hypothesis entails, contradicts or is neutral with respect to the premise. Thus, the task consists in parsing the sentence pair and placing it in one of these three classes. Free-text rationales were collected through crowdsourcing (Camburu *et al.*, 2018). Annotation guidelines encouraged annotators to provide self-contained rationales. Three separate rationales are provided for each observation in the validation and test sets, while training examples only have one. Furthermore, as a first step of the annotation process, before composing a free-text rationale, annotators were required to highlight the words of the input (both premise and hypothesis) that they deemed essential to categorizing their relationship. These selections are used as highlight rationales (DeYoung *et al.*, 2020).

4.2.3.1 Model

The text-to-text transfer Transformer (T5) model is an approach to multitask learning based on reframing multiple natural language tasks as text-to-text tasks (Raffel *et al.*, 2020). As a sequence-to-sequence model, it provides a unified framework for different rationale formats—label-only, label + highlights, and label + free-text—as the label and rationale can be presented as a single sequence, thus eliminating the need for adjustments in model architecture. This is illustrated in Figure 4.2. Such a framework can obtain strong performances (Narang *et al.*, 2020). For the present experiments, T5-base was fine-tuned on each

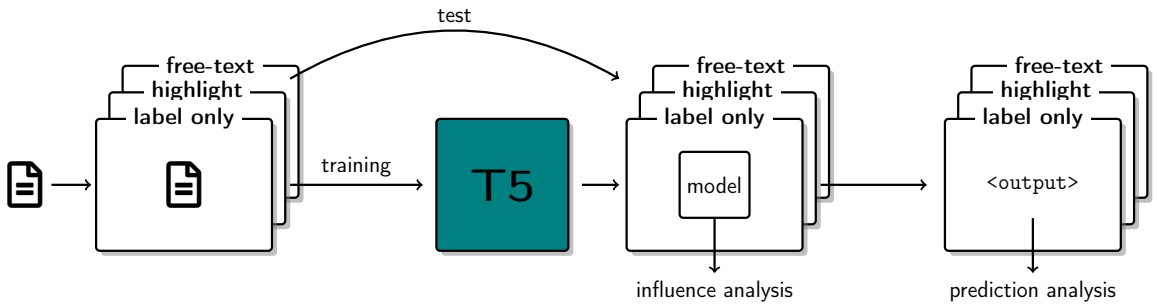


Figure 4.3 – Overview of the methodology. The data are adapted into different input–output pairs for each explanation type. A single sequence-to-sequence model (T5) serves as a common base for models producing the different explanations. Test-predicted labels and explanations are analyzed and compared. Models are probed to assess differences in training data influence and inner representation.

dataset and with each rationale configuration, resulting in nine different models. An overview of the methodology is presented in Figure 4.3. Hyperparameters are constant across these models (Wiegreffe *et al.*, 2021) : drop-out is set to 0.1, batch size to 64, and patience for early stopping to 10. Implementation, software and hardware details are provided in Appendix 4.2.6. At test time, output sequences are decoded greedily, with a length cut-off set to the 95th percentile of the length of target sequences in the training set (20 for Cos-E, 33 for e-SNLI). The quality assessment of the models is two-fold. First, classification power is measured with accuracy as answer choices are question-dependent. Second, output validity is assessed according to the following criteria : an answer is deemed valid if it is one of the available choices, a valid highlight is one that is a subsequence of the input, while free-text explanations do not follow any particular structure (Wiegreffe et Marasović, 2021).

4.2.3.2 Instance Attribution

The importance of training examples is estimated with IF (Koh et Liang, 2017; Rancourt *et al.*, 2022). Influence functions seek to measure the effect of particular training data on the resulting model. This is carried out by upweighting training examples and observing the change in loss value, similarly to jackknife resampling. A training example incurring a large loss variation will be deemed *influential*. Furthermore, the sign of said variation will indicate whether this example helps or hurts model predictions with regard to the loss. Given

a training set $\mathcal{D} = \{\mathbf{x}_k : 1 \leq k \leq n\}$, a machine learning algorithm will seek to minimize the loss, $\mathcal{L}$, a function of model parameters, $\theta : \arg\min_\theta \sum_k \mathcal{L}(\mathbf{x}_k, \theta)$. Upweighting a given observation, $\mathbf{x}_i \in \mathcal{D}$, by ϵ, this objective becomes : $\arg\min_\theta \sum_k \mathcal{L}(\mathbf{x}_k, \theta) + \epsilon\, \mathcal{L}(x_i, \theta)$. Let $\hat{\theta}$, $\hat{\theta}_{\epsilon,i}$ be the solutions to, respectively, the original and amended problems. One can compute the influence of $\mathbf{x}_i$ on the model prediction for some unseen test observation, $\mathbf{x}_t$ by :

$$\mathcal{I}(\mathbf{x}_t, \mathbf{x}_i) := \left.\frac{\mathrm{d}\,\mathcal{L}(\mathbf{x}_t, \hat{\theta}_{\epsilon,i})}{\mathrm{d}\epsilon}\right|_{\epsilon=0} = -\nabla_(\,\mathcal{L}\,\mathbf{x}_t, \hat{\theta})^\top H_{\hat{\theta}}^{-1} \nabla_(\,\mathcal{L}\,\mathbf{x}_i, \hat{\theta})$$

This equation allows the computation of influence without explicitly determining $\hat{\theta}_{\epsilon,i}$. Although, it only holds for loss functions twice differentiable and convex, which will seldom be the case in machine learning settings, it can serve as an adequate approximation for Transformer models (Han *et al.*, 2020; Guo *et al.*, 2021).

However, a difficulty in its application is its computational complexity. We rely on FastIF (Guo *et al.*, 2021) to speed up computation and consider the 1000 closest neighbors. We use the average of token embeddings as a sentence embedder (Ni *et al.*, 2022). Other hyperparameters follow recommendations (Guo *et al.*, 2021) : 1×10^3 iterations for the Hessian–vector product with a batch size of 1, the scale parameter is set to 10^4, and the damping parameter to, 5×10^{-3}. Given the large size of the e-SNLI test set ($\sim$10k data points), we perform instance attribution on a sample of 500 data points.

4.2.4 Results and Discussion

4.2.4.1 Task Performance

Task performance results for all datasets are presented in Table 4.1. For Cos-E, the accuracy of each model is far from human performance while remaining substantially better than random (v1.0 : 33%, v1.11 : 20%). For both versions of the dataset, the models can be ranked as per their accuracy in the following manner :

$$\text{label-only} > \text{label} + \text{free-text} > \text{label} + \text{highlight}.$$

Tableau 4.1 – Accuracy (%) of T5-base models on the Cos-E validation sets and e-SNLI test set. [§] indicates results from Wiegreffe *et al.* (2021) and [†] from Narang *et al.* (2020).

	Cos-E v1.0	Cos-E v1.11	e-SNLI
T5 label-only	69.4 (69.2[§])	60.9 (61.8[§])	91.1 (90.9[§])
T5 label + highlight	60.6	51.2	90.1
T5 label + free-text	65.1 (64.8[§])	56.8 (55.6[§], 59.4[†])	91.0 (90.8[§], 90.9[†])
human	95.3[‡]	88.9[‡]	-

This order corresponds to previous results from Rajani *et al.* (2019) (though they did not consider a T5 model) and Wiegreffe *et al.* (2021), maintaining a decrease in task performance for self-rationalizing models. These results also further suggest that highlights may be inappropriate explanations for commonsense reasoning tasks. For e-SNLI, however, these differences become very small, consistent with previous work (Wiegreffe *et al.*, 2021). Increased task performance on this dataset relative to Cos-E may be due to a variety of factors. Some simple, numerical considerations are the difference in dataset size (e-SNLI being ~ 57 times larger) and labels being a fixed rather than variable set. A consideration more complex and difficult to quantify is the degree to which so-called world knowledge embedded in the task may be required at test time or may be delivered in training.

4.2.4.2 Output Validity

Because T5 models frame their tasks as text-to-text tasks, the validity of the output—answer, highlight, and free-text explanation—must be verified. In particular, the models may not produce a valid label. In practice, this is seldom the case : regardless of the generated explanation (or lack thereof), all models are appropriate classifiers that seldom predict invalid answers ($< 0.5\%$).

Similarly, highlight explanations produced by the models must also be verified. Highlight explanations are subsequences of their input sequence. In training, this format is only enforced through supervision, *i.e.* the target explanations constitute valid highlights. However, the model is not otherwise constrained to adhere to this format. In particular, at prediction

time, there is no guarantee that the explanation sequence provided will be a valid highlight. Nonetheless, the proportion of invalid highlights observed is small across datasets : 1.9% for Cos-E v1.0, 1.4% for v1.11, and 0.1% for e-SNLI. It should be noted that validity is computed for our purposes by verifying independently whether each token is present in the input, ignoring character case. This does not invalidate overlapping highlights. Examining the relationship between prediction correctness and highlight validity reveals accuracy for examples resulting in invalid highlights to be fairly close to overall accuracy (v1.0 : 61.1%, v1.11 : 52.9%, e-SNLI : 100%). Further, manual inspection of these examples reveals a majority of these discrepancies to be attributable to differences in word inflection, *e.g.* "`spending all your time`" rather than "`spend all your time`". We contend the nature of highlights to be at odds with model pre-training and conjecture this conflict as the cause of the above : the models were pre-trained to produce not sequences of words but grammatically sound ones (or at least linguistically plausible ones). This would drive models to "correct" would-be highlights into proper phrases.

As for valid highlights, their overlap with ground-truth highlights is fairly low, with a mean Jaccard index of 36.9% for Cos-E v1.0, 42.8% for v1.11, and 60.8% for e-SNLI. Interestingly, the overlap increases drastically from incorrect predictions to correct ones for e-SNLI (Jaccard index : 30.4% to 64.1%) which is not the case for Cos-E. This may be due to Cos-E ground-truth highlights covering a larger portion of the input (v1.0 : 47.7%, v1.11 : 56.7%, e-SNLI : 19.8%). To boot, Cos-E highlights are more concentrated. We compute the ratio of the distance in words between the last and first highlighted words to the total number of highlighted words. A ratio of 1 would thus indicate that highlighted words form a contiguous subsequence. Cos-E highlights span 1.07 times their own length on average, compared with 2.95 for e-SNLI. These two observations—the greater length and higher concentration of highlights in Cos-E—may indicate summarily selected excerpts, which we contend to be less informative than sparser, noncontiguous highlights. This could be addressed in annotation by way of a highlighting budget.

In turn, free-text explanations are not required to adhere to a specific format. However, ge-

nerated explanations do appear to follow certain patterns. For Cos-E, the most prominent one is "`<answer> is the only <something> that [...]`" which is generated for 46.3% of examples in v1.0 although it appears in only 4.2% of the training data. The same pattern is noticeable for v1.11 with 10.8% generated explanations following the template even though only 0.7% of the training data follows it. This observation is surprising, as a very similar template "`<answer> is the only option that is correct|obvious`" was reannotated (Rajani *et al.*, 2019) and suggests that structured explanations may be more appropriate for this task than they are for RIT (Camburu *et al.*, 2018). Nevertheless, this is consistent with observed annotation practices : annotators tended to justify an answer with a contrastive explanation (Rajani *et al.*, 2019). Sadly, explanations following this template tend to be uninformative—even nonsensical— as shown in Table 4.2. In e-SNLI, prominent patterns are "`Not all <something> are <something else>`" and "`A <something> is [not] <something else>`". Selected examples are presented in Table 4.3. These patterns, akin to predicate quantification, are similarly overrepresented in generated explanations. Respectively, they make up 8.6% of generated explanations against 2.8% of training examples and 31.2% against 11.7%.

4.2.4.3 Probing Model Behaviour

Our experiments indicate that the embedding space is vastly different when the model is trained with certain types of explanations for different tasks. Indeed, analyzing the closest neighbors of test (or validation) examples reveals sharp drops in overlap for specific explanation types. In the case of Cos-E v1.0, the closest neighbors as per label-only and label + highlights models are fairly constant (average overlap of 86%). This proportion drops to roughly 20% when considering the model generating free-text explanations. In contrast, the average neighbor overlaps for v1.11 range from 63% to 77%. This suggests that free-text explanations add a lot of information to the inner representation of the model while highlights do not. Additionally, though the *label-only* and *label + highlights* models have a similar inner representation, their respective influence functions do not correlate, which indicates the presence of the Rashomon effect (Breiman, 2001) : the multiplicity of interpretations of the same

facts. Finally, our results invalidate the hypothesis that training examples with a free-text explanation following the aforementioned templates are highly influential. Indeed, observed correlations (Pearson and Spearman) neared 0 indicating the need for a deeper analysis of previous checkpoints.

As for e-SNLI, the model trained on highlight explanations appears to differ the most in its inner representation from the other models. Indeed, the proportion of shared neighbours drops from 7.5% between label-only and label + free-text to 0.7% when matching against highlight explanations. Such small overlaps, do not allow for significant estimation of influence correlations. Further influence analysis is required to study this issue.

4.2.5 Conclusion

This work has aimed to investigate the effect of training NLP models on rationales–textual explanations for individual predictions—in natural language inference and commonsense question-answering settings. This was carried out by training models to produce their prediction and the appropriate explanation as a single text sequence, using a common text-to-text pre-trained model.

The tasks under consideration require built-in world knowledge to complete adequately. Indeed, knowledge of real-world concepts and their relationships is embedded in the association between a prompt and its target response. As such, evaluating models on these tasks provides an assessment of the presence of this knowledge. It has been argued (Rajani *et al.*, 2019; Camburu *et al.*, 2018) that training a model on data with human-generated explanations may help the model acquire world knowledge more so than without said explanations due to the supplemental information that they provide. This is referred to as "bridging the knowledge gap". Our results indicate that the use of explanations hurts performance on the commonsense question-answering datasets, all the while being ineffectual in natural language inference.

It is difficult to evince how much of these performance discrepancies are attributable to

the nature of the tasks as opposed to the nature of the explanations present in the data. That is, it remains unclear to what extent commonsense question-answering is resistant or at odds with self-rationalization and to what extent the particular sets of explanations may be detrimental. While the annotation guidelines differ, what matters are arguably the material explanations that are present in the data. Their differences are difficult to quantify in a meaningful manner. Highlight explanations are more readily analyzed given their expected origin in the input and lack of added linguistic structure. We have noted that highlight explanations in Cos-E are longer and more concentrated compared to those in e-SNLI, which we contend to be less informative to the model. However, one limitation of our study is that this lack of linguistic structure in highlight explanations is contrary to model pre-training. In contrast, free-text explanations align with model pre-training, but their structure is more difficult to analyze. Further work in this direction is needed to characterize the difference in explanations between these datasets and to address the inherent skew in the comparison between highlight and free-text explanations.

Of course, the primary goal of self-rationalization is not to improve task acquisition but to increase explainability. This remains challenging as free-text explanations appear to follow certain noninformative or nonsensical formulae. Although some of these patterns were present in the training data, they are overrepresented in predictions. Further, our analyses did not show examples of them to be influential in the training process.

To improve our investigation of self-rationalizing models, considering a more complete annotation that includes negative and positive properties (Aggarwal *et al.*, 2021) could further bridge the knowledge gap and improve the quality of free-text explanations. To the best of our knowledge, this annotation scheme is the most thorough and self-contained, which could in turn greatly improve the quality of model-generated explanations.

4.2.5.1 Author Contributions

Conceptualization, F.R, D.M. ; methodology, F.R., D.M., M.-J.M. ; software, F.R., D.M. ; validation, formal analysis, investigation, F.R., P.V., D.M., M.-J.M. ; resources, M.-J.M. ;

writing—original draft preparation, F.R., P.V., D.M.; writing—review and editing, F.R., D.M., M.-J.M.; supervision, D.M., M.-J.M.; funding acquisition, D.M., M.-J.M. All authors have read and agreed to the published version of the manuscript.

4.2.5.2 Funding

This research was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) [M.-J. Meurs, NSERC Grant number 06487-2017] and the Government of Canada's New Frontiers in Research Fund (NFRF), [M.-J. Meurs, NFRFE-2018-00484].

4.2.5.3 Institutional Review Board Statement

Not applicable.

4.2.5.4 Informed Consent Statement

Not applicable.

4.2.5.5 Data Availability Statement

Cos-E test sets are available on `https://github.com/salesforce/cos-e` (accessed on 1 August 2023) for v1.0 and `https://www.tau-nlp.sites.tau.ac.il/commonsenseqa` (accessed on 1 August 2023) for v1.11. The e-SNLI dataset is available at `https://github.com/OanaMariaCamburu/e-SNLI` (accessed on 1 August 2023). Our source code is available under a GPLv3.0 license at `https://gitlab.labikb.ca/ikb-lab/nlp/self-rationalizing-commonsense-reasoning/self-rationalizing-models-for-commonsense-reasoning` (accessed on 1 August 2023).

4.2.5.6 Awknowledgements

This research was enabled in part by support provided by Calcul Québec (`https://www.calculquebec.ca` (accessed on 1 August 2023)) and The Digital Research Alliance of Canada

(`https://alliancecan.ca` (accessed on 1 August 2023)).

4.2.5.7 Conflict of Interests

The authors declare no conflict of interest.

4.2.5.8 Abreviations

The following abbreviations are used in this manuscript :

AI	Artificial Intelligence
Cos-E	Commonsense Explanation (dataset)
GPT	generative pre-trained Transformer
IF	Influence Functions
LLM	Large Language Model
NLP	Natural Language Processing
NLI	Natural Language Inference
T5	text-to-text transfer Transformer
XAI	explainable IA

4.2.6 Appendix : Computation Details

4.2.6.1 Implementation

The software for the experiments was developed for Python 3.8 on Linux (Ubuntu 20, 22) systems. Models were trained and tested with HuggingFace libraries (Transformers v2.9.1, Datasets v2.5.1) on PyTorch (v1.12) (Lhoest *et al.*, 2021; Wolf *et al.*, 2020; Paszke *et al.*, 2019). Influence function computation was largely based on the FastIF software put forth by Guo *et al.* (2021). We refer the reader to the code repository for a full list of dependencies (`https://gitlab.labikb.ca/ikb-lab/nlp/self-rationalizing-commonsense-reasoning/self-rationalizing-models-for-commonsense-reasoning` (accessed on 1 August

2023)).

4.2.6.2 Hardware and Runtimes

Experiments were run on single nodes in the Narval high-performance cluster (Calcul Québec et Digital Research Alliance of Canada, 2023) equipped with AMD Milan 7413 processors with Solid-State Drives (3.85TB) and NVidia A100 (40GB) GPUs. Training on a single GPU took approximately 1.5 hours per epoch for Cos-E and 2 hours per epoch on e-SNLI, for each model. Early stopping occurred before epoch 15 for all models. Prediction ran on a single GPU for less than 5 minutes per model for Cos-E and 1 hour for e-SNLI.

Tableau 4.2 – Selected model-generated free-text explanations following the template "`<answer> is the only <something> that [...]`" for Cos-E v1.0 and v1.11. Correct answer is in **bold**.

Question	**Answer choices**	**Model explanation**
	Cos-E v1.0	
A great teacher can be what when you are attending school ?	graduate, boredom, **inspiration**	inspiration is the only thing that can be done when you are attending school.
According to what book did an apple tree lead to the downfall of man ?	woods, spain, **bible**	bible is the only book that is a book
After he got hired he hoped for success at his what ?	michigan, **new job**, vocation	new job is the only option that is not a vocation.
Aliens, humans and everything is a part of what ?	outer space, **universe**, roswell	the universe is the only place that is a part of the universe.
	Cos-E v1.11	
He has lactose intolerant, but was eating dinner made of cheese, what followed for him ?	digestive, feel better, sleepiness, **indigestion**, illness	indigestion is the only answer that is not intolerant
If a person were going to bed, what would be their goal ?	lazy, insomnia, rest, **falling asleep**, dreaming of	falling asleep is the only option that would help you fall asleep.
If you are hungry and going fishing, why would you be going fishing ?	to see the fish, have fun, **catching fish**, wet clothes, killing	catching fish is the only option that would make you have fun.
If you're remembering something, it's because of your what of it to begin with ?	knowledge, **knowing**, forgetful, pleasure, depression	knowing is the only thing that is remembering

Tableau 4.3 – Selected model-generated free-text explanations following the templates "`Not all <something> are <something else>`" or "`A <something> is [not] <something else>`" for e-SNLI.

Premise	**Hypothesis**	**Label / Prediction**	**Model explanation**
A couple walk hand in hand down a street.	The couple is married.	neutral neutral	Not all couples are married.
Two women are observing something together.	Two women are looking at a flower together.	neutral neutral	Not all things are a flower.
Two people using a water buffalo to cultivate a watery field.	A couple is farming with their pet buffalo.	entailment neutral	Not all water buffalos are pet buffalos.
A man playing an electric guitar on stage.	A man playing banjo on the floor.	contradiction contradiction	A banjo is not an electric guitar.
A little girl with pointy pigtails is sitting in the sand on the beach.	The girl is sitting on the beach.	entailment entailment	A little girl is a girl.
A speaker is talking with a TV in the background.	There is a live bear in the background.	neutral contradiction	A speaker is not a bear.

CONCLUSION

Les récents développements en TALN ont facilité l'adoption de ces technologies dans de nombreux domaines d'application. Ceci est néanmoins accompagné de risques et une étude de ces derniers est nécessaire afin de les atténuer. En plus des stratégies courantes, la gestion efficace des risques liés à l'IA s'appuient sur les critères permettant qu'un système soit digne de confiance. Notamment, l'utilisation de méthodes améliorant l'interprétabilité d'un système à divers phases de son cycle de vie est essentielle (Tabassi, 2023). En effet, cette pratique permet d'identifier des faiblesses du sytème (Bhatt *et al.*, 2020b; Ribeiro *et al.*, 2020) et favorise la supervision adéquate par un humain (Buçinca *et al.*, 2021; Vasconcelos *et al.*, 2023).

Les travaux présentés dans ce mémoire ont abordé diverses applications des FI visant à améliorer l'interprétabilité de modèles en TALN. Plus précisément, les FI sont mises à profit afin d'étudier le comportement du modèle (Koh et Liang, 2017). On analyse premièrement l'association de l'influence et d'une variable latente sur la prédiction. Enfin, on utilise les FI pour analyser des artéfacts d'un modèle autojustifiant.

L'analyse des FI présentée au chapitre 3 a permis de mieux comprendre la valeur de la qualité de l'annotation. Celles-ci considèrent un domaine d'application où la quantité de données disponibles est très restreinte, la détection du risque de suicide. Une piste de solution pour atténuer cet obstacle est d'assouplir les exigences quant à l'expertise en santé mentale des personnes annotatrices (Shing *et al.*, 2018). Nos expériences comparent l'influence des annotations impliquant la production participative d'un large public non-spécialiste à celle des annotations expertes. Nos résultats suggèrent que les annotations produites par des personnes non-expertes sont très utiles pour le modèle. De plus, les FI ont permis d'identifier des observations globalement influentes, c'est-à-dire qu'elles figurent parmi les exemples les plus influents (bénéfiques ou néfastes) pour la majorité des prédictions. Une importante quantité d'observations globalement bénéfiques ont été annotées par des non-expert.es, suggérant ainsi leur valeur pour la classification. Il est aussi possible que ces exemples globalement

bénéfiques soient des observations aberrantes (Barshan *et al.*, 2020), mettant en doute la précédente conclusion. Ainsi, des analyses supplémentaires sont nécessaires pour pour élucider cette question. L'utilisation de techniques d'augmentation de données pourrait être considérée lors de ces expériences.

Ensuite, nos expériences avec un modèle autojustifiant ont permis d'identifier des limites importantes de cette approche au chapitre 4. Premièrement, les explications par surlignage générées par les modèles T5 (Raffel *et al.*, 2020) ont tendance à ne pas être des sous-ensembles du contenu textuel en entrée. Ceci contredit les précédentes observations de Narang *et al.* (2020). De plus, ce type d'explications a grandement heurté les performances des modèles pour la tâche de raisonnement basé sur le sens commun. En effet, la justesse subit des baisses avoisinant les 10 % pour Cos-E (Rajani *et al.*, 2019) alors qu'elle reste stable dans le cas de e-SNLI (Camburu *et al.*, 2018). Il est néanmoins difficile d'expliquer ces écarts comme plusieurs causes sont possibles : la nature de la tâche de même que les différentes directives d'annotation sont des explications autant probables les unes que les autres. Enfin, malgré des efforts soutenus lors de l'annotation pour que les explications ouvertes ne suivent pas de gabarit (Camburu *et al.*, 2018), celles générées par nos modèles ont toute de même tendance à le faire. Ce comportement est d'autant plus étonnant que les gabarits identifiés lors de l'analyse manuelle des sorties du modèle révèle que ceux-ci sont rares dans les données d'entraînement. Ceci nous a mené à poser l'hypothèse suivante : les exemples d'entraînement dont l'explication humaine suit le gabarit sont très influents. Malheureusement, les influences calculées n'attestent d'aucune corrélation en ce sens, un comportement qui nous apparaît contre-intuitif. Ainsi, on peut se poser la question suivante :« s'agit-il d'un cas où les FI sont fragiles ou la cause est autre ? » En effet, plusieurs conditions sont nécessaires à la convergence de l'estimation des FI. Tel que présenté au chapitre 2, vérifier ces conditions pour un modèle simple, la régression linéaire, n'est pas trivial. De plus, la fragilité des FI en apprentissage automatique est bien documentée (Basu *et al.*, 2020; Zhang et Zhang, 2022) . Les travaux de Bae *et al.* (2022) analysant différentes limites des méthodes d'estimation actuellement disponibles peuvent fournir des pistes d'amélioration pour des travaux futurs.

En plus des enjeux théoriques des FI soulevés au chapitre 2, un obstacle de taille à l'adoption de cette technique subsiste : la lourdeur des calculs. Ainsi, il est difficile d'analyser les FI à grande échelle pour les modèles complexes couramment utilisés en TALN. Bien que plusieurs travaux (Yeh *et al.*, 2018; Guo *et al.*, 2021; Sui *et al.*, 2021) se sont attaqués à cette difficulté, le jour où l'on pourra utiliser les FI pour analyser un modèle de langue de grande taille avec des outils tels que AllenNLP Interpret (Wallace *et al.*, 2019) ou Azimuth (Gauthier-Melançon *et al.*, 2022) est encore loin. Ceci est d'autant plus important puisqu'un nombre grandissant d'organisations seront amenées à conduire des analyses plus poussées sur le fonctionnement de leurs systèmes fondés sur l'IA afin de le rendre accessible au grand public. En effet, la Loi sur l'IA de l'Union européenne (UE) [3] de même que le *Artificial Intelligence Risk Management Framework* proposé par le NIST (Tabassi, 2023) ont des exigences élevées en terme d'analyse des systèmes fondés sur l'IA et de leur interprétabilité, particulièrement lorsque les risques qu'ils posent sont élevés. Ainsi, on espère que les travaux présentés dans ce mémoire encourageront la recherche de nouvelles approches améliorant l'efficacité et la robustesse des FI.

3. Au moment de la rédaction de ce mémoire, la version finale de la loi n'existe pas. Le trilogue ainsi que l'approbation sont préalables à l'obtention du texte final (Madiega, 2023).

APPENDICE A
DÉMONSTRATIONS

A.1 Théorème 1

Démonstration. Considérons le problème d'optimisation

$$\widehat{\beta} = \arg\min_{\beta} \|\widehat{\epsilon}\|^2 = \arg\min_{\beta} \|y - X\beta\|^2.$$

Posons $f(\beta) = \|y - X\beta\|^2$. On trouve que le gradient est

$$\nabla_\beta f = 2X^T(y - X\beta) = 2X^T X\beta - 2X^t y$$

et $\beta^* = \left(X^T X\right)^{-1} X^T y$ est un point critique. Comme la hessienne

$$H(h) = \nabla^2_\beta f = X^T X$$

est inversible, on trouve que H est définie positive et β^* est minimal. D'où, $\widehat{\beta} = \beta^*$ est l'estimateur des MCO. □

A.2 Théorème 3

Démonstration. En vertu de (2.9),

$$\begin{aligned}\left(\mathbf{\Sigma}((1-\varepsilon)F + \varepsilon\,\delta_{(\mathrm{x,y})})\right)^{-1} &= \left((1-\varepsilon)\left(\mathbf{\Sigma}(F) + \frac{\varepsilon}{1-\varepsilon}\,\mathbf{x}\,\mathbf{x}^T\right)\right)^{-1} \\ &= \frac{1}{1-\varepsilon}\left(\mathbf{\Sigma}(F) + \frac{\varepsilon}{1-\varepsilon}\,\mathbf{x}\,\mathbf{x}^T\right)^{-1}.\end{aligned}$$

On pose $A = \mathbf{\Sigma}(F)$, $a = \varepsilon\,\mathbf{x}^T$ et $b = \mathbf{x}^T$ et on applique (2.11) :

$$\left(\mathbf{\Sigma}((1-\varepsilon)F + \varepsilon\,\delta_{(\mathrm{x,y})})\right)^{-1} = \frac{1}{1-\varepsilon}\,\mathbf{\Sigma}^{\text{-}1}(F)\Bigg[I_p - \frac{\varepsilon}{1-\varepsilon}\left(1 + \frac{\varepsilon}{1-\varepsilon}\,\mathbf{x}^T\,\mathbf{\Sigma}^{\text{-}1}\,\mathbf{x}\right)^{-1}\mathbf{x}\,\mathbf{x}^T\,\mathbf{\Sigma}^{\text{-}1}(F)\Bigg]$$

Ainsi, en vertu de (2.10) et (2.8), on a

$$\begin{aligned} T((1-\varepsilon)F + \varepsilon\,\delta_{(\mathrm{x,y})}) &= (1-\varepsilon)\left(\mathbf{\Sigma}((1-\varepsilon)F + \varepsilon\,\delta_{(\mathrm{x,y})})\right)^{-1}\left(\gamma(F) + \frac{\varepsilon}{(1-\varepsilon)}y\,\mathbf{x}\right) \\ &= T(F) + \frac{\varepsilon}{1-\varepsilon}\,\mathbf{\Sigma}^{\text{-}1}(F)\,\mathbf{x}\Bigg[y - \frac{1}{1+\frac{\varepsilon}{1-\varepsilon}\,\mathbf{x}^T\,\mathbf{\Sigma}^{\text{-}1}\,\mathbf{x}}\left[\mathbf{x}^T\,T(F) + \frac{\varepsilon}{1-\varepsilon}y\,\mathbf{x}^T\,\mathbf{\Sigma}^{\text{-}1}\,\mathbf{x}\right]\Bigg]. \end{aligned}$$

Puisque $\frac{\frac{\varepsilon}{1-\varepsilon}\,\mathbf{x}^T\,\mathbf{\Sigma}^{\text{-}1}\,\mathbf{x}}{1+\frac{\varepsilon}{1-\varepsilon}\,\mathbf{x}^T\,\mathbf{\Sigma}^{\text{-}1}\,\mathbf{x}} = \left(1 - \frac{1}{1+\frac{\varepsilon}{1-\varepsilon}\,\mathbf{x}^T\,\mathbf{\Sigma}^{\text{-}1}\,\mathbf{x}}\right)$, on trouve

$$\begin{aligned} T((1-\varepsilon)F + \varepsilon\,\delta_{(\mathrm{x,y})}) &= T(F) + \frac{\varepsilon}{1-\varepsilon}\,\mathbf{\Sigma}^{\text{-}1}(F)\,\mathbf{x}\Bigg[y - \frac{1}{1+\frac{\varepsilon}{1-\varepsilon}\,\mathbf{x}^T\,\mathbf{\Sigma}^{\text{-}1}\,\mathbf{x}}\,\mathbf{x}^T\,T(F) + y\left(1 - \frac{1}{1+\frac{\varepsilon}{1-\varepsilon}\,\mathbf{x}^T\,\mathbf{\Sigma}^{\text{-}1}\,\mathbf{x}}\right)\Bigg] \\ &= T(F) + \varepsilon\,\mathbf{\Sigma}^{\text{-}1}(F)\,\mathbf{x}\Bigg[y - \frac{1}{1-\varepsilon+\varepsilon\,\mathbf{x}^T\,\mathbf{\Sigma}^{\text{-}1}\,\mathbf{x}}\,\mathbf{x}^T\,T(F) + y\left(\frac{1}{1-\varepsilon} - \frac{1}{1-\varepsilon+\varepsilon\,\mathbf{x}^T\,\mathbf{\Sigma}^{\text{-}1}\,\mathbf{x}}\right)\Bigg]. \end{aligned}$$

D'où,

$$\begin{aligned} \mathcal{IC}_{F,T}(\mathbf{x}, y) &= \lim_{\varepsilon\to 0}\mathbf{\Sigma}^{\text{-}1}(F)\,\mathbf{x}\Bigg[y - \frac{1}{1-\varepsilon+\varepsilon\,\mathbf{x}^T\,\mathbf{\Sigma}^{\text{-}1}\,\mathbf{x}}\,\mathbf{x}^T\,T(F) + y\left(\frac{1}{1-\varepsilon} - \frac{1}{1-\varepsilon+\varepsilon\,\mathbf{x}^T\,\mathbf{\Sigma}^{\text{-}1}\,\mathbf{x}}\right)\Bigg] \\ &= \mathbf{\Sigma}^{\text{-}1}(F)\,\mathbf{x}\left(y - \mathbf{x}^T\,T(F)\right). \end{aligned}$$

□

A.3 Proposition 1

Démonstration. Puisque T est linéaire, on trouve

$$\mathcal{IC}_{F,T}(z) = T(\delta_z) - T(F)$$
$$\widehat{\mathcal{IC}}_{F,T}(z) = T(\delta_z) - T(\widehat{F}).$$

Comme $T(\widehat{F})$ est consistant par hypothèse, on trouve le résultat. □

A.4 Théorème 6

Cette démonstration reprend celle de Koh et Liang (2017) qui est disponible dans les ressources supplémentaires.

Démonstration. Puisque $\hat{\theta}_{\epsilon,z}$ est un point critique de $\widehat{\mathcal{L}}(\theta) + \epsilon\,\mathcal{L}(\theta, z)$, on a

$$-\frac{1}{n}\sum_{i=1}^{n} \nabla_\theta^2\,\mathcal{L}(\theta, z_i)\big|_{\theta=\hat{\theta}_{\epsilon,z}} = \epsilon\,\nabla_\theta\,\mathcal{L}(\theta, z)|_{\theta=\hat{\theta}_{\epsilon,z}}\,.$$

En dérivant par rapport à ϵ de chaque côté, on trouve

$$-\hat{H}_{\hat{\theta}}\frac{\mathrm{d}\hat{\theta}_{\epsilon,z}}{\mathrm{d}\epsilon} = \nabla_\theta\,\mathcal{L}(\theta, z)|_{\theta=\hat{\theta}_{\epsilon,z}} + \epsilon\,\nabla_\theta^2\,\mathcal{L}(\theta, z)\big|_{\theta=\hat{\theta}_{\epsilon,z}}\frac{\mathrm{d}\hat{\theta}_{\epsilon,z}}{\mathrm{d}\epsilon}$$

$$\iff \quad \frac{\mathrm{d}\hat{\theta}_{\epsilon,z}}{\mathrm{d}\epsilon} = -\hat{H}_{\hat{\theta}}^{-1}\,\nabla_\theta\,\mathcal{L}(\theta, z)|_{\theta=\hat{\theta}_{\epsilon,z}} - \epsilon\hat{H}_{\hat{\theta}}^{-1}\,\nabla_\theta^2\,\mathcal{L}(\theta, z)\big|_{\theta=\hat{\theta}_{\epsilon,z}}\frac{\mathrm{d}\hat{\theta}_{\epsilon,z}}{\mathrm{d}\epsilon}. \tag{A.1}$$

On obtient le résultat en évaluant (A.1) en $\epsilon = 0$. □

GLOSSAIRE

affiné Traduction de *finetuned*.

apprentissage automatique Traduction de *machine learning*.

arrêt prématuré de l'entraînement Traduction de *early stopping*.

attribut Traduction de *feature*.

carte de saillance Traduction de *saliency map*.

comportement du modèle Traduction de *model behavior*.

courbe de l'influence Traduction de *influence curve*.

courbe de l'influence empirique Traduction de *empirical influence curve*..

désapprentissage automatique Traduction de *machine unlearning*.

empoisonnement des données Traduction de *data poisonning*.

estimateur plug-in Traduction de *plug-in estimator*.

état caché Traduction de *hidden state*.

explicabilité Traduction de *explainability*.

impliquant la production participative d'un large public non-spécialiste Traduction de *crowdsourced*.

matrice chapeau Matrice des projections liant la variable dépendante à la prédiction.

modèle autojustifiant Traduction de *self-rationalizing model*.

modèle de langue de grande taille Traduction de *large language model*.

partitionnement de données Traduction de *clustering*.

point à fort effet de levier Traduction de *leverage point*.

raisonnement basé sur le sens commun Traduction de *commonsense reasoning*.

reconnaissance d'implications textuelles Traduction de *natural language inference*.

traitement automatique du language naturel Traduction de *natural language processing*.

tête d'attention Traduction de *attention head*.

RÉFÉRENCES

Agarwal, N., Bullins, B. et Hazan, E. (2017). Second-order stochastic optimization for machine learning in linear time. *Journal of Machine Learning Research*, *18*(116), 1–40.

Aggarwal, S., Mandowara, D., Agrawal, V., Khandelwal, D., Singla, P. et Garg, D. (2021). Explanations for CommonsenseQA : New Dataset and Models. Dans *ACL-IJCNLP*, volume 1, 3050–3065.

Alaa, A. et van der Schaar, M. (2020). Discriminative Jackknife : Quantifying Uncertainty in Deep Learning via Higher-Order Influence Functions. Dans *Proceedings of the 37th International Conference on Machine Learning*, 165–174. PMLR.

Alfrink, K., Keller, I., Kortuem, G. et Doorn, N. (2022). Contestable ai by design : Towards a framework. *Minds and Machines*, 1–27.

Alon-Barkat, S. et Busuioc, M. (2023). Human–ai interactions in public sector decision making :"automation bias" and "selective adherence" to algorithmic advice. *Journal of Public Administration Research and Theory*, *33*(1), 153–169.

Angwin, J., Larson, J., Mattu, S. et Kirchner, L. (2016). *Machine Bias*. Rapport technique, ProPublica

Arnold, M., Bellamy, R. K. E., Hind, M., Houde, S., Mehta, S., Mojsilovic, A., Nair, R., Ramamurthy, K. N., Reimer, D., Olteanu, A., Piorkowski, D., Tsay, J. et Varshney, K. R. (2019). FactSheets : Increasing Trust in AI Services through Supplier's Declarations of Conformity. arXiv :1808.07261 [cs].

Arnoldi, W. E. (1951). The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of applied mathematics*, *9*(1), 17–29.

Bae, J., Ng, N. H., Lo, A., Ghassemi, M. et Grosse, R. B. (2022). If influence functions are the answer, then what is the question ? Dans *Advances in Neural Information Processing Systems*.

Bahdanau, D., Cho, K. et Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. Dans *3rd International Conference on Learning Representations, ICLR 2015*. Récupéré de `http://arxiv.org/abs/1409.0473`

Bao, Y., Chang, S., Yu, M. et Barzilay, R. (2018). Deriving machine attention from human rationales. Dans *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 1903–1913.

Barshan, E., Brunet, M.-E. et Dziugaite, G. K. (2020). Relatif : Identifying explanatory training samples via relative influence. Dans *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 de *Proceedings of Machine Learning Research*, 1899–1909. PMLR.

Bastings, J. et Filippova, K. (2020). The elephant in the interpretability room : Why use attention as explanation when we have saliency methods ? Dans *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, 149–155., Online. Association for Computational Linguistics. http://dx.doi.org/10.18653/v1/2020.blackboxnlp-1.14

Basu, S., Pope, P. et Feizi, S. (2020). Influence functions in deep learning are fragile. Dans *International Conference on Learning Representations*.

Belsley, D. A., Kuh, E. et Welsch, R. E. (2005). *Regression diagnostics : Identifying influential data and sources of collinearity*. John Wiley & Sons.

Bhatt, U., Andrus, M., Weller, A. et Xiang, A. (2020a). Machine learning explainability for external stakeholders. *arXiv preprint arXiv :2007.05408*.

Bhatt, U., Xiang, A., Sharma, S., Weller, A., Taly, A., Jia, Y., Ghosh, J., Puri, R., Moura, J. M. et Eckersley, P. (2020b). Explainable machine learning in deployment. Dans *Conference on Fairness, Accountability, and Transparency*.

Bibal, A., Cardon, R., Alfter, D., Wilkens, R., Wang, X., François, T. et Watrin, P. (2022). Is attention explanation ? an introduction to the debate. Dans *ACL*, volume 1, 3889–3900.

Bilodeau, B., Jaques, N., Koh, P. W. et Kim, B. (2022). Impossibility theorems for feature attribution. *arXiv preprint arXiv :2212.11870*.

Bowman, S. R., Angeli, G., Potts, C. et Manning, C. D. (2015). A large annotated corpus for learning natural language inference. Dans *EMNLP*.

Branson, T. P. (1993). *The functional determinant*. Citeseer.

Breheny, P. (2012). Statistical functionals and influence functions. https://myweb.uiowa.edu/pbreheny/uk/teaching/621/notes/8-28.pdf.

Breiman, L. (2001). Statistical modeling : The two cultures. *Statistical science*, *16*(3), 199–231.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., 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., 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. et Amodei, D. (2020). Language models are few-shot learners. Dans *Advances in Neural Information Processing Systems*, volume 33, 1877–1901.

Buçinca, Z., Malaya, M. B. et Gajos, K. Z. (2021). To trust or to think : Cognitive forcing functions can reduce overreliance on ai in ai-assisted decision-making. *Proceedings of the ACM on Human-Computer Interaction*, *5*(CSCW1), 1–21.

Calcul Québec et Digital Research Alliance of Canada (2023). Narval, a 5.9 petaflops supercomputer for scientific researchers in canada. https://docs.alliancecan.ca/wiki/Narval/en.

Camburu, O.-M., Rocktäschel, T., Lukasiewicz, T. et Blunsom, P. (2018). e-snli : Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems 31* 9539–9549.

Cauchy, A. (1847). Méthode générale pour la résolution des systemès d'équations simultanées. *Comp. Rend. Sci. Paris*, *25*(1847), 536–538.

Chancellor, S. et De Choudhury, M. (2020). Methods in predictive techniques for mental health status on social media : a critical review. *npj Digital Medicine*.

Chancellor, S., Mitra, T. et De Choudhury, M. (2016). Recovery amid pro-anorexia : Analysis of recovery in social media. Dans *Conference on Human Factors in Computing Systems*.

Char, D. S., Shah, N. H. et Magnus, D. (2018). Implementing machine learning in health care–addressing ethical challenges. *The New England journal of medicine*, *378*(11), 981.

Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P. et Robinson, T. (2013). One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv :1312.3005*.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. et Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. Dans *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734., Doha, Qatar. Association for Computational Linguistics. http://dx.doi.org/10.3115/v1/D14-1179

Choi, J. H., Hickman, K. E., Monahan, A. et Schwarcz, D. (2023). ChatGPT Goes to Law School. *Journal of Legal Education*. http://dx.doi.org/10.2139/ssrn.4335905

Commission européenne (2016). Règlement (ue) 2016/679 du parlement européen et du conseil du 27 avril 2016 relatif à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données, et abrogeant la directive 95/46/ce (règlement général sur la protection des données) (texte présentant de l'intérêt pour l'eee). OJ L 119, 4.5.2016, p. 1–88. Récupéré de https://eur-lex.europa.eu/legal-content/FR/TXT/?uri=CELEX:32016R0679

Commission européenne (2021). Proposition de règlement du parlement européen et du conseil établissant des règles harmonisées concernant l'intelligence artificielle (législation sur l'intelligence artificielle) et modifiant certains actes législatifs de l'union. COM/2021/206 final. Récupéré de https://eur-lex.europa.eu/legal-content/FR/TXT/?uri=CELEX:52021PC0206

Cook, R. D. et Weisberg, S. (1982). *Residuals and Influence in Regression.*

Coppersmith, G., Dredze, M. et Harman, C. (2014). Quantifying mental health signals in Twitter. Dans *Workshop on computational linguistics and clinical psychology : From linguistic signal to clinical reality.*

Dasgupta, I., Lampinen, A. K., Chan, S. C., Creswell, A., Kumaran, D., McClelland, J. L. et Hill, F. (2022). Language models show human-like content effects on reasoning. *arXiv preprint arXiv :2207.07051.*

De Angelis, L., Baglivo, F., Arzilli, G., Privitera, G. P., Ferragina, P., Tozzi, A. E. et Rizzo, C. (2023). Chatgpt and the rise of large language models : the new ai-driven infodemic threat in public health. *Frontiers in Public Health*, *11*, 1166120.

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

DeYoung, J., Jain, S., Rajani, N. F., Lehman, E., Xiong, C., Socher, R. et Wallace, B. C. (2020). Eraser : A benchmark to evaluate rationalized nlp models. Dans *ACL*, 4443–4458.

Dhanorkar, S., Wolf, C. T., Qian, K., Xu, A., Popa, L. et Li, Y. (2021). Who needs to know what, when ? : Broadening the Explainable AI (XAI) Design Space by Looking at Explanations Across the AI Lifecycle. Dans *Designing Interactive Systems Conference 2021*, DIS '21, 1591–1602., New York, NY, USA. Association for Computing Machinery. `http://dx.doi.org/10.1145/3461778.3462131`

Doshi-Velez, F. et Kim, B. (2017). Towards A Rigorous Science of Interpretable Machine Learning. *arXiv :1702.08608 [cs, stat].* arXiv : 1702.08608.

Dzindolet, M. T., Peterson, S. A., Pomranky, R. A., Pierce, L. G. et Beck, H. P. (2003). The role of trust in automation reliance. *International Journal of Human-Computer Studies*, *58*(6), 697–718. Trust and Technology.

Ehsan, U., Liao, Q. V., Muller, M., Riedl, M. O. et Weisz, J. D. (2021). Expanding Explainability : Towards Social Transparency in AI systems. Dans *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–19., New York, NY, USA. Association for Computing Machinery. `http://dx.doi.org/10.1145/3411764.3445188`

Erasmus, A. et Brunet, T. D. (2022). Interpretability and unification. *Philosophy & Technology*, *35*(2), 1–6.

Erasmus, A., Brunet, T. D. et Fisher, E. (2021). What is interpretability ? *Philosophy & Technology*, *34*(4), 833–862.

Ernala, S. K., Birnbaum, M. L., Candan, K. A., Rizvi, A. F., Sterling, W. A., Kane, J. M. et De Choudhury, M. (2019). Methodological gaps in predicting mental health states from social media : Triangulating diagnostic signals. Dans *CHI Conference on Human Factors in Computing Systems*.

Fahrmeir, L., Kneib, T., Lang, S. S. M. et Marx, B. D. (2021). *Regression : models, methods and applications* (second edition éd.). Berlin, Germany : Springer. http://dx.doi.org/10.1007/978-3-662-63882-8

Fernholz, L. T. (1983). *von Mises Calculus for Statistical Functionals*, volume 19 de *Lecture Notes in Statistics*. New York : Springer-Verlag.

Fix, E. et Hodges Jr, J. L. (1952). *Discriminatory analysis-nonparametric discrimination : Small sample performance*. Rapport technique, California Univ Berkeley.

Frieder, S., Pinchetti, L., Griffiths, R.-R., Salvatori, T., Lukasiewicz, T., Petersen, P. C., Chevalier, A. et Berner, J. (2023). Mathematical capabilities of chatgpt. *arXiv preprint arXiv :2301.13867*.

Galton, F. (1886). Regression towards mediocrity in hereditary stature. *The Journal of the Anthropological Institute of Great Britain and Ireland*, *15*, 246–263.

Gâteaux, R. (1922). Sur diverses questions de calcul fonctionnel. *Bulletin de la Société Mathématique de France*, *50*, 1–37.

Gauthier-Melançon, G., Marquez Ayala, O., Brin, L., Tyler, C., Branchaud-charron, F., Marinier, J., Grande, K. et Le, D. (2022). Azimuth : Systematic error analysis for text classification. Dans *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing : System Demonstrations*, 298–310. Association for Computational Linguistics.

Ghorbani, A., Abid, A. et Zou, J. (2019). Interpretation of Neural Networks Is Fragile. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(01), 3681–3688. http://dx.doi.org/10.1609/aaai.v33i01.33013681

Goodfellow, I., Bengio, Y. et Courville, A. (2016). *Deep Learning*. MIT Press.

Goodfellow, I. J., Shlens, J. et Szegedy, C. (2015). Explaining and harnessing adversarial examples. Dans *3rd International Conference on Learning Representations*. Récupéré de http://arxiv.org/abs/1412.6572

Graves, A. et Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm networks. Dans *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 4, 2047–2052. http://dx.doi.org/10.1109/IJCNN.2005.1556215

Guo, H., Rajani, N., Hase, P., Bansal, M. et Xiong, C. (2021). Fastif : Scalable influence functions for efficient model interpretation and debugging. Dans *EMNLP*, 10333–10350.

Hampel, F. R. (1974). The influence curve and its role in robust estimation. *Journal of the american statistical association*, *69*(346), 383–393.

Han, X., Wallace, B. C. et Tsvetkov, Y. (2020). Explaining black box predictions and unveiling data artifacts through influence functions. Dans *Annual Meeting of the Association for Computational Linguistics*.

Héder, M. (2023). Explainable AI : A brief history of the concept. *ERCIM NEWS*, (134), 9–10.

Hinkley, D. V. (1977). Jackknifing in unbalanced situations. *Technometrics*, *19*(3), 285–292.

Hoaglin, D. C. et Welsch, R. E. (1978). The Hat Matrix in Regression and ANOVA. *The American Statistician*, *32*(1), 17–22. Publisher : Taylor & Francis, http://dx.doi.org/10.1080/00031305.1978.10479237

Hochreiter, S. et Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hoffman, R. R., Klein, G. et Mueller, S. T. (2018). Explaining Explanation For "Explainable AI". *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *62*(1), 197–201.

Hoffman, S. (2021). The emerging hazard of ai-related health care discrimination. *Hastings Center Report*, *51*(1), 8–9. http://dx.doi.org/https://doi.org/10.1002/hast.1203

Howard, J. et Ruder, S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv :1801.06146*.

Huber, P. J. (1981). Robust statistics. *Wiley Series in Probability and Mathematical Statistics*.

Hulsen, T. (2023). Explainable artificial intelligence (xai) : Concepts and challenges in healthcare. *AI*, *4*(3), 652–666. http://dx.doi.org/10.3390/ai4030034

Isaak, J. et Hanna, M. J. (2018). User data privacy : Facebook, cambridge analytica, and privacy protection. *Computer*, *51*(8), 56–59. http://dx.doi.org/10.1109/MC.2018.3191268

Jacovi, A., Bastings, J., Gehrmann, S., Goldberg, Y. et Filippova, K. (2023). Diagnosing AI Explanation Methods with Folk Concepts of Behavior. *arXiv*. arXiv :2201.11239 [cs].

Jacovi, A. et Goldberg, Y. (2020). Towards faithfully interpretable NLP systems : How should we define and evaluate faithfulness ? Dans *ACL*, 4198–4205.

Jacovi, A. et Goldberg, Y. (2021). Aligning faithful interpretations with their social attribution. *Transactions of the Association for Computational Linguistics*, *9*, 294–310.

Jain, S. et Wallace, B. C. (2019). Attention is not Explanation. Dans *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, 3543–3556., Minneapolis, Minnesota. Association for Computational Linguistics. http://dx.doi.org/10.18653/v1/N19-1357

Jain, S., Wiegreffe, S., Pinter, Y. et Wallace, B. C. (2020). Learning to faithfully rationalize by construction. Dans *58th Annual Meeting of the Association for Computational Linguistics, ACL*, 4459–4473. http://dx.doi.org/10.18653/v1/2020.acl-main.409

James, S. L., Abate, D., Abate, K. H., Abay, S. M., Abbafati, C., Abbasi, N., Abbastabar, H., Abd-Allah, F., Abdela, J., Abdelalim, A. *et al.* (2018). Global, Regional, and National Incidence, Prevalence, and Years Lived with Disability for 354 Diseases and Injuries for 195 Countries and Territories, 1990–2017 : A Systematic Analysis for the Global Burden of Disease Study 2017. *The Lancet*.

Jiao, W., Wang, W., Huang, J.-t., Wang, X. et Tu, Z. (2023). Is chatgpt a good translator ? a preliminary study. *arXiv preprint arXiv :2301.08745*.

Johnson, J., Douze, M. et Jégou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, *7*(3), 535–547.

Khot, T., Clark, P., Guerquin, M., Jansen, P. et Sabharwal, A. (2020). Qasc : A dataset for question answering via sentence composition. Dans *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 8082–8090.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F. et Sayres, R. (2018). Interpretability Beyond Feature Attribution : Quantitative Testing with Concept Activation Vectors (TCAV). Dans *International Conference on Machine Learning*, 2668–2677. PMLR.

Kingma, D. P. et Ba, J. (2014). Adam : A method for stochastic optimization. *arXiv :1412.6980*.

Kleinberg, J., Ludwig, J., Mullainathan, S. et Rambachan, A. (2018). Algorithmic Fairness. *AEA Papers and Proceedings*, *108*, 22–27. http://dx.doi.org/10.1257/pandp.20181018

Kocielnik, R., Amershi, S. et Bennett, P. N. (2019). Will you accept an imperfect ai ? exploring designs for adjusting end-user expectations of ai systems. Dans *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–14.

Koh, P. W. et Liang, P. (2017). Understanding Black-box Predictions via Influence Functions. Dans *International Conference on Machine Learning*.

Koh, P. W. W., Ang, K.-S., Teo, H. et Liang, P. S. (2019). On the accuracy of influence functions for measuring group effects. *Advances in neural information processing systems*, *32*.

Krishnan, M. (2020). Against Interpretability : a Critical Examination of the Interpretability Problem in Machine Learning. *Philosophy & Technology*, *33*(3), 487–502. http://dx.doi.org/10.1007/s13347-019-00372-9

Kutlu, M., McDonnell, T., Elsayed, T. et Lease, M. (2020). Annotator rationales for labeling tasks in crowdsourcing. *Journal of Artificial Intelligence Research*, *69*, 143–189.

Lamm, M., Palomaki, J., Alberti, C., Andor, D., Choi, E., Soares, L. B. et Collins, M. (2021). Qed : A framework and dataset for explanations in question answering. *Transactions of the Association for Computational Linguistics*, *9*, 790–806.

Lee, D., Park, H., Pham, T. et Yoo, C. D. (2020). Learning augmentation network via influence functions. Dans *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10961–10970.

Lhoest, Q., Villanova del Moral, A., Jernite, Y., Thakur, A., von Platen, P., Patil, S., Chaumond, J., Drame, M., Plu, J., Tunstall, L., Davison, J., Šaško, M., Chhablani, G., Malik, B., Brandeis, S., Le Scao, T., Sanh, V., Xu, C., Patry, N., McMillan-Major, A., Schmid, P., Gugger, S., Delangue, C., Matussière, T., Debut, L., Bekman, S., Cistac, P., Goehringer, T., Mustar, V., Lagunas, F., Rush, A. et Wolf, T. (2021). Datasets : A community library for natural language processing. Dans *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing : System Demonstrations*, 175–184., Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. http://dx.doi.org/10.18653/v1/2021.emnlp-demo.21

Lipton, Z. C. (2018). The mythos of model interpretability : In machine learning, the concept of interpretability is both important and slippery. *ACM Queue*.

Liu, E. Z., Haghgoo, B., Chen, A. S., Raghunathan, A., Koh, P. W., Sagawa, S., Liang, P. et Finn, C. (2021). Just Train Twice : Improving Group Robustness without Training Group Information. Dans *International Conference on Machine Learning*, 6781–6792. PMLR.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. et Stoyanov, V. (2019). RoBERTa : A Robustly Optimized BERT Pretraining Approach. *arXiv :1907.11692*.

Losada, D. E., Crestani, F. et Parapar, J. (2020). eRisk 2020 : Self-harm and depression challenges. Dans *European Conference on Information Retrieval*, 557–563. Springer.

Lundberg, S. M. et Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*.

Lyons, H., Velloso, E. et Miller, T. (2021). Conceptualising contestability : Perspectives on contesting algorithmic decisions. *Proceedings of the ACM on Human-Computer Interaction*, *5*(CSCW1), 1–25.

Lyons, J. B., Clark, M. A., Wagner, A. R. et Schuelke, M. J. (2017). Certifiable trust in autonomous systems : Making the intractable tangible. *AI Magazine*, *38*(3), 37–49.

Lyu, Q., Apidianaki, M. et Callison-Burch, C. (2022). Towards faithful model explanation in nlp : A survey. *arXiv preprint arXiv :2209.11326*.

Madiega, T. (2023). Artificial intelligence act. `https://www.europarl.europa.eu/RegData/etudes/BRIE/2021/698792/EPRS_BRI(2021)698792_EN.pdf`.

Manning, C. et Schutze, H. (1999). *Foundations of statistical natural language processing*. MIT press.

Marcinkowski, F., Kieslich, K., Starke, C. et Lünich, M. (2020). Implications of ai (un-)fairness in higher education admissions : The effects of perceived ai (un-)fairness on exit, voice and organizational reputation. Dans *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, p. 122–130., New York, NY, USA. Association for Computing Machinery. `http://dx.doi.org/10.1145/3351095.3372867`

Mathew, B., Saha, P., Yimam, S. M., Biemann, C., Goyal, P. et Mukherjee, A. (2021). Hatexplain : A benchmark dataset for explainable hate speech detection. Dans *AAAI*, volume 35, 14867–14875.

Maupomé, D., Armstrong, M. D., Rancourt, F. et Meurs, M.-J. (2021). Leveraging textual similarity to predict beck depression inventory answers. Dans *Canadian Conference on AI*.

McDonnell, T., Lease, M., Kutlu, M. et Elsayed, T. (2016). Why is that relevant ? collecting annotator rationales for relevance judgments. Dans *AAAI-CHCC*, volume 4, 139–148.

McGorry, P. D. et Mei, C. (2018). Early intervention in youth mental health : progress and future directions. *Evidence-Based Mental Health*, *21*(4), 182–184. Publisher : Royal College of Psychiatrists Section : Clinical review, `http://dx.doi.org/10.1136/ebmental-2018-300060`

Meister, C., Lazov, S., Augenstein, I. et Cotterell, R. (2021). Is sparse attention more interpretable ? Dans *ACL-ICNLP*, volume 2, 122–129.

Meng, Y., Fan, C., Sun, Z., Hovy, E., Wu, F. et Li, J. (2020). Pair the dots : Jointly examining training history and test stimuli for model interpretability. *arXiv preprint arXiv :2010.06943*.

Mercado, J. E., Rupp, M. A., Chen, J. Y., Barnes, M. J., Barber, D. et Procci, K. (2016). Intelligent agent transparency in human–agent teaming for multi-uxv management. *Human factors*, *58*(3), 401–415.

Mikolov, T., Chen, K., Corrado, G. et Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781*.

Miller, T. (2019). Explanation in artificial intelligence : Insights from the social sciences. *Artificial intelligence*, *267*, 1–38.

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D. et Gebru, T. (2019). Model Cards for Model Reporting. Dans *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, 220–229., New York, NY, USA. Association for Computing Machinery. http://dx.doi.org/10.1145/3287560.3287596

Mohammadi, E., Amini, H. et Kosseim, L. (2019). CLaC at CLPsych 2019 : Fusion of neural features and predicted class probabilities for suicide risk assessment based on online posts. Dans *Workshop on Computational Linguistics and Clinical Psychology*.

Molnar, C. (2022). *Interpretable Machine Learning : A Guide for Making Black Box Models Explainable* (2 éd.). Récupéré de christophm.github.io/interpretable-ml-book/

Morris, J., Lifland, E., Lanchantin, J., Ji, Y. et Qi, Y. (2020). Reevaluating Adversarial Examples in Natural Language. Dans *Findings of the Association for Computational Linguistics : EMNLP 2020*, 3829–3839., Online. Association for Computational Linguistics. http://dx.doi.org/10.18653/v1/2020.findings-emnlp.341

Mujtaba, D. F. et Mahapatra, N. R. (2019). Ethical considerations in ai-based recruitment. Dans *2019 IEEE International Symposium on Technology and Society (ISTAS)*, 1–7. http://dx.doi.org/10.1109/ISTAS48451.2019.8937920

Narang, S., Raffel, C., Lee, K., Roberts, A., Fiedel, N. et Malkan, K. (2020). Wt5 ?! training text-to-text models to explain their predictions. *arXiv preprint arXiv :2004.14546*.

Ni, J., Hernandez Abrego, G., Constant, N., Ma, J., Hall, K., Cer, D. et Yang, Y. (2022). Sentence-t5 : Scalable sentence encoders from pre-trained text-to-text models. Dans *Findings of the Association for Computational Linguistics : ACL 2022*, 1864–1874.

Nor, A. K. M., Pedapati, S. R., Muhammad, M. et Leiva, V. (2022). Abnormality detection and failure prediction using explainable bayesian deep learning : Methodology and case study with industrial data. *Mathematics*, *10*(4). http://dx.doi.org/10.3390/math10040554

Organisation de coopération et de développement économiques (2019a). Présentation des principes sur l'ia. Consulté le 31 août 2023. Récupéré de https://oecd.ai/fr/ai-principles

Organisation de coopération et de développement économiques (2019b). Transparence et l'explicabilité (principe 1.3). Consulté le 31 août 2023. Récupéré de https://oecd.ai/fr/dashboards/ai-principles/P7

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A. *et al.* (2022). Training language models to follow instructions with human feedback. *arXiv preprint arXiv :2203.02155*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. et Chintala, S. (2019). PyTorch : An imperative style, high-performance deep learning library. Dans *Advances in Neural Information Processing Systems 32*, 8024–8035. Curran Associates, Inc.

Pennington, J., Socher, R. et Manning, C. D. (2014). Glove : Global vectors for word representation. Dans *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. et Zettlemoyer, L. (2018). Deep contextualized word representations. Dans *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237., New Orleans, Louisiana. Association for Computational Linguistics. http://dx.doi.org/10.18653/v1/N18-1202

Pezeshkpour, P., Jain, S., Wallace, B. et Singh, S. (2021). An empirical comparison of instance attribution methods for NLP. Dans *NAACL-HLT*, 967–975.

Prasetya, Y. (2022). Anns and unifying explanations : Reply to erasmus, brunet, and fisher. *Philosophy & Technology*, *35*(2), 1–9.

Quenouille, M. H. (1956). Notes on bias in estimation. *Biometrika*, *43*(3/4), 353–360.

Radford, A., Narasimhan, K., Salimans, T. et Sutskever, I. (2018). Improving language understanding by generative pre-training.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P. J. *et al.* (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, *21*(140), 1–67.

Rajani, N. F., McCann, B., Xiong, C. et Socher, R. (2019). Explain yourself! leveraging language models for commonsense reasoning. Dans *ACL2019*.

Rancourt, F., Maupomé, D. et Meurs, M.-J. (2022). On the Influence of Annotation Quality in Suicidal Risk Assessment from Text. *Proceedings of the 35th Canadian Conference on Artificial Intelligence*. http://dx.doi.org/10.21428/594757db.36acee93

Rancourt, F. et Maupomé, D. (2022). On the influence of annotation quality in suicidal risk assessment from text. https://gitlab.labikb.ca/ikb-lab/nlp/canadian-ai-2022/on-the-influence-of-

annotation-quality-in-suicidal-risk-assessment-from-text/on-the-infl
uence-of-annotation-quality-in-suicidal-risk-assessment-from-text.

Rancourt, F. et Maupomé, D. (2023). Investigating self rationalizing models for commonsense reasoning.
https://gitlab.labikb.ca/ikb-lab/nlp/self-rationalizing-commonsense-re
asoning/self-rationalizing-models-for-commonsense-reasoning.

Rancourt, F., Vondrlik, P., Maupomé, D. et Meurs, M.-J. (2023). Investigating self-rationalizing models for commonsense reasoning. *Stats*, *6*(3), 907–919.
http://dx.doi.org/10.3390/stats6030056

Ras, G., Xie, N., van Gerven, M. et Doran, D. (2021). Explainable Deep Learning : A Field Guide for the Uninitiated. *arXiv :2004.14545 [cs, stat]*. arXiv : 2004.14545.

Rasmussen, P. (2021). 2021q3 reports : Office manager. Publié le 19 juillet 2021. Récupéré de https://www.aclweb.org/adminwiki/index.php/2021Q3_Reports:
_Office_Manager

Ren, J.-J. et Sen, P. K. (1995). Hadamard differentiability on d [0, 1] p. *Journal of Multivariate Analysis*, *55*(1), 14–28.

Ribeiro, M. T., Singh, S. et Guestrin, C. (2016). "why should i trust you ?" explaining the predictions of any classifier. Dans *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Ribeiro, M. T., Wu, T., Guestrin, C. et Singh, S. (2020). Beyond Accuracy : Behavioral Testing of NLP Models with CheckList. Dans *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4902–4912., Online. Association for Computational Linguistics.
http://dx.doi.org/10.18653/v1/2020.acl-main.442

Rosenblatt, F. (1958). The perceptron : A probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.

Rosenfeld, R. (2000). Two decades of statistical language modeling : Where do we go from here ? *Proceedings of the IEEE*, *88*(8), 1270–1278.

Ross, A., Marasović, A. et Peters, M. (2021). Explaining NLP models via minimal contrastive editing (MiCE). Dans *Findings of the Association for Computational Linguistics : ACL-IJCNLP 2021*, 3840–3852., Online. Association for Computational Linguistics. http://dx.doi.org/10.18653/v1/2021.findings-acl.336

Rozario, S. et Čevora, G. (2023). Explainable ai does not provide the explanations end-users are asking for. *arXiv preprint arXiv :2302.11577*.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, *1*(5), 206–215. http://dx.doi.org/10.1038/s42256-019-0048-x

Saxena, S. (2018). Excess mortality among people with mental disorders : a public health priority. *The Lancet Public Health*.

Schioppa, A., Filippova, K., Titov, I. et Zablotskaia, P. (2023). Theoretical and practical perspectives on what influence functions do. *arXiv preprint arXiv :2305.16971*.

Schioppa, A., Zablotskaia, P., Vilar, D. et Sokolov, A. (2022). Scaling up influence functions. Dans *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 8179–8186.

Schölkopf, B., Herbrich, R. et Smola, A. J. (2001). A generalized representer theorem. Dans *Computational Learning Theory : 14th Annual Conference on Computational Learning Theory, COLT 2001 and 5th European Conference on Computational Learning Theory, EuroCOLT 2001 Amsterdam, The Netherlands, July 16–19, 2001 Proceedings 14*, 416–426. Springer.

Schotanus-Dijkstra, M., Drossaert, C. H. C., Pieterse, M. E., Boon, B., Walburg, J. A. et Bohlmeijer, E. T. (2017). An early intervention to promote well-being and flourishing and reduce anxiety and depression : A randomized controlled trial. *Internet Interventions*, *9*, 15–24. `http://dx.doi.org/10.1016/j.invent.2017.04.002`

Schulam, P. et Saria, S. (2019). Can you trust this prediction ? auditing pointwise reliability after learning. Dans *International Conference on Artificial Intelligence and Statistics*.

Selbst, A. D. et Powles, J. (2017). Meaningful information and the right to explanation. *International Data Privacy Law*, *7*(4), 233–242. `http://dx.doi.org/10.1093/idpl/ipx022`

Shapiro, A. (1991). Asymptotic analysis of stochastic programs. *Annals of Operations Research*, *30*, 169–186.

Sheh, R. et Monteath, I. (2018). Defining Explainable AI for Requirements Analysis. *KI - Künstliche Intelligenz*, *32*(4), 261–266.

Shing, H.-C., Nair, S., Zirikly, A., Friedenberg, M., Daumé III, H. et Resnik, P. (2018). Expert, Crowdsourced, and Machine Assessment of Suicide Risk via Online Postings. Dans *Workshop on Computational Linguistics and Clinical Psychology*.

Shing, H.-C., Resnik, P. et Oard, D. W. (2020). A Prioritization Model for Suicidality Risk Assessment. Dans *Annual Meeting of the Association for Computational Linguistics*.

Simonyan, K., Vedaldi, A. et Zisserman, A. (2014). Deep inside convolutional networks : Visualising image classification models and saliency maps. Dans *Workshop at International Conference on Learning Representations*.

Slack, D., Hilgard, S., Jia, E., Singh, S. et Lakkaraju, H. (2020). Fooling LIME and SHAP : Adversarial Attacks on Post hoc Explanation Methods. Dans *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES '20, 180–186., New York, NY, USA. Association for Computing Machinery. http://dx.doi.org/10.1145/3375627.3375830

Statistics (2023). Glossary : Consistent estimator. Consulté le 31 août 2023. Récupéré de https://www.statistics.com/glossary/consistent-estimator/

Stigler, S. M. (1981). Gauss and the invention of least squares. *The Annals of Statistics*, *9*(3), 465–474.

Strout, J., Zhang, Y. et Mooney, R. (2019). Do Human Rationales Improve Machine Explanations ? Dans *ACL Workshop BlackboxNLP*, 56–62.

Sui, Y., Wu, G. et Sanner, S. (2021). Representer point selection via local jacobian expansion for post-hoc classifier explanation of deep neural networks and ensemble models. *Advances in neural information processing systems*, *34*, 23347–23358.

Sun, X., Yang, D., Li, X., Zhang, T., Meng, Y., Qiu, H., Wang, G., Hovy, E. et Li, J. (2021). Interpreting Deep Learning Models in Natural Language Processing : A Review. *arXiv :2110.10470 [cs]*. arXiv : 2110.10470.

Synced (2019). Acl 2019 reports record-high paper submissions ; begins notifying accepted authors. Publié le 17 mai 2019. Récupéré de https://medium.com/syncedreview/acl-2019-reports-record-high-paper-submissions-begins-notifying-accepted-authors-bbfb13adf405

Tabassi, E. (2023). Artificial intelligence risk management framework (ai rmf 1.0). https://www.nist.gov/itl/ai-risk-management-framework.

Talmor, A., Herzig, J., Lourie, N. et Berant, J. (2019). Commonsenseqa : A question answering challenge targeting commonsense knowledge. Dans *NAACL-HLT*, volume 1, 4149–4158.

Teso, S., Bontempelli, A., Giunchiglia, F. et Passerini, A. (2021). Interactive label cleaning with example-based explanations. Dans *Advances in Neural Information Processing Systems*.

Ting, D. et Brochu, E. (2018). Optimal subsampling with influence functions. *Advances in neural information processing systems*, *31*.

Tunstall, L., von Werra, L. et Wolf, T. (2022). *Natural language processing with transformers*. O'Reilly Media, Inc.

Vasconcelos, H., Jörke, M., Grunde-McLaughlin, M., Gerstenberg, T., Bernstein, M. S. et Krishna, R. (2023). Explanations can reduce overreliance on ai systems during decision-making. *Proceedings of the ACM on Human-Computer Interaction*, *7*(CSCW1), 1–38.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. et Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Vijayakumar, A. K., Cogswell, M., Selvaraju, R. R., Sun, Q., Lee, S., Crandall, D. J. et Batra, D. (2020). Diverse Beam Search : Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv :1610.02424*.

von Mises, R. (1947). On the asymptotic distribution of differentiable statistical functions. *The annals of mathematical statistics*, *18*(3), 309–348.

Wachter, S., Mittelstadt, B. et Floridi, L. (2017). Why a Right to Explanation of Automated Decision-Making Does Not Exist in the General Data Protection Regulation. *International Data Privacy Law*, *7*(2), 76 – 99. http://dx.doi.org/10.1093/idpl/ipx005

Wachter, S., Mittelstadt, B. et Russell, C. (2018). Counterfactual Explanations Without Opening the Black Box : Automated Decisions and the GDPR. *Harvard Journal of Law & Technology*, *31*(2). http://dx.doi.org/10.2139/ssrn.3063289

Wallace, E., Tuyls, J., Wang, J., Subramanian, S., Gardner, M. et Singh, S. (2019). AllenNLP Interpret : A Framework for Explaining Predictions of NLP Models. Dans *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) : System Demonstrations*, 7–12., Hong Kong, China. Association for Computational Linguistics. http://dx.doi.org/10.18653/v1/D19-3002

Wang, J.-L., Chiou, J.-M. et Müller, H.-G. (2016). Functional data analysis. *Annual Review of Statistics and its application*, *3*, 257–295.

White House Office of Science and Technology Policy (2022). Blueprint for an AI Bill of Rights. Publié en octobre 2022. Récupéré de https://www.whitehouse.gov/wp-content/uploads/2022/10/Blueprint-for-an-AI-Bill-of-Rights.pdf

whuber (2018). Taylor expansion for random variables. Cross Validated. Consulté le 31 août 2023. Récupéré de https://stats.stackexchange.com/q/378841

Wiegreffe, S. et Marasovic, A. (2021). Teach me to explain : A review of datasets for explainable natural language processing. Dans *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*. Récupéré de https://openreview.net/forum?id=ogNcxJn32BZ

Wiegreffe, S. et Marasović, A. (2021). Teach me to explain : A review of datasets for explainable nlp. Dans *NeurIPS*.

Wiegreffe, S., Marasović, A. et Smith, N. A. (2021). Measuring association between labels and free-text rationales. Dans *EMNLP*, 10266–10284.

Wiegreffe, S. et Pinter, Y. (2019). Attention is not not Explanation. Dans *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 11–20., Hong Kong, China. Association for Computational Linguistics. http://dx.doi.org/10.18653/v1/D19-1002

Wilson, C., Ghosh, A., Jiang, S., Mislove, A., Baker, L., Szary, J., Trindel, K. et Polli, F. (2021). Building and Auditing Fair Algorithms : A Case Study in Candidate Screening. Dans *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, 666–677., Virtual Event Canada. ACM. http://dx.doi.org/10.1145/3442188.3445928

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q. et Rush, A. M. (2020). Transformers : State-of-the-art natural language processing. Dans *Conference on Empirical Methods in Natural Language Processing : System Demonstrations*, 38–45. http://dx.doi.org/10.18653/v1/2020.emnlp-demos.6

Wu, G., Hashemi, M. et Srinivasa, C. (2022). Puma : Performance unchanged model augmentation for training data removal. Dans *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI-2022)*, Vancouver, Canada.

Yang, Y., Malaviya, C., Fernandez, J., Swayamdipta, S., Le Bras, R., Wang, J.-P., Bhagavatula, C., Choi, Y. et Downey, D. (2020). Generative Data Augmentation for Commonsense Reasoning. Dans *Findings of the Association for Computational Linguistics : EMNLP 2020*, 1008–1025., Online. Association for Computational Linguistics. http://dx.doi.org/10.18653/v1/2020.findings-emnlp.90

Yeh, C.-K., Kim, J., Yen, I. E.-H. et Ravikumar, P. K. (2018). Representer Point Selection for Explaining Deep Neural Networks. *Advances in neural information processing systems*, *31*.

Yin, M., Wortman Vaughan, J. et Wallach, H. (2019). Understanding the Effect of Accuracy on Trust in Machine Learning Models. Dans *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–12., Glasgow Scotland Uk. ACM. http://dx.doi.org/10.1145/3290605.3300509

Zaidan, O. F., Eisner, J. et Piatko, C. D. (2007). Using "annotator rationales" to improve machine learning for text categorization. Dans *NAACL-HLT*, 260–267.

Zhang, R. et Zhang, S. (2022). Rethinking influence functions of neural networks in the over-parameterized regime. Dans *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 9082–9090.

Zhang, W., Huang, Z., Zhu, Y., Ye, G., Cui, X. et Zhang, F. (2021). On Sample Based Explanation Methods for NLP : Faithfulness, Efficiency and Semantic Evaluation. Dans *Proceedings of the 59th Annual Meeting of the Association for Computational*

Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers), 5399–5411., Online. Association for Computational Linguistics. http://dx.doi.org/10.18653/v1/2021.acl-long.419

Zhang, X., Wu, B., Zhang, X., Zhou, Q., Hu, Y. et Liu, J. (2022). A novel assessable data augmentation method for mechanical fault diagnosis under noisy labels. *Measurement*, *198*, 111114.

Zhang, Y., Marshall, I. et Wallace, B. C. (2016). Rationale-augmented convolutional neural networks for text classification. Dans *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 795–804., Austin, Texas. Association for Computational Linguistics. http://dx.doi.org/10.18653/v1/D16-1076

Zhang, Z., Singh, J., Gadiraju, U. et Anand, A. (2019). Dissonance between human and machine understanding. *Proceedings of the ACM on Human-Computer Interaction*, *3*(CSCW), 1–23.

Zirikly, A., Resnik, P., Uzuner, Ö. et Hollingshead, K. (2019). CLPsych 2019 Shared Task : Predicting the Degree of Suicide Risk in Reddit Posts. Dans *Workshop on Computational Linguistics and Clinical Psychology*.