# EBBE-Text : Visualisation de la frontière de décision des réseaux de neurones en classification automatique de textes

Alexis Delaforge* Jérôme Azé* Arnaud Sallaberry*,**
Maximilien Servajean*,** Sandra Bringay*,** Caroline Mollevi***,****

*LIRMM, Université de Montpellier, CNRS
CC477 - 161 rue Ada, 4095 Montpellier Cedex 5, France
prenom.nom@lirmm.fr
http://www.lirmm.fr/
**Groupe AMIS, Université Paul-Valéry Montpellier 3
Route de Mende, 34199 Montpellier Cedex 5, France
***Institut du Cancer Montpellier (ICM)
208 Avenue des Apothicaires, Parc Euromédecine, 34298 Montpellier Cedex 5, France
caroline.mollevi@icm.unicancer.fr,
https://www.icm.unicancer.fr/fr
****Institut Desbrest d'Epidémiologie et de Santé Publique,
UMR Inserm - Université de Montpellier, Montpellier, France

**Résumé.** En classification automatique de textes, de nombreux travaux récents portent sur l'interprétation des réseaux de neurones par la production d'explications des prédictions. L'approche originale présentée dans cet article consiste à visualiser la frontière de décision et le positionnement des données vis-à-vis de celle-ci offrant ainsi une nouvelle approche de l'explication. Notre méthode calcule tout d'abord un espace de représentation des phrases, puis en exploite la structure linéaire afin de visualiser la répartition et le regroupement des données autour de la frontière de décision. Le principal apport de notre méthode est le processus de visualisation de la frontière de décision, permettant d'explorer la distance à la frontière de décision (et donc la certitude d'un réseau en ses prédictions) mais aussi les chemins menant à celle-ci ou encore la proximité entre phrases.

## 1 Introduction

Récemment, les réseaux de neurones ont connu du succès dans les tâches de Traitement Automatique du Langage (TAL) (Young et al., 2018) comme la traduction (Cho et al., 2014; Bahdanau et al., 2015), la reconnaissance d'entités nommées (Collobert et al., 2011) ou encore l'analyse de sentiments (Socher et al., 2013). L'utilisation des techniques d'apprentissage profond soulève des questions sur l'interprétabilité, l'explicabilité, la confiance et la transparence de ces réseaux (Lipton, 2018). Il est, d'ailleurs, primordial de s'intéresser à ceux-ci, d'autant plus que le parlement européen (Goodman et Flaxman, 2016) a fixé des règles parmi les plus strictes au monde concernant l'interprétabilité de ces réseaux de neurones.

D'après Lipton (2018), nous notons deux concepts qui constituent la définition de l'interprétabilité : la **transparence** et les **explications post-hoc**. La **transparence** se définit comme la facilité par laquelle un humain peut comprendre et reproduire le fonctionnement d'un modèle indépendamment d'une prédiction. La transparence d'un modèle peut se diviser en trois parties : une relevant de la compréhension globale du fonctionnement du modèle, une traitant d'une compréhension des différentes parties du modèle, enfin une dernière traitant de la compréhension des mécanismes d'apprentissage et donc de leur convergence vers une solution optimale. En apprentissage profond, la décision d'un réseau de neurones passe par l'activation ou non de milliers de neurones. Il est donc impossible pour un humain de tout analyser. Des informations supplémentaires à la simple prédiction sont donc nécessaires et nous amènent à la construction d'explications post-hoc.

L'explication pour une prédiction donnée (ou **l'explication post-hoc**) est faite lorsque l'on se sert de différents indicateurs issus ou non du fonctionnement d'un modèle pour expliquer le choix qui a été fait. L'explicabilité d'une prédiction peut servir à rendre plus interprétable les modèles lorsque qu'une méthode post-hoc est construite de manière à identifier ce qui, dans le fonctionnement du modèle, a le plus influencé la décision. Si une explication associée à une prédiction sert légèrement à l'interprétabilité d'un réseau, ce n'est que la multiplication des explications qui peut réellement donner des intuitions aux utilisateurs quant au fonctionnement du modèle employé. Lipton (2018) classe les explications post-hoc en quatre catégories :

1. Les explications verbales ou écrites des prédictions qui justifient celles-ci.
2. Les techniques de visualisation qui permettent d'explorer l'espace de représentation des données ou d'afficher des indications sur ce qui, dans les données d'entrée, a participé à la prédiction.
3. Les explications locales qui peuvent donner accès à des explications plus simples concernant seulement une sous-partie de l'espace des données (Ribeiro et al., 2016).
4. Les explications qui présentent les comportements pour des exemples similaires.

Les explications données aux prédictions doivent être d'une complexité modérée. Des techniques en visualisation permettent de visualiser l'intégralité de la structure des réseaux, les interpréter, les expliquer, les déboguer (Hohman et al., 2019). Dans ce contexte, nous proposons une visualisation de **la frontière de décision** d'un réseau de neurones dans le cas de la classification dichotomique. Cette visualisation de la frontière de décision et de la distance des données à celle-ci permet une meilleure identification des données correctement classées ou non et de la certitude avec laquelle le réseau les a classées. Dans nos travaux, les explications se font chacune localement. Nous identifions donc différentes localités dans l'espace de représentation et à l'échelle de la localité nous explorons nos données plus finement. L'exploration des données contenues dans ces localités permet donc d'identifier le voisinage des données. Nous nous plaçons donc, dans les catégories d'explication post-hoc suivantes : visualisation, explication locale et exemples similaires. Ce type d'approche manque actuellement aux méthodes d'explication disponibles en classification automatique de textes dans lesquelles la distance à la frontière de décision et la frontière de décision elle-même n'est pas visualisée.

Dans cet article, nous présentons les travaux menés en interprétabilité en section 2 puis nous développons nos travaux en section 3 [1] avant de proposer un cas d'étude sur des données réelles en section 4. Enfin, nous concluons et proposons des perspectives en section 5.

1. http://advanse.lirmm.fr/template_container.php?template=AD/EBBE.php

## 2 Travaux existants

En classification de textes, différentes techniques d'explication des prédictions sont utilisables pour mettre en lumière les mots ayant le plus participé à la prédiction. Ces techniques s'appuient notamment sur le gradient (Springenberg et al., 2015; Selvaraju et al., 2020; Smilkov et al., 2017; Dimopoulos et al., 1995), sur les portes ("gates unit") que l'on retrouve notamment dans les réseaux récurrents (Karpathy et al., 2015) ou sur la suppression de certaines dimensions des représentations apprises par les réseaux (Li et al., 2016). Le point commun de ces techniques est qu'elles identifient les mots ayant le plus participé à la prédiction, le plus souvent à l'aide de cartes de chaleur. Les mécanismes d'attention (Bahdanau et al., 2015; Vaswani et al., 2017; Raganato et Tiedemann, 2018; Zenkel et al., 2019; Xu et al., 2018) présents dès la construction des réseaux de neurones peuvent, selon leur rôle, expliquer les prédictions. L'attention met en évidence les mots utiles à la prédiction comme le font les méthodes basées sur le gradient avec les mêmes mécanismes de visualisation (cartes de chaleur). Lors de l'utilisation d'espaces de grande dimension construits par l'encodage de mots ou de phrases (Mikolov et al., 2013a; Pennington et al., 2014; Devlin et al., 2019), il est possible d'utiliser les techniques de visualisation de données et de réduction de dimensions pour explorer ces espaces (Mikolov et al., 2013b; Smilkov et al., 2016; Xu et al., 2018). Ces techniques, PCA (Pearson, 1901) / t-SNE (Hinton et Roweis, 2002; van der Maaten et Hinton, 2008) / UMAP (McInnes et Healy, 2018), deviennent essentielles lorsque l'on cherche à trouver des similitudes ou des proximités entre les données. Elles peuvent servir d'explications post-hoc des exemples similaires. Elles participent donc à l'interprétabilité.

Il ressort de ces recherches récentes qu'il est compliqué d'interpréter un réseau de neurones autrement qu'en s'intéressant à des exemples précis. Même s'il existe des techniques pour explorer l'espace entier des données, peu de propositions permettent de l'explorer tout en s'intéressant à certaines localités. Or, les explications locales, parfois distinctes des mécanismes mis en jeu, ne permettent l'interprétation du réseau de neurones que si elles sont nombreuses et se complètent. Une méthode d'exploration globale et locale de l'espace de représentation et du comportement du réseau de neurones dans cet espace est donc une piste prometteuse pour une meilleure interprétation des réseaux de neurones.

## 3 Méthode

La figure 1 montre une vue d'ensemble de notre approche. Nous en détaillons les étapes dans les sections suivantes.

### 3.1 Le réseau et l'entraînement

Pour le plongement lexical, nous utilisons word2vec de Mikolov et al. (2013a). Le réseau que nous entraînons est un réseau auto-encodeur qui encode les phrases dans un espace de plus petite dimension (Hinton et Salakhutdinov, 2006). Il est constitué de **deux** couches d'unités de portes récurrentes ("Gated Recurrent Unit") (Cho et al., 2014). Chaque couche possède des états cachés ("hidden states") de dimension **512**, ce qui fait que la représentation de la phrase est encodée à l'aide de **1 024** valeurs réelles. Cette réduction de dimension conserve le maximum d'informations possible car la tâche de décodage consiste à reconstruire la phrase

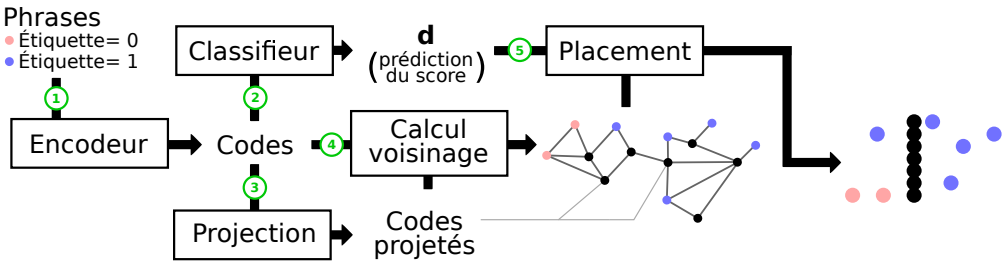


FIG. 1 – *Intégralité de la méthode de visualisation de la frontière de décision d'un réseau de neurones en classification automatique de textes. Les étapes s'appliquent selon l'ordre suivant : ①, ②, ③, ④, ⑤, ⑥. Nous décrivons les étapes d'encodage (①) et de classification (②) en section 3.1 puis la projection (③) en section 3.2. Une description du calcul du voisinage (④) est donnée en section 3.3. Le placement (⑤) est décrit en sections 3.4 et 3.5.*

telle que le réseau l'a reçue. Le calcul de la fonction de coût se fait à l'aide d'une fonction d'entropie croisée ("cross entropy function"). Dans nos cas d'étude (voir section 4), le pas d'apprentissage évolue de 0,05 à 0,0084 au cours de l'apprentissage (neuf epochs).

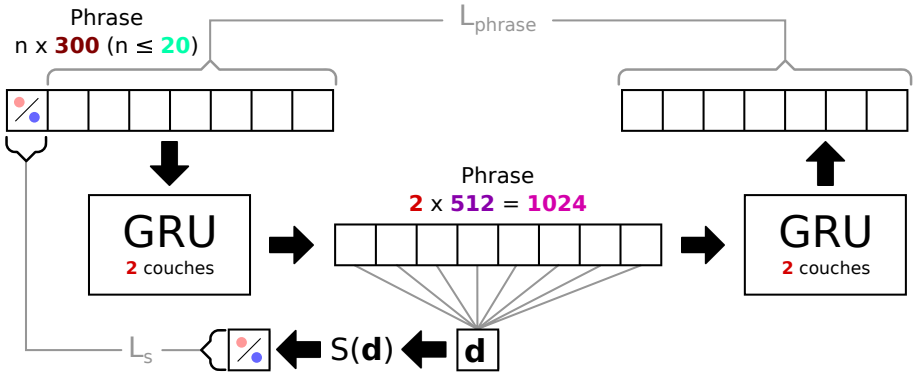


FIG. 2 – *Réseau d'auto-encodage des phrases et procédure de classification de la variable* ***étiquette****. Une fois la classification effectuée, la variable* ***d*** *produite est négative si la classe prédite est 0 et positive si la classe prédite est 1. La valeur absolue de cette variable représente la distance à la frontière de décision de notre classification. Cette variable est ramenée dans l'intervalle [0,1] pour le calcul de la fonction de coût à l'aide de la fonction sigmoïde (S(x)).*

À la suite de l'encodage, une fois que le réseau affiche de bonnes performances (dans nos cas d'étude, une perplexité à 2,01), nous injectons le code produit dans une régression linéaire multiple ("fully-connected layer") qui, suivie d'une *sigmoïde*, prédit la variable dichotomique **étiquette**. On ré-entraîne alors notre réseau. Le calcul de la fonction de coût de la classification est réalisé avec une fonction d'entropie croisée binaire ("binary cross entropy"). Au final $L_{phrase}$, $L_s$ représentent respectivement les calculs des fonctions de coût pour la reconstruc-

tion de la phrase et la prédiction de la variable **étiquette**. Notre fonction de coût finale est de la forme $L_{finale} = (1-\alpha)\times L_{phrase} + \alpha \times L_s$, avec $\alpha$ = 0,9. Dans nos cas d'étude (voir section 4), la perplexité de la tâche d'encodage est de 1,33, et le coefficient de corrélation de Matthews (1975) (MCC) de la tâche de classification est de 0,71 (le MCC, compris en -1 et 1, est égal à 1 lorsqu'une corrélation parfaite existe entre les prédictions et les **étiquettes**, et -1 quand une corrélation négative parfaite existe entre les prédictions et les **étiquettes**).

## 3.2 Projection des points sur la frontière de décision

Une fois toutes les données encodées, nous construisons les données se trouvant sur la frontière de décision. La frontière de décision dans l'espace de représentation est dessinée par un hyperplan, qui coupe l'espace de représentation en deux. L'orientation de cet hyperplan est contrôlée par un vecteur normal $\boldsymbol{\beta}$ (i.e. orthogonal à l'hyperplan). Chaque vecteur de représentation $\boldsymbol{z}$ peut se décomposer comme la somme d'un vecteur $\boldsymbol{u}$ qui appartient à l'hyperplan (i.e. $\langle \boldsymbol{\beta}, \boldsymbol{u} \rangle = 0$) et d'un vecteur $v$ co-linéaire à $\boldsymbol{\beta}$ ou nul. Nous calculons les vecteurs de représentation des données projetées (sur la frontière de décision) en projetant orthogonalement sur l'hyperplan l'ensemble des vecteurs de représentation de nos données : $\boldsymbol{u} = \text{proj}_{\beta}(\boldsymbol{z}) = \boldsymbol{z} - \langle \boldsymbol{\beta}, \boldsymbol{z} \rangle \boldsymbol{\beta} / \left\| \boldsymbol{\beta} \right\|_2^2$. Les représentations de la frontière de décision ainsi créées seront donc aussi nombreuses que le jeu de données d'entrée.

## 3.3 Calcul du voisinage des données

Dans l'ensemble de représentation des données et de leurs projections sur la frontière, nous calculons l'ensemble simplicial flou associé aux données. Cette méthode proposée par Zadeh (1965) est utilisée dans l'algorithme de réduction de dimension UMAP (McInnes et Healy, 2018) [2]. Pour ce faire, Nous créons un ensemble flou simplicial pour chaque donnée. Chacune de ces données aura un référentiel de distance propre à elle-même fixé en fonction de la proximité avec ses voisins. Ce référentiel, étant flou, construit des probabilités de voisinage entre les données. On combine ensuite tous les ensembles flous simplicials locaux en un ensemble grâce à une union floue. Cette union nous donne des liens de voisinage entre les points, ce qui nous permet de construire un graphe (voir figure 3a-c). Cette union donne aux couples de données deux probabilités différentes d'être voisins, l'une résultant du référentiel de distance de la première donnée et l'autre résultant du référentiel de distance de la seconde donnée. La distance finale entre les deux données est la probabilité d'existence d'au moins une des probabilités de voisinage ($a + b - a \cdot b$, with $a, b$ des probabilités). Une distance non nulle construit donc un lien entre deux données dans le graphe de proximité produit.

## 3.4 Séparation des composantes connexes de trop grande taille

Dans le graphe obtenu, nous divisons les plus grandes composantes de la façon suivante (voir figure 3d). Nous commençons par définir deux valeurs maximum pour le nombre de sommets et le nombre d'arêtes par composante connexe. Pour chaque composante connexe dépassant le nombre de sommets ou d'arêtes maximum, un algorithme calculant la centralité intermédiaire des arêtes ("betweeness centrality") est exécuté (Brandes, 2001; Newman et

2. Une description détaillée du fonctionnement de UMAP disponible ici : `https://umap-learn.readthedocs.io/en/latest/how_umap_works.html`

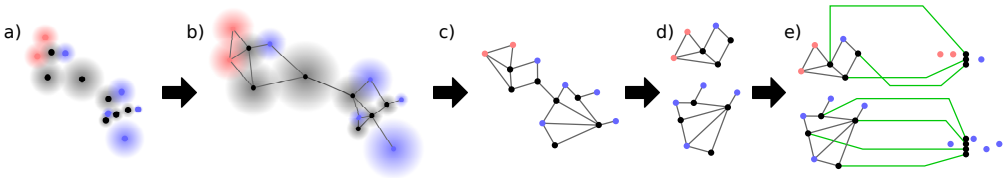


FIG. 3 – *Création de graphe et placement des données (représentées par ● ou ● selon si leur* ***étiquette*** *est 0 ou 1) et des données projetées (représentées par ●). En a) sont représentés les ensembles flous simpliciels (cercles fondus) de toutes les données. Le graphe en b) est produit par l'union des distances entre deux données dans leurs ensembles flous respectifs. Le graphe en c) est le graphe final de toute nos données. Le graphe en d) est produit à la suite de la séparation des composantes. Le graphe en e) est issu du placement des données projetées sur la frontière et du placement des données. Les données à gauche de la frontière de décision sont donc prédites à 0, celles à droite à 1. Les traits* ***verts*** *montrent le placement d'un point de frontière dans la visualisation finale.*

Girvan, 2004). Ensuite, l'arête avec la plus grande centralité est supprimée pour diviser la composante en deux. On ré-exécute cette procédure tant que les composantes obtenues dépassent les seuils fixés.

## 3.5 Arrangement linéaire de la frontière de décision et positionnement des données pour une composante

Notre objectif est de représenter les points projetés sur la frontière de décision sur une ligne et les données autour en fonction de leur distance à la frontière. Afin d'ordonner les points de la frontière, nous dérivons de celle-ci un graphe dont les sommets sont ces points et les liens représentent leur proximité, l'objectif étant de trouver l'ordre des sommets minimisant la somme de la longueur des liens. Ce problème est connu sous le nom d'arrangement linéaire minimum et est NP-complet. Dans notre contexte, nous utilisons l'heuristique de Mcallister (1999) couplé avec la stratégie de permutation des sommets de Rodriguez-Tello et al. (2008) pour en trouver une solution approchée.

À la suite du placement de la frontière de décision, pour chacune des composantes, nous plaçons les données (voir figure 3e) selon la procédure itérative suivante : nous ordonnons nos données, n'étant pas des points projetés, de la plus proche à la plus éloignée de la frontière de décision. Cela détermine pour chaque donnée sa position en abscisse. Puis, pour déterminer leur position en ordonnée, nous calculons la médiane de la position de ces différents voisins déjà placés et nous répétons cette procédure jusqu'à avoir itéré sur toutes les données d'une composante. Les données qui n'auraient pas pu être placées durant la première procédure sont placées selon la même procédure au cours des itérations suivantes. Les données se trouvant dans des composantes sans point de frontière ne sont pas affichée. Dans nos cas d'étude (voir section 4), 0,1% des données ne sont pas affichées.

# 4 Cas d'étude

Cette section présente deux cas d'étude illustrant les bénéfices que l'on peut avoir en terme d'interprétabilité grâce à notre méthode. Dans la prochaine section, nous présentons les données utilisées. Nous décrivons les cas d'étude dans les deux sections suivantes.

## 4.1 Données et localité

Le jeu de données AmazonReview [3] (He et McAuley, 2016) comprend des avis anglophones sur différents produits vendus sur Amazon. Ce jeu contient une **étiquette** qui représente la note donnée par les utilisateurs (entre 1 et 5). Nous filtrons les **étiquettes** pour ne garder que celles égales à 1 ou 5 afin d'obtenir une variable binaire (0 et 1). Nous ne conservons que les avis comportant au plus 20 mots. Finalement, le jeu de données d'entraînement pour la classification comporte 1,3 million d'entrées. Pour ces travaux de visualisation, nous travaillons avec un échantillon de 50 000 phrases. En amont, nous avons construit un jeu de données d'entraînement pour l'encodage des phrases de 2,5 millions d'entrées à l'aide d'une concaténation de deux jeux de données : un de Botha et al. (2018) [4] et l'autre de Bowman et al. (2015) [5]. Ce sont des phrases anglaises de tous types. La première partie des données a été utilisée pour entraîner des réseaux de neurones à découper des phrases en deux parties et à reformuler chacune de ces parties. La deuxième partie des données a été utilisée pour entraîner des réseaux de neurones à inférer si une phrase est en accord, désaccord ou est neutre vis à vis d'une autre.

Les cas d'étude présentés ici sont choisis sur la plus grande localité construite par notre méthode. Elle est composée de 662 données (projetées ou non). Les données non projetées se répartissent comme indiqué dans la table 1. Cette localité comporte en proportion 91% de données en plus dans la classe négative que dans le jeu de données complet et par conséquent 21% de moins dans la classe positive que dans le jeu de données complet. Pareillement, notre prédiction tend à prédire la classe négative 77% plus souvent et la classe positive 26% moins souvent que dans le jeu de données complet. Cette composante a un coefficient de corrélation de Matthews de 0,75. Notre réseau de neurones est donc légèrement plus efficace dans cette localité que dans les autres. Enfin, lorsque l'on est confronté à différentes localités de l'espace de représentation des phrases, il est important de comprendre ce qui fait la particularité de chacune de ces localités. Ainsi, pour compléter la frontière de décision, un classement des mots les plus pertinents (Sievert et Shirley, 2014) est établi par localité, de manière à identifier ce qu'on peut y trouver. Ce classement qui permet de trouver les phrases qui, dans la visualisation de la frontière de décision, contiennent ces mots, est présenté en figure 4.

| Prédiction \| Étiquette | Négatif | Positif | $\sum$ |
|---|---|---|---|
| Négatif | 112 (0,34) | 36 (0,11) | 148 (0,45) |
| Positif | 6 (0,02) | 177 (0,53) | 183 (0,55) |
| $\sum$ | 118 (0,36) | 213 (0,64) | 331 |

TAB. 1 – *Répartition des classifications et des données sur une localité produite.*

3. http://jmcauley.ucsd.edu/data/amazon/index_2014.html
4. https://github.com/google-research-datasets/wiki-split
5. https://nlp.stanford.edu/projects/snli/

## 4.2 Voisin classés différemment et abords de la frontière

Dans ce cas d'étude, nous observons une phrase et ses trois phrases voisines directes dans la localité. Comme présentés en figure 5), cette phrase et tous ses voisins sont classés différemment. On peut observer que la proximité de ces phrases est due soit à l'utilisation d'un début de phrase similaire (ici, "je pense") soit au sujet (mots "livre" ou "lire" présent). La phrase sélectionnée et mal classée avec beaucoup de certitude est la suivante : "Je [pensais] que ce ne serait pas une bonne idée d'économiser en achetant un exemplaire d'occasion de ce livre.". Ici le réseau n'a pas capturé le fait qu'une affirmation commençant ainsi se termine souvent par dire que l'on se trompait en pensant cela. Cela explique cette très mauvaise prédiction.

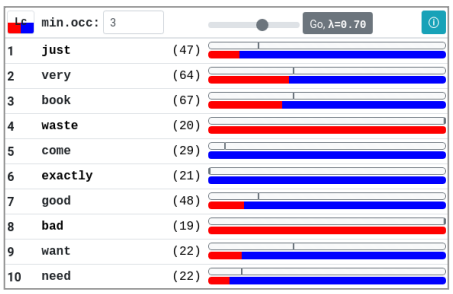


FIG. 4 – *Mots les plus pertinents.*

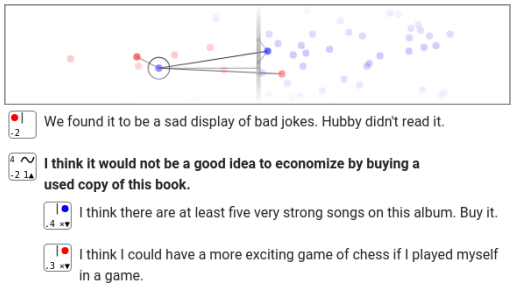


FIG. 5 – *Phrases proches.*

## 4.3 Cohérence de la classification pour les phrases similaires

Nous allons maintenant nous pencher sur une phrase ayant de nombreux voisins. Dans la figure 6, les traductions des phrases encadrées sont les suivantes :

— Le livre est arrivé en peu de temps, et était en très bon état. (encadré **rose**)
— Le livre est arrivé au bout de quelques jours seulement et il était en très bon état. (encadré **cyan**)
— Le livre est arrivé dans l'état qui était indiqué et je l'ai reçu en quelques jours. (encadré **marron**)

Ces phrases parlent toutes d'un livre, dans un état donné, arrivé rapidement. Comme on peut le voir dans l'espace à deux dimensions issu de la réduction de dimension par l'algorithme UMAP (McInnes et Healy, 2018), elles se trouvent proches les unes des autres dans l'espace de représentation. Néanmoins les nuances dans chacune amènent des disparités de classification visibles dans notre visualisation. La première phrase est la plus éloignée de la frontière, alors que la dernière est la plus proche. Les incertitudes sont faibles (d'ordre $10^{-2}$ ou moins). Lorsque l'on s'intéresse au contenu de ces phrases, être arrivé en peu de temps semble mieux qu'être arrivé en quelques jours. De même, être en très bon état semble mieux qu'être dans l'état indiqué. Ces observations peuvent induire que la compréhension des nuances faite par le réseau de neurones semble correcte et l'amène à classer ces phrases avec certitude et cohérence.

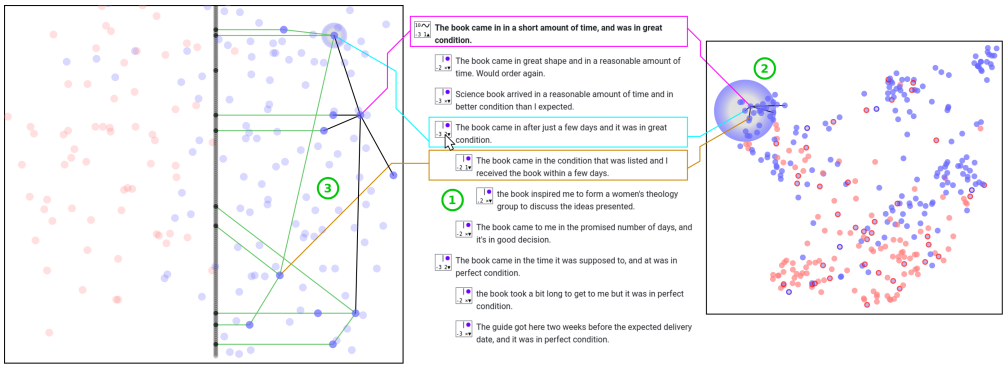


FIG. 6 – *Visualisation des classifications de phrases similaires. Ici, en ① se trouvent les phrases et les chemins possibles d'une phrase jusqu'à la frontière de décision. En ②, se trouve une visualisation de l'espace de représentation produit par l'algorithme UMAP (McInnes et Healy, 2018). On peut observer que les phrases étudiées sont en effet très proches dans l'espace de représentation. En ③, se trouvent les données et la frontière, on peut ainsi inspecter les distances à la frontière de décision pour ces phrases proches et observer des disparités.*

# 5 Conclusion

Dans cet article, nous avons proposé une nouvelle méthode pour expliquer globalement le fonctionnement des réseaux de neurones pour une tâche de classification automatique de texte basée sur la visualisation de la frontière de décision. Les méthodes actuelles n'offrent pas l'opportunité de visualiser facilement à quel point un réseau de neurones peut être certain de sa prédiction. Notre méthode offre, face à ce constat, une aide à l'interprétabilité des réseaux de neurones. La visualisation de la frontière de décision permet la visualisation d'espaces de grande dimension, aussi, c'est une méthode d'exploration innovante qui doit ensuite amener à des traitements plus fins des parties des espaces de représentation. Nos futurs travaux porteront sur l'amélioration des associations entre les informations produites par la visualisation de la frontière de décision et d'autres métriques pouvant aider à l'interprétabilité notamment le caractère justifiable des explications justifiable (Laugel et al., 2019).

# 6 Remerciements

Ce travail a été soutenu et subventionné par la Région Occitanie [Programme "Allocation Doctorale 2019"] et le SIRIC Montpellier Cancer [Grant INCa_Inserm_DGOS_12553].

# Références

Bahdanau, D., K. Cho, et Y. Bengio (2015). Neural machine translation by jointly learning to align and translate. *Computing Research Repository (CoRR) abs/1409.0473*.

Botha, J. A., M. Faruqui, J. Alex, J. Baldridge, et D. Das (2018). Learning to split and rephrase from Wikipedia edit history. In *Proceedings of the 2018 ACL Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 732–737.

Bowman, S. R., G. Angeli, C. Potts, et C. D. Manning (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 ACL Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 632–642.

Brandes, U. (2001). A faster algorithm for betweenness centrality. *The Journal of Mathematical Sociology 25*(2), 163–177.

Cho, K., B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, et Y. Bengio (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1724–1734. Association for Computational Linguistics.

Collobert, R., J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, et P. Kuksa (2011). Natural language processing (almost) from scratch. *J. Mach. Learn. Res. 999888*, 2493–2537.

Devlin, J., M.-W. Chang, K. Lee, et K. Toutanova (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, pp. 4171–4186. Association for Computational Linguistics.

Dimopoulos, Y., P. Bourret, et S. Lek (1995). Use of some sensitivity criteria for choosing networks with good generalization ability. *Neural Processing Letters 2*(6), 1–4.

Goodman, B. et S. Flaxman (2016). Eu regulations on algorithmic decision-making and a "right to explanation". *AI Magazine 38*, 50–57.

He, R. et J. McAuley (2016). Ups and downs : Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*, pp. 507–517.

Hinton, G. et S. Roweis (2002). Stochastic neighbor embedding. In *Proceedings of the 15th International Conference on Neural Information Processing Systems*, NIPS'02, Cambridge, MA, USA, pp. 857–864. MIT Press.

Hinton, G. E. et R. R. Salakhutdinov (2006). Reducing the dimensionality of data with neural networks. *science 313*(5786), 504–507.

Hohman, F., M. Kahng, R. Pienta, et D. H. Chau (2019). Visual analytics in deep learning : An interrogative survey for the next frontiers. *IEEE Transactions on Visualization and Computer Graphics (TVCG) 25*(8), 2674–2693.

Karpathy, A., J. Johnson, et L. Fei-Fei (2015). Visualizing and understanding recurrent networks. *ArXiv abs/1506.02078*.

Laugel, T., M.-J. Lesot, C. Marsala, X. Renard, et M. Detyniecki (2019). The dangers of post-hoc interpretability : Unjustified counterfactual explanations. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2801–2807.

Li, J., W. Monroe, et D. Jurafsky (2016). Understanding neural networks through representation erasure. *ArXiv abs/1612.08220*.

Lipton, Z. C. (2018). The mythos of model interpretability : In machine learning, the concept of interpretability is both important and slippery. *Queue 16*(3), 31–57.

Matthews, B. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure 405*(2), 442 – 451.

Mcallister, A. J. (1999). A new heuristic algorithm for the linear arrangement problem. Technical report, Faculty of Computer Science, University of New Brunswick.

McInnes, L. et J. Healy (2018). Umap : Uniform manifold approximation and projection for dimension reduction. *ArXiv abs/1802.03426*.

Mikolov, T., K. Chen, G. S. Corrado, et J. Dean (2013a). Efficient estimation of word representations in vector space. *Computing Research Repository (CoRR) abs/1301.3781*.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, et J. Dean (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS)*, Volume 26, pp. 3111–3119. Curran Associates, Inc.

Newman, M. E. J. et M. Girvan (2004). Finding and evaluating community structure in networks. *Phys. Rev. E 69*, 026113.

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2*(11), 559–572.

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

Raganato, A. et J. Tiedemann (2018). An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 ACL Empirical Methods in Natural Language Processing (EMNLP) - Workshop BlackboxNLP : Analyzing and Interpreting Neural Networks for NLP*, pp. 287–297.

Ribeiro, M. T., S. Singh, et C. Guestrin (2016). "why should i trust you ?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM International Conference on Knowledge Discovery & Data Mining (SIGKDD)*, pp. 1135–1144.

Rodriguez-Tello, E., J.-K. Hao, et J. Torres-Jimenez (2008). An effective two-stage simulated annealing algorithm for the minimum linear arrangement problem. *Computers & Operations Research 35*(10), 3331 – 3346. Part Special Issue : Search-based Software Engineering.

Selvaraju, R. R., M. Cogswell, A. Das, R. Vedantam, D. Parikh, et D. Batra (2020). Grad-cam : Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision (IJCV) 128*(2), 336–359.

Sievert, C. et K. Shirley (2014). LDAvis : A method for visualizing and interpreting topics. In *Proceedings of the ACL Workshop on Interactive Language Learning, Visualization, and Interfaces*, pp. 63–70.

Smilkov, D., N. Thorat, B. Kim, F. Viégas, et M. Wattenberg (2017). Smoothgrad : removing noise by adding noise. *ArXiv abs/1706.03825*.

Smilkov, D., N. Thorat, C. Nicholson, E. Reif, F. Viégas, et M. Wattenberg (2016). Embedding projector : Interactive visualization and interpretation of embeddings. *ArXiv abs/1611.05469*.

Socher, R., A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, et C. Potts (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642.

Springenberg, J. T., A. Dosovitskiy, T. Brox, et M. A. Riedmiller (2015). Striving for simplicity : The all convolutional net. *Computing Research Repository (CoRR) abs/1412.6806*.

van der Maaten, L. et G. Hinton (2008). Visualizing data using t-sne. *Journal of Machine Learning Research (JMLR) 9*(86), 2579–2605.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, et I. Polosukhin (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, Volume 30, pp. 5998–6008. Curran Associates, Inc.

Xu, Y., S. Biswal, S. R. Deshpande, K. O. Maher, et J. Sun (2018). Raim : Recurrent attentive and intensive model of multimodal patient monitoring data. In *Proceedings of the 24th ACM International Conference on Knowledge Discovery & Data Mining (SIGKDD)*, pp. 2565–2573.

Young, T., D. Hazarika, S. Poria, et E. Cambria (2018). Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine 13*(3), 55–75.

Zadeh, L. (1965). Fuzzy sets. *Information and Control 8*(3), 338 – 353.

Zenkel, T., J. Wuebker, et J. DeNero (2019). Adding interpretable attention to neural translation models improves word alignment. *ArXiv abs/1901.11359*.

## Summary

In text classification, many recent works deal with the interpretation of neural networks by producing explanations of predictions. The original approach presented in this paper consists in visualizing the decision boundary and the positioning of our data with respect to it, thus offering a new approach to explanation. Our method first computes a sentence representation space and then exploits its linear structure in order to visualize the distribution and clustering of data around the decision boundary. The main contribution of our method is the decision boundary visualization process, allowing to explore the distance to the decision boundary (and thus the certainty of a network in its predictions) but also the paths leading to it or the proximity between sentences.