Synthèse

Transparence des algorithmes de Deep Learning
pour les données textuelles assurantielles*

C. Francon, K. Larbi et W. Sanchez

21 mai 2021

Problématique

La Société Générale Assurances collecte et reçoit une grosse volumétrie de données textuelles à travers différents canaux et notamment par mail. Ces mails proviennent d'agents du réseau ou de clients et sont centralisés par un département (par exemple, lorsqu'un client souhaite se renseigner, gérer ses contrats ou se plaindre auprès d'un service). Ces messages doivent être redirigés vers le bon département : initialement, un opérateur humain lisait tous ces mails et triait en fonction du contenu vers le service le plus pertinent. Actuellement seuls les cas complexes ou particuliers sont traités par cet opérateur. Compte tenu du coût pour réaliser cette action, des modèles de classification textuelle ont été développés au sein du Datalab de la Société Générale Assurances, afin d'assigner à chaque mail reçu, un service vers lequel il devrait être transféré.

Depuis quelques années, les utilisations des réseaux de neurones se multiplient. En effet, ces modèles permettent de répondre à un large spectre de problématiques : classification d'images en utilisant des réseaux de neurones à convolution (ou *Convolutional Neural Network*) (AlexNet [KSH12]), traitement du langage par l'intermédiaire de réseaux de neurones (Word2Vec [Mik+12], FastText [Boj+16], Bert [Dev+19]).

Malgré des performances remarquables, ces méthodes souffrent d'un manque d'interprétabilité. Il est délicat d'obtenir des éléments expliquant les prédictions d'un modèle constituant généralement un frein pour les équipes métiers. Cette problématique est bien connue en *machine learning*, allant même jusqu'à renommer les réseaux de neurones de *modèles boîtes noires*.

Le but de ce projet est donc d'étudier des méthodes de transparence des algorithmes sur une sélection de modèles et de comparer les résultats à des données réelles. On s'intéressera à trois méthodes : une qui exploitera la structure du modèle d'apprentissage (GradCAM) et deux méthodes agnostiques d'interprétabilité (LIME et SHAP).

Présentation des modèles et des données

Description

Étant donné la valeur stratégique des données de l'entreprise, les modèles ont été entraînés sur le jeu de données open-source *allocine_review* contenant des avis sur des films issus du site *allocine*. Les avis ont été écrits entre 2006 et 2020. Pour chaque avis, l'utilisateur indique une note comprise entre 0.5 et 5 sur 5 : si la note est inférieure ou égale à 2 alors l'avis est considéré comme négatif sinon il est considéré comme positif.

Les avis subissent des prétraitements (*preprocessing*). Ces traitements permettent, par exemple, de découper une phrase en une liste de mots (ou même caractères) ou encore de prendre la racine d'un mot pour éviter de coder de manière différente des mots très proches (par exemple : "chantez", "chant", "chanson" renvoient à la même idée, ces trois mots seront remplacés dans les phrases par le mot "chant"). Certains mots, appelés *stop-word*, apportent peu d'informations pour la prédiction (par exemple, les prépositions) : ils sont donc supprimés lors des traitements de la

*Projet disponible sur Github : https://github.com/ENSAE-CKW/nlp_understanding

base de données. Ce processus nécessaire afin d'éviter des problèmes de surapprentissage (mot rare par exemple) ou d'augmenter artificiellement la dimensionnalité du problème.

Les modèles qui ont été utilisés lors de ce projet sont des architectures déjà en production au sein de la Société Générale Assurances. Ils ont en commun l'utilisation de couches de convolution, qui seront utilisées pour la méthode GradCAM. En plus de ces derniers, un modèle de régression logistique basé sur des sacs de mots (BoW) a été développé en tant que benchmark. Voici donc les performances des différentes architectures :

— Un réseau qui prend en entrée des mots et qui est composé d'une couche d'*embedding* (*Fast Text*), d'une couche de convolution et d'une couche de *maxpooling* puis de *fully-connected*
— Un modèle qui prend en entrée des caractères et qui est composé de plusieurs couches de convolution et d'une couche de *maxpooling* puis de *fully-connected*
— Un réseau qui prend en entrée des mots et qui est composé d'une couche d'*embedding*(*word2vec* pré-entraîné), d'une couche BiLSTM, d'une couche de convolution et d'une couche de *maxpooling* puis de *fully-connected*

Performance des modèles

Voici les résultats obtenus avec nos différents modèles sur la base de données *allocine_review* :

Table 1 – Résultats sur base de test

Modèles	AUC	Précision	Temps d'entraînement
Bag of words (n=100)	0.74	82%	7min
Embedding (Fast Text) + CNN	0.96	89%	8.2min
Caractères CNN	0.97	92%	32min
Embedding (w2v) + Bi-LSTM + CNN	0.97	92%	12.7min

Les trois réseaux de neurones obtiennent des résultats assez similaires tandis que le modèle régression logistique est un peu moins bon.

Ces trois réseaux ont été implémentés dans le but de pouvoir comparer les différentes méthodes d'interprétabilité.

Méthodes d'interprétabilité

Présentation

Une méthode d'interprétabilité a pour but de comprendre quels sont les éléments qui ont permis à un modèle donnée d'assigner aux observations telle ou telle classe. Trois différentes méthodes d'interprétabilité ont été utilisées dans ce projet :

— la méthode LIME : il s'agit d'une méthode agnostique. Nous avons utilisé un package pour appliquer cette méthode.
— la méthode SHAP avec l'algorithme KernelSHAP : il s'agit d'une méthode agnostique. Nous avons utilisé un package pour appliquer cette méthode.
— la methode GradCAM : il s'agit d'une méthode qui utilise la structure spécifique d'un CNN. Il est donc obligatoire d'avoir une couche de convolution dans son réseau pour l'utiliser. Cette méthode est courante dans le cas de classification d'images mais peu courante dans un problème du traitement du langage. Nous avons implémenté nous-même GradCAM.

Nos résultats

Deux points de vue sont possibles pour l'interprétabilité :

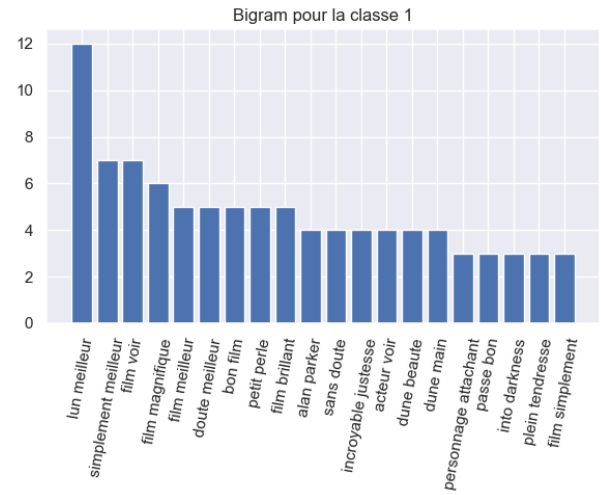
— un point de vue global : quels mots ou groupes de mots ont le plus influencé la prédiction du modèle ?
— un point de vue local : pour une observation donnée, quels mots ou groupes de mots ont permis au modèle de le classer ainsi ?

Point de vue global

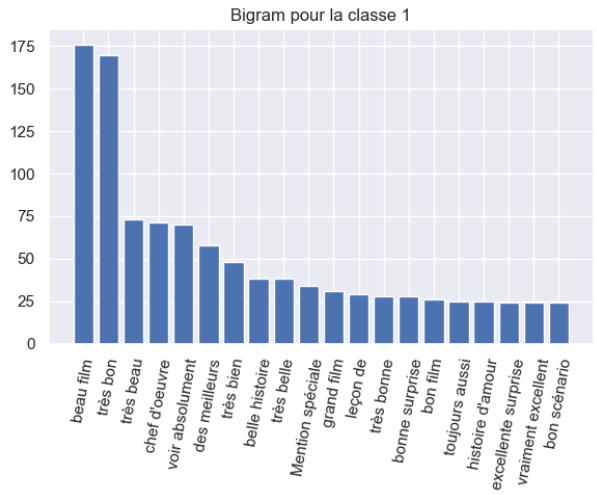
Dans un premier temps, nous nous sommes intéressés au point de vue global. Nous avons pour cela utilisé la méthode d'interprétabilité que nous avons implémentée : GradCAM [1].

Nous avons d'abord cherché les mots les plus importants dans la prédiction d'une classe donnée. Néanmoins, nous nous sommes rendu compte que des mots comme "très", ou bien "film" apparaissaient importants pour les deux classes. Nous avons donc établi l'hypothèse suivante : certains de ces mots, qu'on retrouve à la fois dans l'importance de la classe 0 et 1 pour tous les modèles, sont des mots qui peuvent être suivis ou précédés d'autres mots qui, eux, peuvent signaler le caractère positif ou négatif d'un commentaire. Ainsi, ils sont importants car les filtres des couches de convolution (selon leurs tailles) vont conjuguer le sens des mots porteurs du sentiment d'un commentaire sur ces mots "génériques". Une partie du signal est donc distribuée à ces mots.

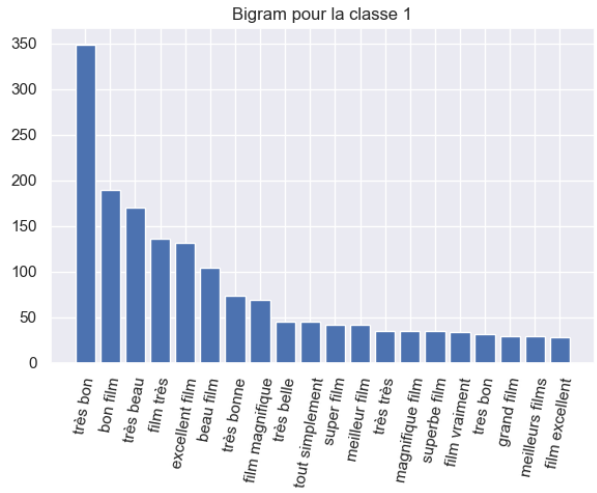
Pour tester cette hypothèse, nous avons voulu mettre en avant les *bi-gram* (groupe de 2 mots) qui sont les plus influents dans la prédiction d'une classe donnée. En exploitant les couches de convolution, la méthode GradCAM permet de mettre en lumière ces *bi-gram* d'importance. Voici donc les résultats pour l'explication de la classe 1 :



(a) Vanilla CNN



(b) Caractères CNN



(c) Bi-LSTM CNN

FIGURE 1 – Fréquence des bi-grams les plus importants pour la classe 1, pour les commentaires classifiés positifs

1. https://github.com/ENSAE-CKW/nlp_understanding/tree/master/src/deep_nlp/grad_cam

En effet, les mots comme "très" ou "film" se retrouvent dans les bi-gram des deux classes, mais sous la forme "très bon"/"très mauvais"" et "excellent film"/"film ennuyeux".

Point de vue local

Puis, nous avons utilisé les trois méthodes présentées auparavant pour un point de vue local. Pour cela, plusieurs observations de la base de données de test ont été choisis. Nous avons fait le choix de ne pas prendre des exemples où les modèles de classification marchaient très bien car, en pratique, ce n'est pas très utile de comprendre pourquoi un modèle a bien fonctionné. Quatres exemples types ont été choisis :

- un exemple classé positivement par le modèle (probabilité d'appartenance à la classe "Positif" proche de 1) et dont l'étiquette est "Négatif".
- un exemple classé négativement par le modèle (probabilité d'appartenance à la classe "Positif" proche de 0) et dont l'étiquette est "Positif".
- un exemple où un modèle prédit très mal alors que les deux autres modèles prédisent correctement
- un exemple pour lesquels le modèle a eu des difficultés à classer (probabilité d'appartenance à la classe "Positif" environ de 0.5)

D'une manière générale, les trois méthodes d'interprétabilité sont intéressantes quand le modèle prédit correctement le commentaire. Même s'il est parfois possible de comprendre pourquoi le modèle est indécis (grâce à SHAP et au LIME notamment), il est souvent difficile de comprendre pourquoi une prédiction n'est pas bonne. Dans ces cas-là, il est compliqué de savoir si le problème vient de la méthode d'interprétabilité ou du modèle de classification. Par exemple, lorsque des mots comme "Gérard Jugnot" apparaissent importants dans la prédiction du modèle, *a priori,* il est difficile de savoir si cela provient d'un problème de sur-apprentissage des modèles, ou bien des méthodes d'interprétabilité.

Conclusion

Le but de ce projet était de construire un outil d'interprétabilité pour les équipes métiers. Il existe des méthodes agnostiques comme SHAP ou LIME mais chacune présente des limites. Par exemple pour LIME, il est nécessaire d'ajuster les hyperparamètres d'un modèle pour construire des interprétations. Pour autant, il n'existe pas de fonctions objectif permettant de quantifier l'interprétabilité d'une explication et donc ces hyperparamètres ne sont pas le fruit d'un problème d'optimisation et ne peuvent être trouvés automatiquement avec des méthodes numériques. Des règles du pouce sont donc utilisées pour trouver ces paramètres.

La méthode GradCam permet de prendre en compte la structure des réseaux de neurones contenant des couches convolutionnelles. Ces méthodes semblent donner des résultats plus concluants que LIME ou Sharp sauf dans le cas où le modèle est indécis. De plus, cette méthode ne nécessite pas d'utilisation de surmodèles : il n'est pas nécessaire d'apprendre un nouveau modèle et donc d'ajouter une source supplémentaire d'approximation et d'erreur.

Ces méthodes ont été utilisées sur du texte ou des images. Cependant, les réseaux de neurones convolutionnels sont également utilisés pour la prédiction pour des séries temporelles. Un reproche pouvant être fait à certaines méthodes de prédiction en série temporelle est le manque d'explication. En effet, ces explications peuvent être fondamentales si ces prédictions sont utilisées par le décideur public par exemple. On pourrait utiliser ces méthodes d'interprétabilité par exemple pour avoir des éléments explicatifs lors de prédictions de séries temporelles comme des prédictions démographiques.

Bibliographie

[1] Alex Krizhevsky, Ilya Sutskever et Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In : *Advances in Neural Information Processing Systems*. Sous la dir. de F. Pereira et al. T. 25. Curran Associates, Inc., 2012. url : https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[2] Tomas Mikolov et al. “Distributed Representations of Words and Phrases and their Compositionality”. In : (2012). url : https://papers.nips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf.

[3] Yoon Kim. “Convolutional Neural Networks for Sentence Classification”. In : *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar : Association for Computational Linguistics, oct. 2014, p. 1746-1751. doi : 10.3115/v1/D14-1181. url : https://www.aclweb.org/anthology/D14-1181.

[4] Karen Simonyan et Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In : *CoRR* abs/1409.1556 (2014). url : http://arxiv.org/abs/1409.1556.

[5] Xiang Zhang, Junbo Zhao et Yann LeCun. “Character-level Convolutional Networks for Text Classification”. In : *Advances in Neural Information Processing Systems*. Sous la dir. de C. Cortes et al. T. 28. Curran Associates, Inc., 2015. url : https://proceedings.neurips.cc/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf.

[6] Bolei Zhou et al. “Learning Deep Features for Discriminative Localization.” In : *CoRR* abs/1512.04150 (2015). url : http://dblp.uni-trier.de/db/journals/corr/corr1512.html#ZhouKLOT15.

[7] Piotr Bojanowski et al. “Enriching Word Vectors with Subword Information”. In : *https ://arxiv.org/pdf/1607.04606.pdf* (2016).

[8] Ian J. Goodfellow, Yoshua Bengio et Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. Cambridge, MA, USA : MIT Press, 2016.

[9] Marco Tulio Ribeiro, Sameer Singh et Carlos Guestrin. “"Why Should I Trust You ?" : Explaining the Predictions of Any Classifier”. In : (2016), p. 1135-1144.

[10] Peng Zhou et al. “Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling”. In : *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics : Technical Papers*. Osaka, Japan : The COLING 2016 Organizing Committee, déc. 2016, p. 3485-3495. url : https://www.aclweb.org/anthology/C16-1329.

[11] Ramprasaath R. Selvaraju et al. “Grad-CAM : Visual Explanations from Deep Networks via Gradient-Based Localization.” In : *ICCV*. IEEE Computer Society, 2017, p. 618-626. isbn : 978-1-5386-1032-9. url : http://dblp.uni-trier.de/db/conf/iccv/iccv2017.html#SelvarajuCDVPB17.

[12] Jacob Devlin et al. “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 : Association for Computational Linguistics, juin 2019, p. 4171-4186. doi : 10.18653/v1/N19-1423. url : https://www.aclweb.org/anthology/N19-1423.

[13] Aston Zhang et al. *Dive into Deep Learning*. http://www.d2l.ai. 2019.

[14] I. Elizabeth Kumar et al. “Problems with Shapley-value-based explanations as feature importance measures”. In : *CoRR* abs/2002.11097 (2020). arXiv : 2002.11097. url : https://arxiv.org/abs/2002.11097.