Apprentissage actif profond pour le classement de textes en plusieurs classes

Yves Mercadier*, Jérôme Azé*, Sandra Bringay*,**

*LIRMM UMR 5506, Université de Montpellier, CNRS, France
name@lirmm.fr, http://www.lirmm.fr/
**AMIS, Département MIAp, Université Paul-Valéry Montpellier, France

Résumé. Récemment, le classement de documents textuels a beaucoup progressé. Cependant, les modèles utilisés doivent généralement s'entraîner au préalable avec de nombreux échantillons étiquetés. Il est possible de diminuer ce nombre d'échantillons en choisissant mieux les données à annoter via des techniques d'apprentissage actif. Cela peut permettre de diminuer le coût du processus en réduisant l'intervention humaine. Dans cette étude, nous adapterons les techniques récentes d'apprentissage actif profond utilisées pour le classement d'images, au cas de l'analyse de textes. En particulier, nous serons attentifs à l'apport de l'apprentissage actif profond selon l'architecture utilisée (LSTM ou CNN). Nous validerons nos hypothèses sur des jeux de données de la littérature.

1 Introduction

En classement de textes, la phase d'étiquetage nécessaire à l'apprentissage du classifieur peut s'avérer longue et fastidieuse. Dans ce contexte, l'apprentissage actif, pendant lequel l'oracle intervient pour choisir les exemples à étiqueter, s'avère prometteur. L'intuition est la suivante : en choisissant les exemples intelligemment et non aléatoirement, les modèles devraient s'améliorer avec moins d'effort et donc à moindre coût (c'est-à-dire avec moins d'exemples annotés). Dans cet article, nous conduisons une étude dans l'intention d'évaluer la qualité des processus d'apprentissage actif pour une tâche spécifique de classement multi-classes de textes.

Dernièrement, les réseaux de neurones se sont avérés très efficaces pour le classement de textes, notamment en utilisant des classifieurs de type LSTM (Long Short-Term Memory) et CNN (Convolutional Neural Network). Or, si beaucoup d'approches d'apprentissage actif profond ont été évaluées pour de le classement d'images, à notre connaissance, il n'existe que peu d'études portant sur le texte et sur ce type de classifieur.

Pour cette raison, nous allons évaluer dans cet article les méthodes de réseaux profonds combinées à des approches par apprentissage actif afin de généraliser à de gros volumes de données les connaissances acquises sur un petit échantillon.

2 État de l'art

Le type classement supervisé muti-classes consiste à apprendre un modèle à partir d'un ensemble de données préalablement annotées permettant d'associer une étiquette à un exemple, puis à partir de ce modèle, de prédire une étiquette pour un nouvel exemple donné. L'étiquetage est généralement réalisé par des humains, ce qui représente une tâche fastidieuse et coûteuse.Nous nous intéressons ici au processus d'apprentissage actif dont l'objectif est d'optimiser l'intervention humaine pendant la phase d'étiquetage des données, en posant les questions les plus pertinentes à l'oracle pour améliorer les performances du modèle, avec l'objectif de réduire la quantité de questions posées.

Il existe de nombreux classifieurs utilisés avec succès sur des données textuelles comme les classifieurs SVM, les arbres de décision, etc. Récemment, des architectures de type LSTM et CNN se sont révélées particulièrement efficaces sur les textes. Dans des travaux précédents Mercadier et al. (2018), nous avons montré l'efficience de ces architectures pour ce type de tâche. Les couches LSTM ont tout d'abord été introduites par Hochreiter et Schmidhuber (1996) et c'est la combinaison de couches bidirectionnelles LSTM qui s'est révélée très efficace pour le classement de textes (Zhou et al., 2016). Les approches CNN ont été introduites par LeCun et Bengio (1998). Elles reposent sur le principe des filtres de convolution qui ont d'abord été utilisés pour les images, puis ont donné de très bons résultats sur diverses tâches de traitement automatique de la langue naturelle (Kim, 2014; Schwenk et al., 2017), dont le classement de textes. Des études comparatives de ces deux types d'architectures ont été réalisées notamment par Yin et al. (2017). Il apparaît que les hyper-paramètres ont une grande influence sur les résultats, même si les réseaux LSTM semblent plus efficaces pour le classement de textes. À notre connaissance, il n'existe pas d'étude comparant l'apport de l'apprentissage actif profond de textes pour ces deux architectures.

L'apprentissage actif de modèles supervisés permet de sélectionner durant la phase d'entraînement les échantillons à étiqueter au lieu de les prendre au hasard dans les données encore non étiquetées. L'objectif est alors, pour atteindre les mêmes performances d'un modèle, de réduire le nombre d'exemples choisis. Cette sélection se conduit sous différentes stratégies décrites par Settles (2009) dont les plus utilisées sont les suivantes :

— Apprentissage passif (*Baseline*) : le lot d'exemples présenté à l'oracle pour être étiqueté est choisi de manière aléatoire dans l'ensemble non étiqueté ;
— Apprentissage actif par échantillonnage incertain (*Uncertainty sampling*) : le lot d'exemples est choisi en fonction de la règle de décision du classifieur afin de repérer ceux pour lesquels le modèle est le plus incertain en se basant sur une fonction de coût comme l'entropie ;
— Apprentissage actif de type bayésien (*Deep bayesian*) : Gal et al. (2017) ont proposé une amélioration de la mesure d'incertitude précédente en utilisant la technique de *Monte Carlo dropout*. L'entropie a été de nouveau utilisée comme fonction de coût ;
— Apprentissage actif de type variation du gradient (*Expected gradient*) : Bouneffouf (2014) choisissent le lot d'exemples parmi ceux qui modifient le plus les poids du réseaux, c'est-à-dire ceux ayant la plus grande norme de gradient ;
— Corset : Sener et Savarese (2018) proposent à l'oracle les échantillons se répartissant le plus homogènement possible dans l'espace de représentation choisi. Pour cela, ils résolvent un problème de positionnement de k centres (k-*center problem*) en utilisant

l'algorithme *Farthest-first traversal*.

La littérature sur l'apprentissage actif appliqué au classement de textes est importante (Olsson, 2009; Figueroa et al., 2012) mais porte essentiellement sur des algorithmes de classement statistique. Comme évoqué précédemment, les classifieurs de textes les plus performants sont actuellement de type réseaux de neurones. Aussi, il convient d'évaluer la combinaison d'une architecture de type apprentissage profond et du processus d'apprentissage actif. Cette combinaison a déjà été explorée dans le cas du classement d'images (Wang et Ye, 2013; Sener et Savarese, 2018) ou de contenus multimédias Budnik (2017). Or, peu de propositions ont été faites dans le cas de données textuelles. Nous pouvons citer les travaux de Shen et al. (2017) qui ont travaillé sur la reconnaissance d'entités nommées qui est une tâche différente de celle que nous voulons réaliser. Bang et al. (2018) ont proposé une approche basée sur un réseau RNN qui ne nécessite pas d'extraire des caractéristiques particulières mais utilise son état interne pour traiter des séquences d'entrées. Zhang et al. (2017) ont utilisé un réseau CNN selon une stratégie de type gradiant attendu appliquée à la couche de plongement (lexical). Siddhant et Lipton (2018) ont réalisé une large étude empirique sur l'application de l'apprentissage actif profond pour de multiples tâches, qui montre que dans la plupart des contextes, l'apprentissage actif de type bayesien, surpasse les autres approches. Nous ne retrouvons pas dans les articles cités précédemment d'étude combinant les derniers algorithmes d'apprentissage actif profond avec des classifieurs de type LSTM comme nous nous proposons de le faire dans ces travaux.

Dans cet article, nous nous intéresserons donc à la mise en place d'un processus d'apprentissage actif profond spécifique pour le classement multi-classes de données textuelles. Nous évaluerons cinq stratégies d'apprentissage actif qui se sont montrées efficaces pour le classement d'images. Nous comparerons l'apport de ce processus d'apprentissage actif profond pour les deux architectures désormais de référence pour le classement de textes : LSTM et CNN. Nous utiliserons quatre corpus de textes aux particularités différentes.

3 Méthode

Les méthodes d'apprentissage actif profond fonctionnent globalement très bien pour certains types de données comme les images (Gal et al., 2017). Il reste à adapter ces méthodes à l'analyse des textes et actuellement peu d'applications existent (Siddhant et Lipton, 2018).

3.1 Description des textes

En apprentissage profond, il est courant de faire appel à une description des textes par des vecteurs dans un espace de représentation continu appelé plongement (*embedding*). Les premières solutions comme, word2vec (Mikolov et al., 2013) ou encore GLOVE (Pennington et al., 2014), ont permis une amélioration substantielle des classifieurs. FasText, une technique introduite par Joulin et al. (2017), construit les vecteurs de description à partir des n-grammes de chaque mot. Récemment, de nouvelles architectures très efficaces ont été proposé comme ELMo (Embeddings from Language Models) (Peters et al., 2018) ou BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018). Préalablement à cette étude, nous avons comparé sur les jeux de données de l'article les approches précédemment citées. ELMO et BERT se sont avérées les plus efficaces. Toutefois, dans ce travail, nous avons retenu

la représentation FasText qui représente un bon compromis entre temps de calcul et efficience pour des résultats assez proches en termes d'exactitude.

3.2 Apprentissage actif par lot

Nous considérons $\mathcal{L}$ un ensemble de données étiquetées et $\mathcal{U}$ un ensemble de données non étiquetées. Il est classique en apprentissage actif de procéder au questionnement de l'oracle question après question. Le modèle est entraîné sur le jeu de données étiquetées. Puis, un unique nouvel exemple est proposé à l'oracle pour étiquetage. Avec un classifieur de type réseaux de neurones, ce fonctionnement n'est pas exploitable car il n'est pas pertinent d'entraîner le réseau sur un seul texte à la fois. C'est pour cette raison que nous procédons par lot. Cela revient à demander à l'oracle d'étiqueter un lot d'exemples et non plus un unique exemple. On recherche donc un ensemble d'exemples à étiqueter dans $\mathcal{U}$ qui va maximiser les performances du classifieur. Cette sélection peut se faire selon différentes approches. Soit x^* l'ensemble des instances les plus informatives selon $\varphi(x_i, \theta)$ avec φ la fonction utilisée pour évaluer les instances x_i de $\mathcal{U}$ selon l'ensemble des paramètres du modèle θ.

$$x^* = \underset{x_i \in \mathcal{U}}{\text{argmax}} \varphi(x_i, \theta)$$

3.3 Cinq stratégies d'apprentissage actif

Nous allons comparer dans ce travail cinq stratégies d'apprentissage actif profond.

La première stratégie appelée **baseline** (Apprentissage passif) consiste à proposer à l'oracle une sélection aléatoire des échantillons.

La deuxième stratégie, appelée **Uncertainty Sampling** (Apprentissage actif par échantillonnage incertain), est connue depuis longtemps (Lewis et Catlett, 1994). Dans la littérature, plusieurs fonctions de coût ont été proposées pour le choix des exemples. Avec la mesure Least confidence (Culotta et McCallum, 2005), les échantillons les moins sûrs sont choisis pour être annotés. Pour évaluer la confiance, ces auteurs utilisent la probabilité de la classe la plus sûre de l'échantillon pour le modèle. Avec la mesure Smallest-margin (Scheffer et al., 2001), la sélection des échantillons se fait par la marge minimale entre les probabilités des classes de chaque échantillon. Avec l'Entropie (Shannon, 2001), le tri des échantillons s'opère à partir de l'estimation de la variabilité des probabilités des étiquettes. Généralement, c'est l'entropie qui est choisie car elle donne de bons résultats par son estimation qui agrège la totalité des probabilités de chaque exemple pour chaque étiquette.

$$x^* = \underset{x_i \in \mathcal{U}}{\text{argmax}} [- \sum_c P(y_i = c|x_i; \theta).log(P(y_i = c|x_i; \theta))]$$

Où c représente l'indice des classes, θ l'état du réseau.

La troisième stratégie, appelée **Expected Gradient** (Apprentissage actif selon la variation du gradient), a été appliquée par Zhang et al. (2017) pour le classement binaire de textes avec un réseau CNN. Elle revient à sélectionner les échantillons qui vont conduire à une modification maximale de la couche de plongement (lexical) du classifieur lors de sa phase d'apprentissage. Pour cela, les échantillons sont classés en fonction de la variation de la norme du

gradient sur la représentation de chaque mot pour toutes les étiquettes possibles, pour chaque échantillon.

$$x^* = \underset{x_i \in \mathcal{U}}{\text{argmax}}[average_{j \in x_i} \sum_c P(y_i = c|x_i;\theta) \mid\mid \nabla J_{E(j)}((x_i, y_i = c);\theta) \mid\mid]$$

Où $\nabla J_{E(j)}$ est le gradient de la couche de plongement (lexical) des mots j dans le document x_i.

La quatrième stratégie, appelée **Deep Bayesian** (Apprentissage actif bayésien profond), a été proposée pour une tâche de reconnaissance d'entités nommées (Siddhant et Lipton, 2018). Elle revient à sélectionner les échantillons qui maximisent le gain informationnel du modèle pour une succession de prédictions associées à différents abandon de neurone.

$$x^* = \underset{x_i \in \mathcal{U}}{\text{argmax}}[-\sum_c (\frac{1}{T}\sum_t \widehat{p}_c^t) log(\frac{1}{T}\sum_t \widehat{p}_c^t) + \frac{1}{T}\sum_{c,t} \widehat{p}_c^t log \widehat{p}_c^t]$$

Où T représente la norme de l'ensemble des projections et $\widehat{p}_c^t = P(y_i = c|x_i;\omega^t)$ la probabilité que l'entrée x_i soit de la classe c pour un abandon de neurone ω^t.

La dernière stratégie, appelée **Core-Set**, a été appliquée avec succès pour le classement d'images avec un classifieur de type CNN Sener et Savarese (2018). L'approche consiste à sélectionner de façon la plus homogène qui soit les échantillons dans un espace de représentation, par exemple la dernière couche du réseau de neurone. Il est courant d'utiliser pour cela l'algorithme de Gonzalez (1985) qui est une bonne approximation du problème du positionnement de k centre et qui permet de réduire les calculs.

3.4 Classifieurs LSTM et CNN

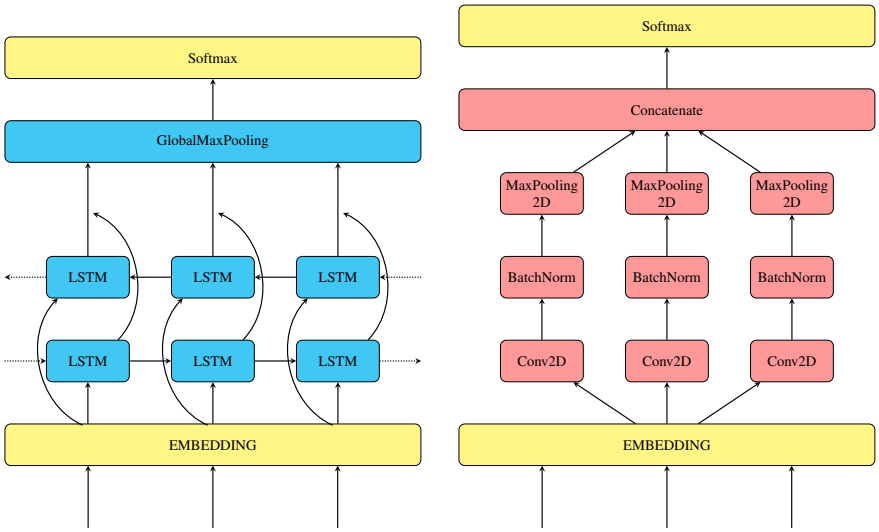
Nous utilisons deux types de classifieur décrits dans les figures 1a et 1b. Pour les deux architectures, le réseau est construit à partir d'une première couche de type plongement (lexical) où chaque mot est associé à un vecteur de dimension 300.

Concernant le réseau **LSTM** (*Long Short-Term Memory*), après cette première couche, on retrouve une couche de type LSTM bidirectionnelle permettant de faire une analyse de la structure des séquences proposées à l'entrée du réseau. LSTM est un sous-type de RNN (*Recurrent Neural Network*) efficaces pour les séquences de données. Ces réseaux, composés de plusieurs couches, estiment leurs sorties en fonction des états de leurs couches précédentes en utilisant une mémoire intermédiaire. Les LSTM sont basés sur des blocs de mémoire qui sont utilisés comme unités dans la couche récurrente pour capturer des dépendances à plus longue portée que les couches RNN classiques. Dans ces travaux, nous utiliserons une telle architecture en positionnant après la couche bidirectionnelle une couche de *global max pooling* comme dans (Sachan et al., 2019).

Les classifieurs de type **CNN** (*Convolutionnal Neural Network*) projettent d'abord les mots des textes dans une couche de plongement (lexical) et appliquent ensuite des opérations de convolution sur la matrice résultante. Dans ce travail, nous utilisons l'architecture introduite par Kim (2014), dans laquelle la couche de plongement (lexical) est suivie d'une couche de convolution en deux dimensions qui rend possible une analyse de la structure des séquences. Nous avons ajouté, par rapport à l'architecture initiale, une couche de *batch normalisation* afin

de limiter le sur-apprentissage comme Ioffe et Szegedy (2015). On trouve ensuite une couche de *pooling* qui résume l'information par région de traitement.

Enfin, on trouve en sortie des deux architectures LSTM et CNN, une couche totalement connectée comprenant un neurone par classe de document avec une fonction d'activation de type sigmoïde.



(a) Réseau LSTM (b) Réseau CNN

FIG. 1: Architecture des deux classifieurs pour le classement de textes.

4 Expérimentations

4.1 Jeux de données

Pour réaliser notre étude, nous avons choisi quatre jeux de données couramment utilisés dans les articles d'apprentissage profond pour des tâches de NLP (Joulin et al., 2017; Zhang et al., 2015). La tableau 1 présente des statistiques clés sur ces quatre jeux de données. Les jeux de données que nous utilisons sont équilibrés en classe.

Jeu de données	nombre de classes	longueur d'une séquence	nombre de documents	Tâche de classement
Amazon Review Polarity	2	37	3 600 000	Analyse de sentiments
Yelp Review Polarity	2	57	560 000	Analyse de sentiments
Ag news	4	20	120 000	Classement de texte
DBPedia	14	28	560 000	Classement selon une ontologie

TAB. 1: Jeux de données utilisés dans nos expériences pour le classement de textes. Voir Zhang et al. (2015) pour une description détaillée.

4.2 Pré-traitements des données

Pour chaque jeu de données, nous avons appliqué les pré-traitements suivants :

— suppression des ponctuations ;
— suppression des caractères spéciaux ;
— changement des majuscules en minuscules ;
— suppression des stop words ;
— lemmatisation consistant à ne conserver que le radical des mots, pour regrouper sous le même radical tous les mots d'une famille.

Chaque texte peut être associé à zéro, une ou plusieurs classes selon le jeu de données. Ces classes sont utilisées comme sortie de la prédiction.

4.3 Protocole d'évaluation

Nous procédons comme suit pour chaque jeu de données. Dans un premier temps, nous extrayons 20000 textes en respectant la pondération des classes que nous séparons en deux ensembles de 10000 textes toujours stratifiés. Le premier ensemble sert de jeu d'entraînement tandis que le deuxième ensemble sert de jeu de test. En ce qui concerne la phase d'apprentissage, le jeu d'entraînement est découpée en 6 sous ensembles. Le premier sous ensemble, très petit de 0.1% des 20000 documents, est utilisé pour démarrer le processus d'apprentissage. Pour cela, nous appliquons un algorithme de clustering *K-means* comme proposé dans Kang et al. (2004). Puis, pour chaque sous ensemble, une stratégie d'apprentissage actif est utilisée pour évaluer et ordonner les échantillons dans la perspective de questionner l'oracle pour une proportion de questions fixée. À la suite de l'étiquetage réalisé par l'oracle, le classifieur est entraîné puis est utilisé pour étiqueter le reste du sous ensemble sur l'ensemble de test. Pour estimer la qualité de l'apprentissage, nous utilisons la métrique d'exactitude, qui se calcule comme le rapport du nombre d'étiquettes correctement attribuées par le classifieur sur le nombre total d'étiquettes. Les jeux de données utilisés dans cette étude étant équilibrés en classe, nous ne donnons pas dans la suite les informations sur le rappel et la précision. Nous réitérons ce processus cinq fois et moyennons la valeur d'exactitude.

4.4 Hyper-paramétrage et entraînement

Dans notre étude, nous avons implémenté le système avec six lot de données. L'ensemble des hyper-paramètres du réseau est fixé pour l'ensemble des jeux de données : un taux d'apprentissage de 0.001, époques de 10, abandon de neurone de 0.2. Tous nos réseaux de neurones sont entraînés avec un processus de back-propagation utilisant une fonction d'erreur (*loss*) de type entropie croisée. Les optimiseurs de calcul du gradient sont de type *Rmsprop* pour les LSTM et *Adam* pour les CNN. Ils sont assortis d'un taux d'apprentissage de 0.001 décrémentés à chaque période. Nous limitons pour des raisons de temps de calcul une taille de dictionnaire de 10000 mots. La couche de plongement (lexical) est dynamique et entraînée avec le réseau sur 20 époques. Elle comprend des vecteurs de description composés de 300 dimensions. Chaque point de mesure sera recalculé 5 fois avec des conditions initiales stochastiques.

5 Résultats des expérimentations

Dans cette section, nous reportons les résultats de nos expérimentations dans la figure 2. Dans les différents graphes, nous représentons en abscisse la proportion des questions posées à l'oracle au regard du nombre de questions possibles, et en ordonnée, nous représentons l'exactitude du classifieur.

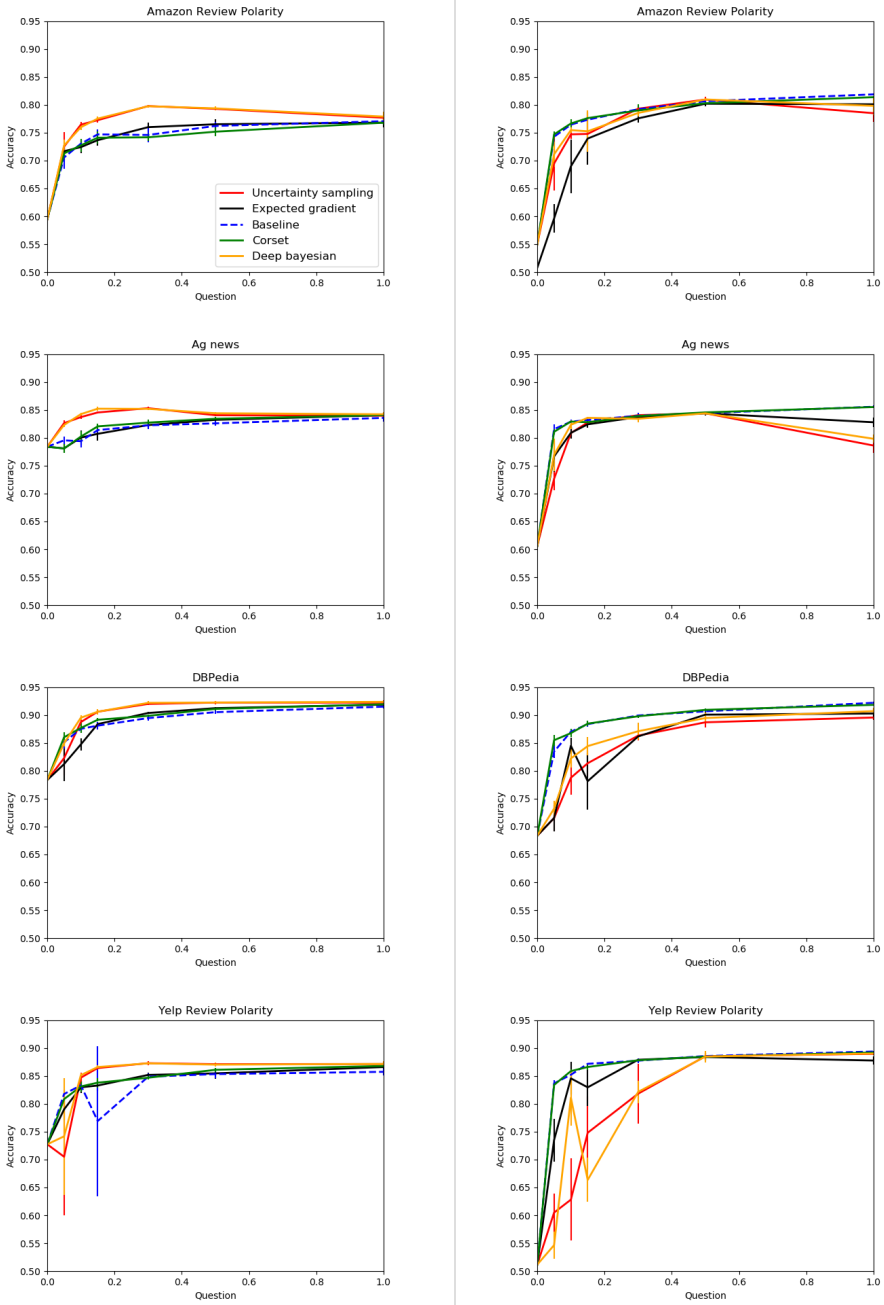
Comme nous pouvions nous y attendre, nous constatons pour les deux architectures, une amélioration des résultats avec l'augmentation du nombre d'échantillons fournis à l'oracle. Si l'apprentissage progresse fortement pour des taux de questionnement compris entre 0% et 30% dans la quasi totalité des situations, un palier est observé au-delà. Cela tendrait à indiquer que le classifieur ne tire pas parti des exemples supplémentaires de manière linéaire. L'effort d'étiquetage par un humain devrait alors se concentrer au niveau du coude sur la courbe.

Le réseau CNN obtient les meilleurs résultats sur tous les jeux de données avec le paramétrage de l'étude. Les méthodes d'apprentissage actif ne donnent pas de résultats améliorant un choix aléatoire de questions.Cela semble cohérent avec l'interprétation de Zhang et al. (2017), pour qui l'estimation et le réglage des plongements (pour une tâche spécifique de classement) peut être vu comme un problème d'apprentissage de représentation. Il est donc alors raisonnable d'optimiser ces vecteurs de caractéristiques avant d'améliorer les paramètres d'un modèle les recevant en entrée.

En revanche, pour une architecture de type LSTM, l'apprentissage actif améliore globalement les résultats pour toutes proportions de questions. Ce constat se vérifie pour les quatre corpus de textes étudiés. Cette amélioration semble être optimale aux alentours des 25% correspondant au coude sur la courbe. Finalement, il y a une stratégie d'apprentissage actif qui surpasse toutes les autres sur tous les jeux de données. Il s'agit de l'uncertainty sampling avec la fonction de coût entropie. Les réseaux Deep Bayesien de Siddhant et Lipton (2018) obtiennent toutefois des résultats très similaires.

6 Conclusion

Dans cet article, nous avons présenté une étude comparative des différentes stratégies d'apprentissage actif combinées à deux architectures très récentes d'apprentissage profond, LSTM et CNN, pour une tâche spécifique de classement de textes. Nous avons clairement montré d'une part qu'avec un réseau de type CNN, les résultats n'étaient pas améliorés quelque soit la stratégie d'apprentissage actif mise en œuvre. D'autre part avec les réseaux de type LSTM, la stratégie d'apprentissage actif par uncertainty sampling associée à la fonction de coût entropie s'est avérée être la plus efficace. Bien que ces travaux restent préliminaires, ils nous permettent de suggérer l'utilisation de la combinaison de réseaux de type LSTM et d'apprentissage actif dans le cas des petits jeux de données étiquetés. En perspective, nous nous proposons d'utiliser d'autres architectures de type CNN telles que VDCNN au niveau des caractères (Schwenk et al., 2017) de poursuivre cette étude comparative avec des plongement (lexical) de type ELMO (Peters et al., 2018) ou BERT (Devlin et al., 2018). Toujours dans l'objectif de pouvoir traiter de petits jeux de données étiquetés, nous envisageons d'explorer d'autres approches telles que l'augmentation des données (Belohlávek et al., 2018; Abulaish et Sah, 2019) qui consiste à générer de nouvelles données d'entraînement à partir des données existantes.



architecture LSTM architecture CNN

FIG. 2: L'abscisse représente la proportion des questions posées à l'oracle au regard du nombre de questions possibles et l'ordonnée, l'exactitude du classifieur.

Références

Abulaish, M. et A. K. Sah (2019). A text data augmentation approach for improving the performance of cnn. In *Proceedings of the 11th International Conference on Communication Systems & Networks*, COMSNETS 2019.

Bang, A., W. Wu, et H. Han (2018). Deep active learning for text classification. In *Proceedings of the 2nd International Conference on Vision, Image and Signal Processing*, ICVISP 2018, New York, NY, USA, pp. 22 :1–22 :6. ACM.

Belohlávek, P., O. Plátek, Z. Zabokrtský, et M. Straka (2018). Using adversarial examples in natural language processing. In *Proceedings of the 11th International Conference on Language Resources and Evaluation, Miyazaki, Japan, May 7-12, 2018.*, LREC 2018.

Bouneffouf, D. (2014). Exponentiated gradient exploration for active learning. *Computers 5*(1).

Budnik, M. (2017). *Active and deep learning for multimedia. (Apprentissage actif et profond pour le multimédia)*. Ph. D. thesis, Université Grenoble Alpes.

Culotta, A. et A. McCallum (2005). Reducing labeling effort for structured prediction tasks. In M. M. Veloso et S. Kambhampati (Eds.), *Proceedings of the Conference on Artificial Intelligence*, AAAI 2005, pp. 746–751. AAAI Press / The MIT Press.

Devlin, J., M. Chang, K. Lee, et K. Toutanova (2018). BERT : pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1*, NAACL-HLT 2018.

Figueroa, R., Q. Zeng-Treitler, L. H Ngo, S. Goryachev, et E. Wiechmann (2012). Active learning for clinical text classification : Is it better than random sampling ? *Journal of the American Medical Informatics Association : JAMIA 19*, 809–16.

Gal, Y., R. Islam, et Z. Ghahramani (2017). Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1183–1192. JMLR.org.

Gonzalez, T. F. (1985). Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science 38*, 293 – 306.

Hochreiter, S. et J. Schmidhuber (1996). Lstm can solve hard long time lag problems. In *Proceedings of the 9th International Conference on Neural Information Processing Systems*, NIPS'96, Cambridge, MA, USA, pp. 473–479. MIT Press.

Ioffe, S. et C. Szegedy (2015). Batch normalization : Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6-11 July 2015*, ICML 2015, pp. 448–456.

Joulin, A., E. Grave, P. Bojanowski, et T. Mikolov (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics : Volume 2*, ACL 2017, pp. 427–431. Association for Computational Linguistics.

Kang, J., K. R. Ryu, et H.-C. Kwon (2004). Using cluster-based sampling to select initial

training set for active learning in text classification. In H. Dai, R. Srikant, et C. Zhang (Eds.), *Advances in Knowledge Discovery and Data Mining*, Berlin, Heidelberg, pp. 384–388. Springer Berlin Heidelberg.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2014, Doha, Qatar, pp. 1746–1751. Association for Computational Linguistics.

LeCun, Y. et Y. Bengio (1998). The handbook of brain theory and neural networks. In M. A. Arbib (Ed.), *The Handbook of Brain Theory and Neural Networks*, Chapter Convolutional Networks for Images, Speech, and Time Series, pp. 255–258. Cambridge, MA, USA : MIT Press.

Lewis, D. D. et J. Catlett (1994). Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the 11th International Conference on Machine Learning*, ICML 1994, pp. 148–156. Morgan Kaufmann.

Mercadier, Y., J. Azé, S. Bringay, V. Clavier, E. Cuenca, C. Paganelli, P. Poncelet, et A. Sallaberry (2018). #aids analyse information dangers sexualité : caractériser les discours à propos du VIH dans les forums de santé. In *Actes des 29es Journées francophones d'Ingénierie des Connaissances, Nancy, France, July 4-6, 2018.*, IC 2018, pp. 71–86.

Mikolov, T., W.-t. Yih, et G. Zweig (2013). Linguistic regularities in continuous space word representations. In *Proceedins of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT-NAACL 2013, pp. 746–751.

Olsson, F. (2009). A literature survey of active machine learning in the context of natural language processing. Technical Report 2009 :06, SICS.

Pennington, J., R. Socher, et C. D. Manning (2014). Glove : Global vectors for word representation. In *Proceedings of the Empirical Methods in Natural Language Processing*, EMNLP 2014, pp. 1532–1543.

Peters, M. E., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, et L. Zettlemoyer (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1*, NAACL-HLT 2018, pp. 2227–2237.

Sachan, D., M. Zaheer, et R. Salakhutdinov (2019). Revisiting lstm networks for semi-supervised text classification via mixed objective function. *Proceedings of the Conference on Artificial Intelligence 33*, 6940–6948.

Scheffer, T., C. Decomain, et S. Wrobel (2001). Active hidden markov models for information extraction. In F. Hoffmann, D. J. Hand, N. Adams, D. Fisher, et G. Guimaraes (Eds.), *Advances in Intelligent Data Analysis*, Berlin, Heidelberg, pp. 309–318. Springer Berlin Heidelberg.

Schwenk, H., L. Barrault, A. Conneau, et Y. LeCun (2017). Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1 : Long Papers*, pp. 1107–1116.

Sener, O. et S. Savarese (2018). Active learning for convolutional neural networks : A core-set

approach. In *Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, April 30 - May 3, 2018*, ICLR 2018.

Settles, B. (2009). Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Shannon, C. E. (2001). A mathematical theory of communication. *Mobile Computing and Communications Review 5*(1), 3–55.

Shen, Y., H. Yun, Z. C. Lipton, Y. Kronrod, et A. Anandkumar (2017). Deep active learning for named entity recognition. In *Proceedings of the 2nd Workshop on Representation Learning for NLP, Vancouver, Canada, August 3, 2017*, Rep4NLP@ACL 2017, pp. 252–256.

Siddhant, A. et Z. C. Lipton (2018). Deep bayesian active learning for natural language processing : Results of a large-scale empirical study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, EMNLP 2018, pp. 2904–2909.

Wang, Z. et J. Ye (2013). Querying discriminative and representative samples for batch mode active learning. In *Proceedings of the 19th ACM International Conference on Knowledge Discovery and Data Mining*, KDD '13, New York, NY, USA, pp. 158–166. ACM.

Yin, W., K. Kann, M. Yu, et H. Schütze (2017). Comparative study of CNN and RNN for natural language processing. *CoRR abs/1702.01923*.

Zhang, X., J. Zhao, et Y. LeCun (2015). Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, Cambridge, MA, USA, pp. 649–657. MIT Press.

Zhang, Y., M. Lease, et B. C. Wallace (2017). Active discriminative text representation learning. In *Proceedings of the 31th Conference on Artificial Intelligence*, AAAI'17, pp. 3386–3392. AAAI Press.

Zhou, P., Z. Qi, S. Zheng, J. Xu, H. Bao, et B. Xu (2016). Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. In *Proceedings of the 26th International Conference on Computational Linguistics*, COLING 2016, Osaka, Japan, pp. 3485–3495.

Summary

Recently, there has been considerable progress in the classification of textual documents. However, the models used must generally be trained beforehand with many labelled samples. It is possible to reduce this number of samples in order to perform this task by better selecting the examples to be annotated using active learning techniques. This can reduce the cost of the process by reducing human intervention. In this study, we will adapt recent deep active learning techniques used for image classification to the case of text analysis. In particular, we will be attentive to the contribution of deep active learning depending on the architecture used (LSTM or CNN). We will validate our hypotheses on data sets from the literature.