Réseau de neurones à double convolution pour la classification de sentiments multi-domaines

Abdelhalim Rafrafi, Vincent Guigue, Patrick Gallinari

Laboratoire d'Informatique de Paris 6, UMR 7606 - UPMC
4 pl. Jussieu F-75005 Paris

Résumé :
Nous proposons d'utiliser un réseau de neurones pour classer les sentiments positifs et négatifs de commentaires issus du web participatif. Le réseau de neurones que nous présentons utilise une double convolution : dans chaque phrase, les mots sont projetés sur une première couche cachée. Ces représentations de mots sont fusionnées pour obtenir une représentation des phrases en dimension fixe (première convolution). Les phrases sont ensuite projetées sur une seconde couche cachée puis fusionnées au niveau du document (seconde convolution). Une dernière couche classique permet de prendre une décision globale. Cette approche présente la particularité de projeter les mots dans un espace sémantique continu de dimension paramétrable, à la manière des algorithmes à variables latentes de type PLSA ou LDA. Toutefois, la construction de cet espace est totalement supervisée, les coordonnées des mots étant apprises par rétro-propagation de l'erreur sur l'ensemble du réseau. Nous montrons l'intérêt de cette architecture sur la tâche de classification de sentiments. Les performances en mono-domaine sont intéressantes mais c'est surtout la robustesse lors du passage au multi-domaines qui distingue notre approche du reste de l'état de l'art. Nous proposons des résultats qualitatifs et quantitatifs pour illustrer le fonctionnement du réseau.

Mots-clés : Classification de sentiments, Opinion Mining, Réseau de neurones à convolutions, multi-domaines

1. Introduction

La fouille d'opinion est un domaine qui a émergé avec le web participatif (2.0) et les commentaires laissés par les utilisateurs (*user generated contents*). Six ans après leur article fondateur de 2002 (Pang *et al.*, 2002), Pang et Lee ont publié un état de l'art (Pang & Lee, 2008) qui fait encore autorité. Ils insistent sur la place particulière d'une tâche : la classification de sentiments. Toutes les autres tâches de la fouille d'opinion s'appuient à un moment ou un autre sur un classifieur de polarité. Prédire efficacement la polarité d'un

texte est donc un enjeu majeur pour l'ensemble du domaine. Les applications directes sont multiples, de l'estimation de ventes de jeux vidéos (Marcoux & Selouani, 2009) à la prédiction de votes au sénat américain (Gerrish & Blei, 2011) en passant par les outils de gestion de la e-réputation (Yi & Niblack, 2005). D'autres applications dérivent de la classification de sentiments comme la classification subjectivité/objectivité (Pang & Lee, 2004) ou la détection de faux commentaires ou *spam revue* (Jindal & Liu, 2007).

L'exploitation des modèles conçus pose un second problème : l'apprentissage de fonction de transfert. En effet, les approches à base d'apprentissage supervisé sont généralement optimisées en utilisant des données étiquetées par les utilisateurs eux-mêmes : sur certains sites, les revues accompagnées d'une notation quantitative, généralement sous forme d'un nombre d'étoiles compris entre 1 et 5. C'est le cas des corpus les plus utilisés dans la littérature (Pang *et al.*, 2002; Blitzer *et al.*, 2007; Whitehead & Yaeger, 2009). *In fine*, il est évident que les modèles appris seront exploités sur d'autres sources (non étiquetées). Cette étape est critique car elle remet en cause l'hypothèse i.i.d. Les documents de test ne suivent plus la même distribution que les données d'apprentissage. Plusieurs stratégies d'adaptation ont été développées pour quantifier et limiter les pertes liées au passage à un nouveau domaine pour le test (Blitzer *et al.*, 2007; Whitehead & Yaeger, 2009; Pan *et al.*, 2010; Glorot *et al.*, 2011). C'est le formalisme multi-domaines : les modèles sont appris sur un corpus et testé sur un corpus différent, traitant un autre thème (e.g. apprentissage sur des revues d'appareils photos et test sur des critiques de films). Ces techniques se basent sur de l'apprentissage semi-supervisé : une partie des données cibles est utilisée (avec ou sans) étiquette pour affiner le modèle initial. La plupart des premières publications ont utilisé la factorisation matricielle pour réaliser l'adaptation entre une source et une cible : (Blitzer *et al.*, 2007) avec un critère d'information mutuelle et (Pan *et al.*, 2010) en ajoutant un alignement de caractéristiques. Plusieurs approches ajoutent une dernière étape d'affinage utilisant quelques données étiquetées du domaine cible. (Dredze *et al.*, 2010) a testé l'utilisation de sources multiples pour améliorer les performances sur une cible. L'approche de (Glorot *et al.*, 2011) va plus loin, en montrant l'intérêt d'apprendre un auto-encodeur sur un très grand nombre de sources de manière non-supervisée avant d'apprendre la classification de sentiments.

Nous proposons d'utiliser une projection dans un espace sémantique continu afin de limiter les pertes de performances en multi-domaines. L'idée de base est la même que dans les algorithmes à variables latentes mais nous utilisons

une toute autre technique d'apprentissage : un réseau de neurones inspiré de (Collobert & Weston, 2008) et (Bengio *et al.*, 2000). Comme dans (Glorot *et al.*, 2011), l'approche passe à l'échelle et tire efficacement parti des grands corpus. Cependant, l'architecture que nous utilisons est très différente des auto-encodeurs. Nous avions développé une première stratégie mono-couche qui est décrite dans (Rafrafi *et al.*, 2011). Les résultats obtenus étaient très proches des SVM. Nous nous sommes ensuite intéressés à une architecture plus originale à deux niveaux de convolution : les documents sont découpés en phrases, elles-mêmes découpées en mots. La première convolution permet de fusionner toutes les représentations des mots d'une phrase dans un vecteur de taille fixe. La seconde convolution est effectuée au niveau du document pour regrouper les représentations des phrases. Un perceptron linéaire permet ensuite de proposer une classe pour le document. Cette architecture présente plusieurs avantages : elle donne de bonnes performances en mono-domaine ainsi qu'en multi-domaines et elle permet d'interpréter les prédictions en séparant les contributions des différentes phrases à la manière de (Pang & Lee, 2004).

En section 2., nous présentons le modèle connexionniste en détails. Les corpus de test sont décrits en section 3.. Nous donnons ensuite un aperçu qualitatif du fonctionnement de ce réseau de neurones (section 4.). Enfin, nous analysons les résultats quantitatifs en classification de sentiments (section 5.).

2. Modèle connexionniste

L'architecture du réseau de neurones à convolutions (RNC) est décrite en Fig. 1. Nous détaillons les mécanismes de propagation et de rétro-propagation puis nous discutons les choix architecturaux que nous avons effectués. Nous utilisons également un réseau à convolution simple (RNCS) comme référence. Celui-ci est une version simplifiée de (Collobert & Weston, 2008) et décrit en détail dans (Rafrafi *et al.*, 2011). Toutes les phrases sont alors fusionnées en entrée du réseau et il n'y a plus qu'une projection/convolution pour obtenir une représentation en dimension fixe. Par rapport à l'architecture de (Collobert & Weston, 2008), la convolution est somme et la projection sur la couché cachée ne prend en compte qu'un seul mot (sans fenêtrage).

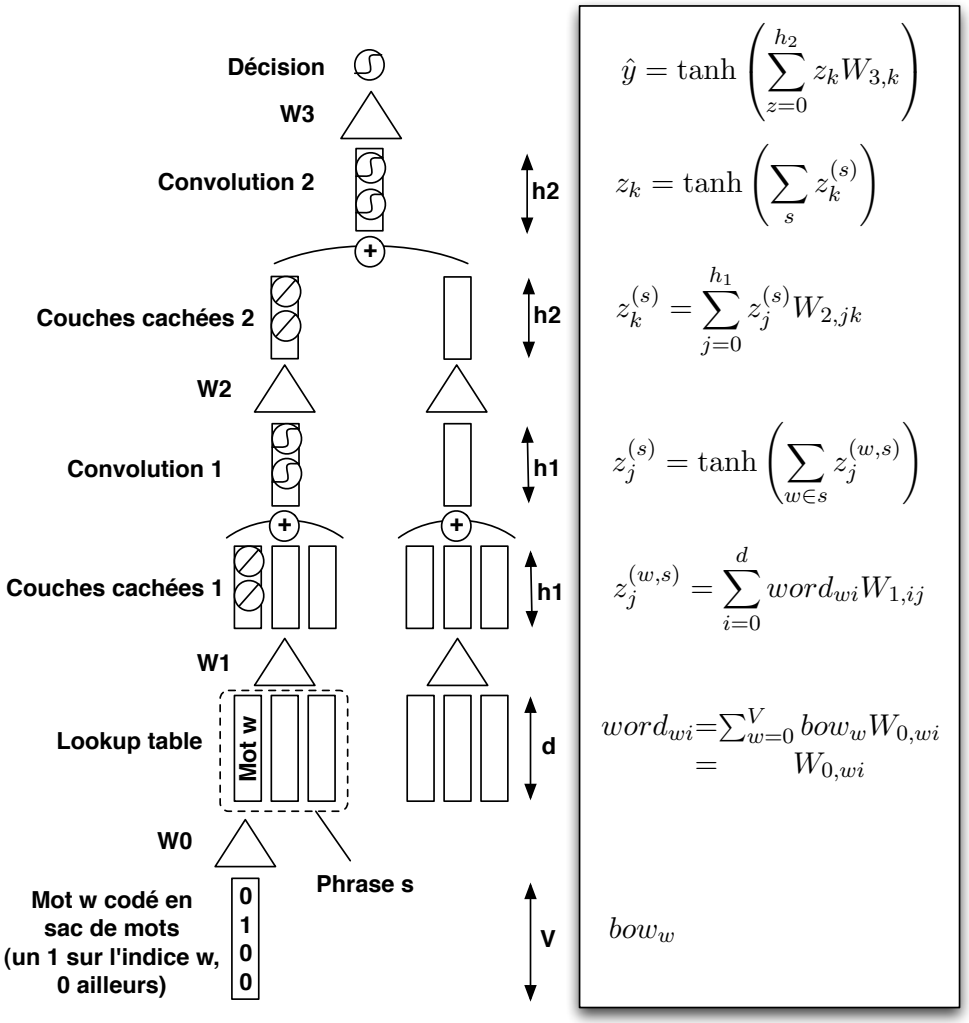

FIGURE 1: Réseau de neurones à convolutions. Les mots sont projetés sur une couche cachée en utilisant W_1, les phrases sont projetées en utilisant W_2. W_3 est le vecteur de paramètres du perceptron linéaire qui fournit une décision en sortie de réseau. Le tableau de droite donne les notations et les indices utilisés pour les sorties des différentes couches. Pour plus de détails sur les notations, voir le tableau 1.

2.1. Propagation : utilisation du réseau

Les mots sont projetés dans un espace continu $\mathbb{R}^d$ dont la dimension est un hyper-paramètre. Cette projection est stockée dans la matrice W_0 (cf Fig. 1). Tous les indices utilisés ainsi que les informations sur les paramètres appris (ou fixés par l'utilisateur) sont donnés dans le tableau 1. Les coordonnées des mots sont stockées dans une table de référencement (appelée *Lookup Table* dans l'article (Collobert & Weston, 2008)). Pour un document donné, les mots utilisés sont extraits de cette table et regroupés phrase par phrase. Les mots w d'une phrase s sont ensuite projetés sur une première couche cachée à l'aide de la matrice W_1 (cf. Fig 1) : $z_j^{(w,s)} = \sum_{i=0}^{d} word_{wi} W_{1,ij}$. Une première convolution permet d'obtenir une représentation de taille fixe pour chaque phrase. Ici, la convolution est une simple somme : $z_j^{(s)} = \tanh\left(\sum_{w \in s} z_j^{(w,s)}\right)$. La couche de convolution est donc de la même taille que les couches cachées des mots. Par soucis de clarté, nous avons noté l'activation de la convolution $\tanh$. En réalité, nous avons utilisé la formule classique $1.716 \tanh(\frac{2}{3}x)$ décrite dans (Bottou & Le Cun, 1996) afin de limiter les phénomènes de saturation [1]. Au niveau des phrases, la projection et la convolution sont les mêmes : $z_k^{(s)} = \sum_{j=0}^{h_1} z_j^{(s)} W_{2,jk}$ et $z_k = \tanh\left(\sum_s z_k^{(s)}\right)$. Une fois le problème ramené à une représentation en dimension fixe, nous utilisons un perceptron seuillé pour calculer la décision : $\hat{y} = \tanh\left(\sum_{k=0}^{h_2} z_k W_{3,k}\right)$.

2.2. Apprentissage : rétro-propagation et initialisation

L'apprentissage est basée sur une rétro-propagation classique (stochastique).

2.2.1. Rétro-propagation

Nous calculons le gradient de l'erreur par rapport aux différents paramètres W pour mettre à jour itérativement le réseau. L'apprentissage de la table de référencement ne pose pas de problème particulier. Il suffit d'imaginer une couche supplémentaire virtuelle dans le réseau : en entrée, nous fournirions un vecteur de la taille du dictionnaire avec un 1 pour l'indice du mot considéré (et 0 ailleurs). Une matrice de projection W_0 permettrait d'obtenir la représentation du mot w. Il est évident que cette représentation est en réalité identique

1. La fonction $\tanh$ tend rapidement vers 1 ou -1 et sa dérivée est alors proche de 0 : la retropropagation de l'erreur est alors stoppée. Le phénomène est connu sous la dénomination de saturation (cf Sec. 2.3.3.).

à la ligne w de la matrice W_0. La table de référencement et la matrice W_0 ne font qu'un et la mise à jour se fait en calculant le gradient du coût global par rapport aux paramètres W_0.

Nous utilisons deux critères d'arrêt : le premier sur le nombre d'itérations maximum (fixé à 500), le second lorsque tous les documents d'apprentissage sont bien classés. Une itération consiste à faire passer tous les documents d'apprentissage dans le réseau. Dans la pratique le second critère est toujours atteint en moins de 200 itérations.

2.2.2. Optimisation des hyper-paramètres

Notre modèle se base sur de nombreux hyper-paramètres. Trouver les bonnes valeurs des ces hyper-paramètres (taille des différentes couches, choix des fonctions d'activation, valeurs des pas de gradient ε_ℓ pour chaque couche) est un défi. Nous avons procédé classiquement, en calculant les performances en classification pour des grilles de valeurs. Pour limiter le nombre de tests, nous avons fait plusieurs hypothèses : les valeurs d'epsilon augmentent avec la profondeur du réseau (proportionnellement à la décroissance du gradient et donc à la taille des couches cachées). Nous n'avons pas testé de couches cachées de taille supérieures à 200, les temps d'apprentissage devenant particulièrement longs au delà.

Toutes les valeurs obtenues sont données dans le tableau 1.

2.2.3. Initialisation de l'apprentissage

Nous utilisons une initialisation gaussienne aléatoire pour la table de référencement. Les autres matrices de paramètres sont initialisées en utilisant une loi uniforme de paramètre $1/\sqrt{n}$, où n est le nombre de neurones de la couche de projection. Cette initialisation est discutée dans la section suivante ainsi que la section des résultats quantitatifs.

2.3. Discussions sur l'architecture

L'architecture de notre réseau présente plusieurs particularités que nous discutons ici.

Indices	
w	indice des mots dans le vocabulaire V
s	indice des phrases
i	indice sur la dimension de la table de référencement (*lookup table*)
j	indice sur la dimension de la première couche cachée, j indice également la dimension de la première couche de convolution
k	indice sur la dimension de la seconde couche cachée, k indice également la représentation du document.
Hyper-paramètres (& valeurs)	
$d = 10$	dimension de la table de référencement
$h_1 = 100$	dimension de la première couche cachée
$h_2 = 100$	dimension de la seconde couche cachée
$\varepsilon_0, \varepsilon_1, \varepsilon_2, \varepsilon_3$	pas de gradient pour la table de référencement, puis pour W_1, W_2, W_3. $\varepsilon_0 = 1, \varepsilon_1 = 10^{-5}, \varepsilon_2 = 10^{-5}, \varepsilon_3 = 10^{-7}$
Paramètres appris	
$W_1 \in \mathbb{R}^{d \times h_1}$	Matrice de projection de la table de référencement vers la première couche cachée
$W_2 \in \mathbb{R}^{h_1 \times h_2}$	Matrice de projection de la représentation des phrases vers la seconde couche cachée
$W_3 \in \mathbb{R}^{h_2}$	Paramètres du perceptron
$W_0 \in \mathbb{R}^{V \times d}$	Table de référencement.

TABLE 1: Notations et paramètres du réseau de neurones à convolutions

2.3.1. Analyse sémantique

Le fait de projeter les mots dans un espace continu fait immanquablement penser à l'algorithme Probabilistic Latent Semantic Analysis (Hofmann, 1999). En effet, cette approche non supervisée repose sur l'optimisation de la matrice $P(w|\theta)$ qui regroupe les probabilités d'apparition des mots pour k classes de données. La table des probabilités jointes peut être vue comme une projection des mots dans un espace continu. Il est possible d'analyser cet espace du point de vue linguistique et d'interpréter les distance entre mots (Pariollaud *et al.*, 2002).

Une telle analyse est plus complexe dans notre approche car nous n'utilisons pas un cadre probabiliste : les vecteurs obtenus ne sont pas normalisés. Nous avons tout de même tenté d'analyser l'espace optimisé mais nous n'en avons tirer aucune conclusion exploitable. Nous avons créé une sémantique qui n'est pas intelligible. Cependant, nous verrons que c'est cette sémantique qui nous permet d'obtenir de bonnes performances en multi-domaines.

2.3.2. Analyse au niveau de la phrase

Nous proposons une méthode pour estimer la contribution des phrases sur la classification du document général à la manière de (Pang & Lee, 2004; Zhai *et al.*, 2011). Alors que les réseaux de neurones sont souvent utilisés comme des boites noires, nous analysons les sorties des couches intermédiaires à la manière de (Féraud & Clérot, 2002). En négligeant la tangente hyperbolique de la seconde convolution en couche linéaire, il devient possible de calculer : $\hat{y}^{(s)} = \sum_{k=0}^{h_2} z_k^{(s)} W_{3,k}$

La valeur $\hat{y}^{(s)}$, où s référence la phrase, peut être utilisé de deux manières. Dans un premier temps, nous nous en sommes servis pour interpréter les résultats fournis par le réseau sur différents documents : cela donne des résultats intéressants qui sont détaillés en section 4.. En passant à la valeur absolue, ce score devient une mesure de subjectivité qui pourrait être utilisé pour filtrer les phrases et se focaliser sur les informations discriminantes [2]. Cependant, nous aboutissons à la même conclusion que (Pang & Lee, 2004) : nous n'avons pas réussi à améliorer nos performances en éliminant les phrases jugées objectives.

2.3.3. Le phénomène de saturation

Comme nous l'avons mentionné précédemment, le réglage des hyper paramètres et le choix des fonctions d'activation est un point clé de ce type d'architecture à convolutions. Nous avons testé un grand nombre de combinaisons et retenu finalement l'architecture la plus performante en se basant sur le taux de reconnaissance en validation croisée. L'architecture retenue (décrite précédemment) est loin d'être intuitive, elle pose même un certain nombre de problèmes pour l'apprentissage.

Comme le montrent clairement (Bottou & Le Cun, 1996) dans leur article, l'un des principaux obstacle rencontré dans l'apprentissage de réseaux de neurones à convolutions vient de la saturation des neurones. Lorsque l'entrée d'un neurone sinusoïdal présente une grande valeur absolue, sa tangente hyperbolique est très proche de 1 (ou -1). La dérivée associée, qui est utilisée pour la mise à jour des paramètres W, est alors très proche de 0 et le réseau n'évolue plus : l'apprentissage est figé. Il n'est pas possible non plus de se

2. Les revues sur lesquelles nous travaillons sont parfois divisées en deux parties : une partie subjective où l'auteur exprime son avis et une partie plus descriptive de l'objet qui risque de déstabiliser la décision.

limiter à des neurones linéaires car cela pose des problèmes de stabilité. Etant établi ce phénomène de saturation, l'architecture que nous avons proposée est particulièrement étonnante : les couches cachées sont à activation linéaire (non bornées) et la convolution est une somme (et non une moyenne). L'entrée des couches de convolution est donc potentiellement très grande en valeur absolue. Dans la pratique, la saturation est évitée en jouant sur l'initialisation : nous prenons des écarts-types très faibles pour initialiser les paramètres de notre modèle. Au fil des itérations, les paramètres grandissent mais le réglage des pas de gradient garantit que le réseau fonctionne correctement avant que ne surviennent d'éventuels problèmes de saturation.

3. Corpus, descripteurs & protocole expérimental

Afin de nous comparer avec l'état de l'art du domaine (Blitzer *et al.*, 2007; Whitehead & Yaeger, 2009; Pan *et al.*, 2010; Glorot *et al.*, 2011), nous avons utilisé des corpus de revues issues du site de e-commerce *Amazon*. Nous proposons une première série de résultats sur quatorze corpus puis nous nous concentrons sur quatre d'entre eux pour détailler les résultats. Ces revues concernent respectivement des livres (*books*), des dvd (*dvd*), des produits électroniques (*electronics*) et des articles liés à la cuisine (*kitchen*).

Ces données sont intéressantes pour étudier le problème d'apprentissage multi-domaines car deux des corpus sont assez proches (*books* et *dvd*) tandis que le troisième est plus général (*electronics*) et que le dernier repose sur un vocabulaire nettement différent (*kitchen*). En ce qui concerne les descripteurs, nous nous sommes limités aux unigrammes, sans traitement particuliers (ni lemmatisation, stemming ou autre). Nous avons tout de même éliminé les mots-outils (déterminants, pronom...) et les mots rares (qui n'apparaissent qu'une fois dans le corpus). Les caractéristiques détaillées des corpus sont fournies dans le tableau 2.

Corpus	Tailles des corpus (N)	Long. moy. des revues	Vocabulaire (V)
Books	2000	240	10536
Dvd	2000	235	10392
Electronics	2000	154	5611
Kitchen	2000	133	5314

TABLE 2: Descriptions des quatre principaux corpus utilisés.

Tous les résultats mono-domaines proposés sont calculés en validation

croisée (5 sous-ensembles, 80% des données utilisées en apprentissage)[3]. Pour le multi-domaines, le protocole est différent : nous apprenons sur l'ensemble des données d'un domaine et nous testons sur l'ensemble du domaine cible. Les résultats sont moyennés sur 5 expériences similaires pour ne pas dépendre des aléas du gradient stochastique.

4. Résultats qualitatifs

Les résultats qualitatifs que nous présentons dans cette section sont globalement négatifs comme nous le disions en section 2.3.1.. Malgré l'architecture originale qui promettait une interprétabilité au niveau des phrases, nous concluons finalement que le réseau se comporte comme une boite noire. Les performances (quantitatives) sont intéressantes mais ne sont pas analysables dans le détail.

4.1. Interprétation de l'espace sémantique

L'espace créé étant de grande dimension, nous avons testé deux approches pour interpréter les résultats : une interprétation graphique après une projection des mots dans un espace à deux dimensions (avec ACP et t-SNE) ; une interprétation locale en analysant les 15 mots les plus proches d'un mot requête (pour une liste de marqueurs sentimentaux reconnus). Nous n'avons pu tirer aucune conclusion satisfaisante de ces tentatives. Nous ne proposons pas d'illustration particulière tant les résultats nous semblent aléatoires.

4.2. Contributions des phrases des revues

Nous utilisons la formule définie en section 2.3.2. pour calculer la contribution des phrases sur le score global d'un document. Sur l'exemple proposé dans le tableau 3, nous voyons que le réseau a été capable de séparer relativement efficacement la partie descriptive (avec des scores plus faibles) des parties subjectives. Cette illustration de réussite doit cependant être relativisée car nous n'avons pas pu tirer statistiquement profit des mécanismes de sélection de phrases en nous basant sur ce score local. Comme (Pang & Lee, 2004)

3. En apprentissage, les performances sont toujours de 100% de reconnaissance. Cela est logique étant donné le rapport entre le nombre de revues et la taille du vocabulaire, cf tableau 2. Toute tentative de régularisation trop forte se traduit immédiatement par une perte de performance en test, ce phénomène est classique dans les applications en texte.

précédemment, la sélection des informations subjective est finalement préjudiciable sur les taux de reconnaissance de sentiments. Nous envisageons deux explications : soit aucune information n'est vraiment subjective (nous sommes cependant convaincu qu'une machine ne saurait pas détecter des nuances dans l'expression des sentiments) ; soit les informations de contexte sont importantes pour la classification de sentiments. Nous penchons pour cette seconde explication : un certain nombre d'article sont systématiquement plébiscités sur les site de e-commerce. Tout vocabulaire permettant d'identifier un tel produit peut conduire à une décision fiable sur la revue, sans tenir compte des marqueurs sentimentaux [4]. Cette hypothèse est renforcée par les travaux récents de (Zhang & Liu, 2011).

best leadership book i have read.	1.66
i have been studing leadership for over number number years both by reading and experience.	0.50
i consider this book to be the best book i have read because it describes leadership as a learned activity.	0.99
it also says that leadership is in motivating people to do their own work in solving difficult problems.	0.36
i found that as president of my congregation i was continually going back to the concepts in the book to lead it through a very difficult situation involving placement of the flags in the sanctuary.	0.54
it was very difficult to get people to do their own work and not try to step in to solve everything.	-0.45
(that would have been impossible anyway) i found that he described president lyndon johnson as a successful leader (civil rights) and unsuccessful leader (vietnam).	0.32
his discussion on leading without authority is new ground for me.	1.09
this is a great book with great stories of a variety of leaders in our society.	2.03

TABLE 3: Interprétation des scores par phrase sur une revue du corpus *Books*. Les parties subjectives sont en vert, elles sont associées à des scores plus élevés que les parties descriptives.

5. Résultats quantitatifs

Bien que les résultats qualitatifs soient difficilement interprétables, nous verrons dans cette section qu'ils sont intéressants du point vue quantitatif.

4. NB : nous sommes conscient qu'il s'agit d'une forme de sur-apprentissage, le problème étant alors de classer un produit et non un sentiment.

5.1. Impact de l'initialisation

Etant donnés la difficulté de régler le réseau de neurones (cf section 2.3.3.) et les problèmes de saturation, nous avons décidé de nous concentrer sur l'initialisation du réseau de neurones. L'idée était d'introduire des connaissances a priori, à la manière de (Glorot *et al.*, 2011), pour améliorer les taux de reconnaissance. Nous avons d'abord comparé des initialisations aléatoires en dimension 10 et 50 pour la table de référencement. Les résultats ne penchent pas significativement d'un coté ou de l'autre. Nous avons ensuite utilisé la matrice $P(w|\theta)$ issue de l'apprentissage LDA (*Latent Dirichlet Allocation*) sur d'autres corpus sentiments. Nous avons enfin récupéré la table de référencement de (Collobert & Weston, 2008), apprise sur l'ensemble de wikipedia anglophone et aboutissant à un modèle de langue. Quelles que soient les initialisations, les performances ne changent pas significativement comme le montre la figure 2.

Comme en section 4.2., notre explication de ces résultats repose sur la spécificité des données issues du e-commerce. Les commentaires utilisateurs sont rédigés dans un style très différent de celui de wikipedia et chaque corpus est si spécifique que les informations issues d'autres sources (même assez proches) n'apportent pas de bonus par rapport à l'apprentissage seul.

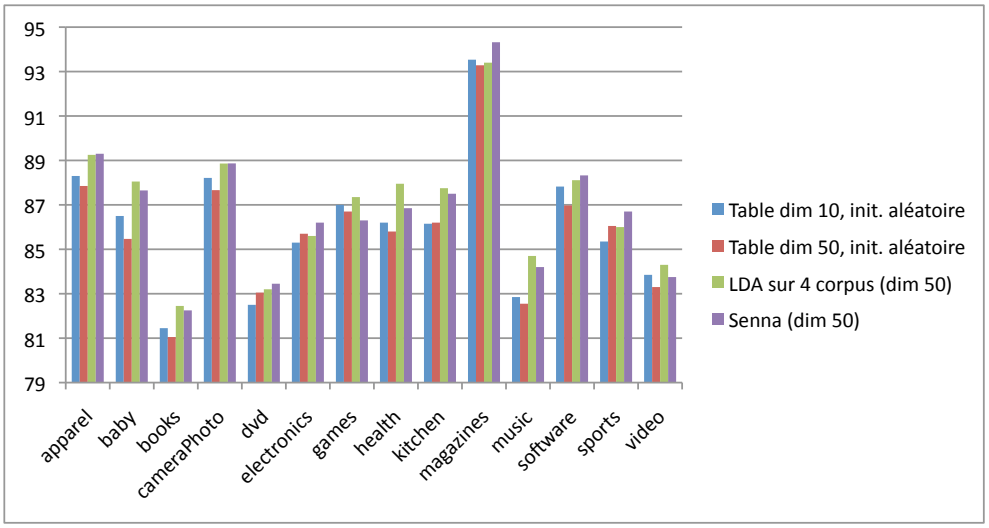

FIGURE 2: Taux de reconnaissance obtenus en mono-domaines, sur 14 corpus *Amazon* pour différentes initialisation du réseau de neurones à convolutions.

5.2. Détails des expériences mono-domaines

Nous présentons les expériences réalisées en mono-domaine sur les corpus *books, dvd, electronics* et *kitchen* dans le tableau 4. Ces quatre corpus permettent de bien rendre compte des différences entre les différents algorithmes : les expériences menées sur les autres corpus vont toujours dans le même sens. Le réseau à double convolution (RNC) est présenté en dernière colonne et ses performances sont très nettement au dessus des deux modèles de référence (SVM linéaire et réseau à convolution simple RNCS [5] de (Rafrafi *et al.*, 2011)). Afin de présenter des résultats comparables, toutes les expériences ont été réalisées avec les mêmes représentations numériques (unigrammes basés sur le même dictionnaire). Les performances sont les taux de reconnaissances calculés en validation croisée (5x). Les performances du RNC sont au niveau de l'état de l'art du domaine qui fait généralement appel à des descripteurs de plus haut niveau (N-grammes, traitements issus de la langue naturelle...).

Le SVM est clairement pénalisé par l'usage des unigrammes (les formes plus complexes étant plus efficaces pour capturer les informations de sentiments). Les performances du RNCS sont plus difficiles à expliquer : nous avons passer beaucoup de temps à régler les paramètres sans trouver de solution efficace. A l'inverse, le plus grand nombre de paramètres du RNC (à double convolution) a permis de débloquer la situation. Les performances sont particulièrement élevées pour des unigrammes.

Corpus *Amazon*	SVM	RNCS	RNC
Books	80.8	80.51	82.45
DVD	82.7	80.55	83.2
Electronics	82.2	82.1	85.6
Kitchen	83.5	81.86	87.75

TABLE 4: Performances obtenues en mono-domaines pour différents modèles (5x validations croisées). SVM : *Support Vector Machine* linéaires. RNCS : réseau de neurones à convolution simple. RNC : Réseau de neurones à convolutions.

5. L'architecture du RNCS est identique au RNC avec une couche en moins. L'activation est linéaire sur les couches cachées et sinusoïdale sur les couches de convolution et en sortie.

5.3. Expériences multi-domaines

Nous abordons maintenant les expériences en multi-domaines : le modèle est appris sur un corpus et testé sur un autre (protocole détaillé en section 3.). Les taux de reconnaissance sont comparés (entre SVM et RNC) sur la figure 3. L'écart de performance entre le RNC et le SVM est très net. En moyenne sur les 12 expériences multi-domaines (4 cibles x 3 domaines d'apprentissage), nous obtenons un taux de reconnaissance de 77.4% avec le RNC contre 74.9% avec le SVM. Au niveau de l'état de l'art, ces performances sont à comparer avec (Blitzer *et al.*, 2007) et (Pan *et al.*, 2010). Les premiers obtiennent une performance moyenne de 77.95% et les second 78.65% sur les mêmes données brutes. Cependant, les descripteurs et le protocole sont différents : ces deux articles utilisent un large corpus de données cibles non-étiquetées (entre 3685 (pour *dvd*) et 5945 nouveaux documents (*electronics*)) pour apprendre une fonction de transfert, ils utilisent également 50 documents cibles étiquetés pour améliorer leurs modèles. Les méthodes mises en oeuvre dans ces articles sont lourdes et n'aboutissent finalement qu'à des gains très minimes par rapport à notre RNC.

La comparaison entre le SVM et le RNC est particulièrement valorisante. Si le SVM était pénalisé par les unigrammes en mono-domaine, le raisonnement est différent en multi-domaine. Les représentations complexes posent des problèmes de sur-apprentissage et le multi-domaine requiert justement une plus grande régularité : les unigrammes sont plus adaptés à cette tâche. L'écart entre les modèles n'en est que plus significatif.

6. Conclusion

Dans cet article, nous avons décrit en détail le fonctionnement d'un réseau de neurones à double convolution sur le problème de classification de sentiments. Cette architecture est originale et permet théoriquement l'apprentissage d'une sémantique spécialisée à partir d'un corpus étiqueté.

Nous tirons plusieurs conclusions de ce travail. L'analyse qualitative des résultats est très difficile et le modèle fonctionne pour nous comme une boite noire. Au niveau quantitatif, les résultats sont beaucoup plus intéressants : la grande complexité du réseau permet d'apprendre efficacement la classification de sentiments (par rapport au réseau simple couche) et l'espace sémantique engendré permet de passer efficacement au formalisme multi-domaines.

Les perspectives autour de ce travail consistent principalement à enrichir

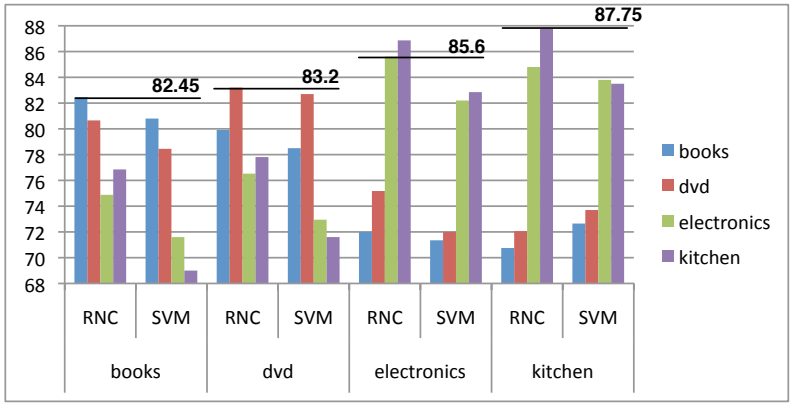

FIGURE 3: Taux de reconnaissance obtenus en multi-domaines. Le corpus de test (cible) est spécifié en abscisse, celui d'apprentissage est donné dans la légende. Les résultats des expériences mono-domaines RNC ont été reportés ici pour pouvoir juger graphiquement la perte liée au passage à une autre cible.

le codage de la table de référencement avec des négations et des informations sur la structure des phrases afin de dépasser le codage *naïf* en unigrammes. Nous cherchons aussi à travailler sur de nouvelles problématiques comme la reconnaissance des auteurs pour exploiter les capacités multi-classes des réseaux de neurones.

Remerciements

Ce travail est soutenu financièrement par la DGCIS (projet DOXA) et l'ANR (projet Fragrances ANR-08-CORD-008).

Références

BENGIO Y., DUCHARME R. & VINCENT P. (2000). A neural probabilistic language model. In *NIPS'00*.

BLITZER J., DREDZE M. & PEREIRA F. (2007). Biographies, Bollywood, boom-boxes and blenders : Domain adaptation for sentiment classification. In *ACL*.

BOTTOU L. & LE CUN Y. (1996). The backpropagation cookbook. In *NIPS workshop : Trick of the Trade*.

COLLOBERT R. & WESTON J. (2008). A unified architecture for natural language processing : Deep neural networks with multitask learning. In *International Conference on Machine Learning, ICML*.

DREDZE M., KULESZA A. & CRAMMER K. (2010). Multi-domain learning by confidence-weighted parameter combination. *Machine Learning Jour.*, **79**(1–2), 123–149.

FÉRAUD R. & CLÉROT F. (2002). A methodology to explain neural network classification. *Neural Networks*, **15**(2), 237–246.

GERRISH S. & BLEI D. (2011). Predicting legislative roll calls from text. In *ICML*, p. 489–496.

GLOROT X., BORDES A. & BENGIO Y. (2011). Domain adaptation for large-scale sentiment classification : A deep learning approach. In *ICML*.

HOFMANN T. (1999). Probabilistic latent semantic indexing. In *SIGIR*.

JINDAL N. & LIU B. (2007). Review spam detection. In *WWW*.

MARCOUX J. & SELOUANI S.-A. (2009). A hybrid subspace-connectionist data mining approach for sales forecasting in the video game industry. *Computer Science and Information Engineering*, **5**, 666–670.

PAN S., NI X., SUN J.-T., YANG Q. & CHEN Z. (2010). Cross-domain sentiment classification via spectral feature alignment. In *WWW*.

PANG B. & LEE L. (2004). A sentimental education : Sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL*.

PANG B. & LEE L. (2008). Opinion mining and sentiment analysis. *Information Retrieval*, **2**, 1–135.

PANG B., LEE L. & VAITHYANATHAN S. (2002). Thumbs up ? : sentiment classification using machine learning techniques. In *ACL-Empirical Methods in NLP*, volume 10, p. 79–86.

PARIOLLAUD F., DENHIÈRE G. & VERSTIGGEL J. (2002). Le traitement des expressions idiomatiques : Intérêt d'un corpus et de l'analyse sémantique latente. In *IPMU*, volume 3, p. 1481–1484.

RAFRAFI A., GUIGUE V. & GALLINARI P. (2011). Réseau de neurones profond et svm pour la classification de sentiments. In *CORIA*.

WHITEHEAD M. & YAEGER L. (2009). Building a general purpose cross-domain sentiment mining model. In *IEEE World Congress on Computer Science and Information Engineering*, p. 472–476.

YI J. & NIBLACK W. (2005). Sentiment mining in webfountain. In *ICDE*, p. 1073–1083.

ZHAI Z., LIU B., ZHANG L., XU H. & JIA P. (2011). Identifying evaluative opinions in online discussions. In *AAAI*.

ZHANG L. & LIU B. (2011). Identifying noun product features that imply opinions. In *ACL*.