RNN & modèle d'attention pour l'apprentissage de profils textuels personnalisés

Charles-Emmanuel Dias*, Clara Gainon de Forsan de Gabriac*, Vincent Guigue*, Patrick Gallinari *.

*Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, LIP6 , F-75005 Paris, France

RÉSUMÉ. *Nous nous intéressons dans cet article à la construction de profils issus à la fois des données d'interaction des utilisateurs (notes sur les produits) et des données textuelles associées (revues). L'enjeu est de s'éloigner des approches de factorisation matricielle pour mieux exploiter les données textuelles. Nous proposons de personnaliser une architecture de réseau de neurones hiérarchique dédiée à la classification de sentiments en apprenant des paramètres d'attention spécifiques pour les différents utilisateurs. Nous démontrons ensuite l'intérêt de ces paramètres pour établir des profils de recommandation pertinents. Non seulement les prédictions proposées par notre système dépassent les performances des systèmes de factorisation matricielle, mais le fait de travailler dans un espace latent textuel nous permet en plus de proposer des explications autour des recommandations pour sortir des boîtes noires habituelles du filtrage collaboratif.*

ABSTRACT. *In this article, we aim at learning relevant profiles from both rating and textual data. We attempt to make better use of review texts and seek to build a recommender system that does not fully rely on matrix factorization. In that sense, we propose to add some personalized attention parameters to a hierarchical neural network, which was first dedicated to sentiment analysis. We demonstrate that this modified attention is useful to build effective recommendation profiles: not only does it improve the global recommender system performance, but it enables us to provide suggestion explanations by exploiting the underlying learnt textual latent space. This latter point is a noteworthy way to overcome the classical black-box phenomenon in collaborative filtering approaches.*

MOTS-CLÉS : *Recommandation, Analyse de Sentiments, Réseaux de Neurones.*

KEYWORDS: *Recommender system, Sentiment Analysis, Neural Networks.*

1. Introduction

Afin d'aider les utilisateurs à accéder à des ressources en ligne toujours plus foisonnantes, des systèmes de recommandation ont été développés pour sélectionner et trier les contenus susceptibles de leur plaire. Ils permettent, par exemple, de proposer où dîner, de composer des newsletters, de fournir des conseils individualisés sur les produits à acheter ou, plus simplement, quels films regarder. Ces systèmes, qui cherchent à modéliser les préférences des utilisateurs et les attributs des produits, ont pour but de proposer un véritable accès personnalisé à l'information.

Pour atteindre cet objectif, il faut que les profils appris puissent fournir des recommandations justes. Mais, il est tout aussi important d'expliquer en quoi celles-ci sont appropriées (Tintarev et Masthoff, 2007). Les algorithmes de recommandation ont franchi un premier palier d'efficacité (et de popularité) avec le challenge Netflix (Bennett *et al.*, 2007) qui a consacré le filtrage collaboratif par factorisation matricielle (Koren *et al.*, 2009). Ces dernières années, ces algorithmes sont devenus de plus en plus performants grâce à la prise en compte d'un nombre croissant de facteurs comme le temps (Koren, 2010 ; McAuley et Leskovec, 2013a), les liens sociaux (Guy, 2015) ou encore le texte (McAuley et Leskovec, 2013b).

Pourtant, ces améliorations quantitatives n'ont que peu amélioré le côté explicatif de la recommandation et les méthodes récentes de filtrage collaboratif sont encore souvent qualifiées de boites noires (Ben-Elazar et Koenigstein, 2014). Ce manque d'interprétabilité au profit de la performance résulte directement des choix de modélisation des profils. En effet, depuis la compétition Netflix en 2006, les profils utilisateurs (et produits) sont issus d'un processus de factorisation matricielle d'un ensemble de notes qui structure un espace latent abstrait où les dimensions sont difficilement interprétables. Pourtant, dans la pratique, les notes sont souvent accompagnées d'avis écrits. Ces textes ont déjà été exploités avec succès pour améliorer la qualité de prédiction (McAuley et Leskovec, 2013a ; Ling *et al.*, 2014). Des travaux ont aussi entrepris de re-projeter la recommandation dans l'espace des revues afin d'obtenir des explications et ainsi une recommandation interprétable (Poussevin *et al.*, 2015 ; Almahairi *et al.*, 2015). Les données textuelles représentent également un moyen élégant de faire face au démarrage à froid, si problématique dans les approches de filtrage collaboratif (Dias *et al.*, 2017).

Dans cet article, notre objectif est de présenter une nouvelle architecture à base de réseaux de neurones profonds capable de définir des profils utilisateurs et items très pertinents à partir des notes et des textes des revues. Nous repartons des travaux récents en analyse de sentiments qui, en s'appuyant sur un modèle d'attention hiérarchique, permettent de détecter automatiquement les mots et phrases d'intérêt (Yang *et al.*, 2016 ; Chen *et al.*, 2016). L'hypothèse originale de notre approche est la suivante : nous considérons le classifieur de sentiments comme un modèle de lecture personnalisé et nous pensons que l'attention qu'un utilisateur prête aux mots et aux phrases permet de définir son profil de manière originale et pertinente. Notre système est parfaitement intégré, il est donc possible de le voir comme un système de classi-

fication de sentiments avec des paramètres de pondération personnalisés sur les mots et les phrases ou à l'inverse, tel un système de recommandation utilisant le texte et l'attention comme une manière de régulariser l'apprentissage, tout en permettant d'interpréter les propositions émises. De plus, nous estimons que les profils collaboratifs encodent principalement les biais utilisateurs et items alors que le texte nous permet de mieux comprendre les aspects importants d'un produit pour l'utilisateur, en se focalisant sur les exemples où les biais susmentionnés sont insuffisants pour reconstruire la note observée.

Cet article est organisé de la façon suivante : tout d'abord, nous exposons le contexte bibliographique de nos travaux (section 2) avant d'expliciter le modèle associé à nos travaux (section 3). Enfin, nous démontrons l'intérêt de notre approche sur les plans quantitatif et qualitatif (section 4).

2. Cadre bibliographique

Notre objectif premier est ici d'utiliser le texte comme support pour modéliser les utilisateurs et les produits dans un espace vectoriel interprétable. Pour ce faire, nous exploiterons un réseau de neurones récurrents dédié à l'analyse de sentiments. La première partie de cette bibliographie leur est donc dévolue. Nous montrerons ensuite comment tirer parti des paramètres d'attention pour construire des profils personnalisés, ce qui nous conduit à la seconde section sur les systèmes de recommandation.

2.1. *Réseaux de Neurones Récurrents et Analyse de Sentiments*

Les réseaux de neurones récurrents (RNN, *Recurrent Neural Network*) présentent l'avantage de traiter naturellement des séquences de longueurs variables, ce qui est bien adapté à la modélisation des données textuelles. En effet, contrairement aux réseaux de neurones classiques, le calcul d'une sortie y_t à un l'instant t prend en compte à la fois l'entrée actuelle $\mathbf{x}_t \in \mathbb{R}^d$ et une représentation des entrées précédentes $\mathbf{h}_{t-1} \in \mathbb{R}^z$ (Elman, 1990) :

$$\begin{cases} \mathbf{h}_t = \sigma_h(W_h \mathbf{x}_t + U_h \mathbf{h}_{t-1} + b_h) \\ y_t = \sigma_y(W_y \mathbf{h}_t + b_y) \end{cases} \quad [1]$$

C'est cet état caché $\mathbf{h}_t$, qui est une composition non linéaire de l'ensemble des entrées $\mathbf{x}_1, .., \mathbf{x}_t$ modulée par une fonction d'activation σ_h, qui permet de conserver une mémoire de la séquence ; il s'agit donc d'une représentation latente de l'ensemble du passé. Néanmoins, cette version classique du réseau de neurones récurrent souffre du problème de disparition du gradient ; en effet, celui-ci décroit de manière exponentielle du fait des nombreuses multiplications lors de la rétro-propagation temporelle. Cela rend difficile l'apprentissage sur de longues séquences. C'est dans ce contexte qu'ont été développées les cellules "à portes" dont les plus connues sont le LSTM (Long

Short Term Memory) (Hochreiter et Schmidhuber, 1997) et sa variante simplifiée, la GRU (Gated Recurrent Unit) (Cho *et al.*, 2014).

Plus récemment, le concept d'attention, issu de la traduction automatique (Bahdanau *et al.*, 2014), a apporté des gains significatifs dans plusieurs applications en TALN, dont la classification de sentiments (Yang *et al.*, 2016 ; Parikh *et al.*, 2016). Il est la pierre angulaire d'un nouveau paradigme qui permet d'atteindre l'état de l'art : *embed, encode, attend, respond* (Honnibal, 2016) ; les paramètres d'attention pondèrent les informations à prendre en compte dans la séquence et permettent de se focaliser sur les éléments discriminants. L'idée générale est la suivante : soit une séquence d'éléments $S = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$. Sa représentation $\mathbf{e}_s$ est en fait la somme de ses éléments s_i pondérée par des coefficients α_i (eq. 2) :

$$\mathbf{e}_s = \sum_{i=1}^{n} \alpha_i \mathbf{s}_i \qquad [2]$$

Il existe différentes manières d'obtenir ces coefficients α_i, mais ils doivent généralement résulter d'une interaction entre deux vecteurs. Dans (Yang *et al.*, 2016) un vecteur "d'attention" $\mathbf{a}$ est appliqué sur les $\mathbf{s}_i$ en utilisant le produit scalaire :

$$\alpha_i = \frac{\exp(\mathbf{a}^\intercal \mathbf{s}_i)}{\sum_i \exp(\mathbf{a}^\intercal \mathbf{s}_i)} \qquad [3]$$

Ce coefficient α_i est alors interprété comme un score d'importance du mot ou de la phrase pour la tâche considérée. La phase d'apprentissage permet d'identifier le vecteur d'attention $\mathbf{a}$ qui conduit à une pondération optimale des termes en jeu.

Notre approche est originale car elle repose sur un mécanisme d'attention personnalisée en fonction de l'utilisateur ; le vecteur d'attention ainsi appris servant de base au profil utilisé en recommandation.

2.2. *Filtrage collaboratif pour la recommandation*

En recommandation, le filtrage collaboratif est une méthode de prédiction personnalisée qui se base sur l'historique des notes données par les utilisateurs. L'hypothèse sous-jacente est que les personnes ayant exprimé des intérêts communs pour certaines choses dans le passé partagent probablement des affinités sur d'autres produits. La principale implémentation du filtrage collaboratif est la factorisation matricielle. Cette approche, en particulier sa variante positive, a fait ses preuves lors du concours Netflix (Bennett *et al.*, 2007) grâce aux travaux de référence de (Koren *et al.*, 2009). Elle permet d'extraire des profils latents continus d'utilisateurs $P = \{\mathbf{p}_u\}_{u=1,\ldots,N_u} \in \mathbb{R}^{N_u \times Z}$ et de produits $Q = \{\mathbf{q}_i\}_{i=1,\ldots,N_i} \in \mathbb{R}^{N_i \times Z}$ directement à partir de la matrice de notes, de façon à ce que la note prédite $\hat{r_{ui}}$ résulte du produit scalaire entre les profils du produit i et de l'utilisateur u :

$$\hat{r_{ui}} \approx \mathbf{q}_i^T \mathbf{p}_u, \qquad \mathbf{q}_i, \mathbf{p}_u \in \mathbb{R}^{Z \times Z} \qquad [4]$$

À la manière des méthodes de segmentation thématique, chaque dimension latente des profils est supposée correspondre à un aspect concret ; la présence du même aspect dans deux profils provoque une correspondance locale faisant monter la note.

En recommandation, les notes sont fortement biaisées. Certaines personnes ont tendance à toujours mettre de bonnes/mauvaises notes et certains produits ont tendance à être sur/sous notés. Classiquement, ces biais sont modélisés en ajoutant une moyenne globale μ, un biais produit b_i et un biais utilisateur b_u à (eq. 4) ; la prédiction prend alors la forme suivante :

$$\hat{r_{ui}} = \mu + b_i + b_u + \mathbf{q}_i^T \mathbf{p}_u \quad [5]$$

et les profils sont appris de manière régularisée pour éviter le sur-apprentissage :

$$q^\star, p^\star, b^\star = \arg\min_{q,p,b} \sum_{(u,i)} (r_{ui} - \mu + b_i + b_u + \mathbf{q}_i^T \mathbf{p}_u)^2 + \lambda(||P||_F^2 + ||Q||_F^2 + b_u^2 + b_i^2) \quad [6]$$

Dans cette formulation, les profils latents capturent uniquement la déviation à la moyenne, ce qui permet de proposer une estimation même lorsqu'un profil est manquant. De nombreuses variantes ont été proposées pour exploiter différents facteurs de contexte comme le temps (Koren, 2010 ; McAuley et Leskovec, 2013a), les liens sociaux (Guy, 2015) ou le texte. Ces facteurs apportent de l'information dans les profils et permettent de les contraindre, introduisant une meilleure régularisation, et donc, de meilleures prédictions.

Nos travaux sont philosophiquement proches de ceux qui contraignent les facteurs latents à l'aide du texte (McAuley et Leskovec, 2013b ; Almahairi *et al.*, 2015). Néanmoins, ils s'en distinguent sur deux plans. Tout d'abord, nous proposons d'utiliser un algorithme d'apprentissage récent, basé sur les RNN (Réseaux de neurones récurrents), modélisant hiérarchiquement les mots et les phrases dans les revues. Ensuite, l'alignement entre les profils utilisateurs et le modèle d'attention sur le texte est axé sur l'interprétabilité de notre modèle : le but est de comprendre comment un utilisateur sélectionne les informations dans les revues qu'il lit pour mieux lui expliquer les propositions du système en inférence.

3. Réseau de neurones récurrent hiérarchique attentif pour la recommandation

Notre modèle, baptisé Réseaux de Neurones Récurrent Hiérarchique Attentif pour la Recommandation (RHAR), est composé de deux modules parallèles. Le premier est un perceptron multicouche classique –à deux couches– ; il prend en entrée les profils de l'utilisateur et du produit pour estimer une note. En soi, il joue le même rôle que la factorisation matricielle dans le filtrage collaboratif. Le second est un réseau de neurones récurrent hiérarchique composé de deux modules récurrents bidirectionnels-attentifs (**RBA**) qui permettent d'encoder les avis en analysant successivement les

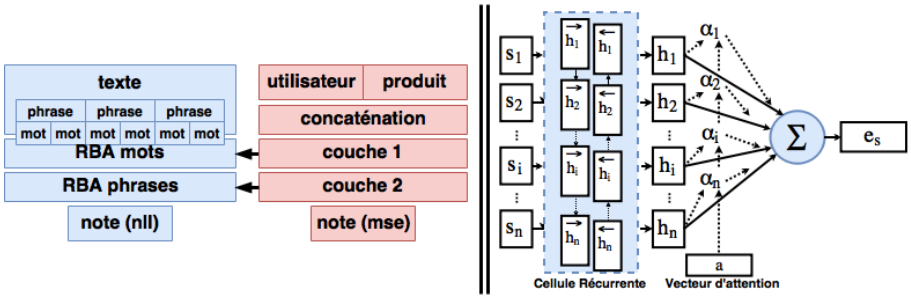

Figure 1 : **(A gauche) - Vue générale du modèle.**
(A droite) - Détail d'un module récurrent bidirectionnel-attentif (RBA). Les $\mathbf{s}_i$ désignent soit les mots, soit les phrases selon le niveau hiérarchique considéré.

mots d'une phrase puis les phrases de la revue. Son rôle est d'analyser le texte d'un utilisateur sur un produit et d'en prédire sa polarité. Ce dernier réseau est dépendant du premier car ses deux modules **RBA** sont liés avec les couches cachées du premier : l'attention que porte un utilisateur aux mots et aux phrases permet de définir son profil pour les applications de recommandation (fig. 1 - gauche).
Dans un premier temps, nous explicitons la construction d'un **RBA**. Ensuite, nous détaillons formellement l'architecture bicéphale du réseau global.

3.1. *Module récurrent bidirectionnel-attentif (RBA)*

Ce module est le principal bloc du modèle de prédiction de sentiments. Il prend en entrée une séquence et un vecteur d'attention –pondérant indirectement les éléments de la séquence– pour retourner une représentation de celle-ci (fig. 1 - droite). L'originalité de notre approche, à ce niveau, réside dans la personnalisation du calcul des attentions.

Formellement, soit une séquence $seq = \{\mathbf{s}_1, \ldots, \mathbf{s}_i, \ldots, \mathbf{s}_n\}$ composée de n éléments. Pour obtenir sa représentation $\mathbf{e}_s$, la séquence est d'abord passée par un réseau de neurones récurrent bi-directionnel $RF = \{\overrightarrow{RF}, \overleftarrow{RF}\}$ qui, en parcourant la séquence dans les deux sens, encode le contenu intra-séquence. Les sorties du réseau récurrent sont concaténées à chaque pas de temps pour obtenir l'ensemble des représentations cachées $\mathbf{h}_i$ (eq. 7). Ici, nous utilisons un LSTM (Hochreiter et Schmidhuber, 1997) comme cellule récurrente.

$$\mathbf{h}_i = [\overrightarrow{\mathbf{h}_i}; \overleftarrow{\mathbf{h}_i}], \qquad \overrightarrow{\mathbf{h}_i} = \overrightarrow{RF}(\mathbf{s}_i), \qquad \overleftarrow{\mathbf{h}_i} = \overleftarrow{RF}(\mathbf{s}_i), \qquad [7]$$

Ensuite, chaque élément $\mathbf{h}_i$ est projeté de manière non linéaire dans l'espace d'attention afin de calculer son affinité α_i avec un vecteur d'attention **a** selon la formule suivante :

$$\mathbf{e}_s = \sum_{i=1}^{n} \alpha_i \mathbf{h}_i, \qquad \mathbf{t}_i = \tanh(W^{tu}\mathbf{h}_i + b_u), \qquad \alpha_i = \frac{\exp(\mathbf{a}^\intercal \mathbf{t}_i)}{\sum_i \exp(\mathbf{a}^\intercal \mathbf{t}_i)} \quad [8]$$

Ces affinités α sont normalisées à l'aide d'une fonction *softmax* afin qu'elles somment à 1. Le vecteur d'attention **a** –qui peut être vu comme une représentation moyenne de ce qui est important– apprend automatiquement les éléments discriminants en fonction de la tâche.

Afin de personnaliser l'attention, nous définissons le vecteur **a** comme une combinaison d'un vecteur d'attention global et d'un vecteur d'attention personnalisé issu du réseau de neurones dédié à la prédiction de note. Formellement, le vecteur **a** final est défini comme :

$$\mathbf{a} = \tanh(\mathbf{a}_{u,i} + \mathbf{a}_g), \qquad \mathbf{a}_{u,i} = W^\ell \ell \quad [9]$$

où $\mathbf{a}_{u,i}$ est issue du réseau de recommandation et correspond au profil fusionné de l'item et de l'utilisateur ciblé ℓ (cf section suivante). La matrice W^ℓ permet de passer d'un réseau à l'autre de manière efficace.

Le vecteur global a pour but d'encoder les adjectifs généraux afférents à la prédiction de polarité (i.e *bad, great, awesome*) alors que $\mathbf{a}_{u,i}$ permet au réseau de se focaliser sur des mots plus précis en lien avec les affinités d'une personne par rapport à un produit (i.e *nom de marque, spécificités, attributs*). Des exemples de mots sélectionnés par l'attention sont présentés figure 4.

3.2. *Architecture générale du réseau*

Comme le montrent les figures 1 et 2, notre architecture (RHAR) est composée de deux réseaux de neurones distincts. Un perceptron multicouche pour la recommandation –prenant en entrée des profils et prédisant une note– et un réseau de neurones hiérarchique récurrent pour l'analyse de sentiments –prenant du texte et estimant sa polarité–. Nous décrivons dans un premier temps le perceptron multicouche avant de revenir sur la hiérarchie du module d'analyse de sentiments.

3.2.1. *Perceptron multicouche pour la recommandation*

La première partie du réseau est classique, il s'agit d'un perceptron multicouche composé de deux couches cachées et d'une couche de régression. Il prend en entrée un couple de représentations $(utilisateur, item)$ qu'il concatène et transforme séquentiellement en deux sous-représentations ℓ_w et ℓ_s. Formellement, soit $\ell = [u; i]$ la concaténation d'une paire de représentations utilisateur et produit. La note prédite $\hat{r}_{ui}$

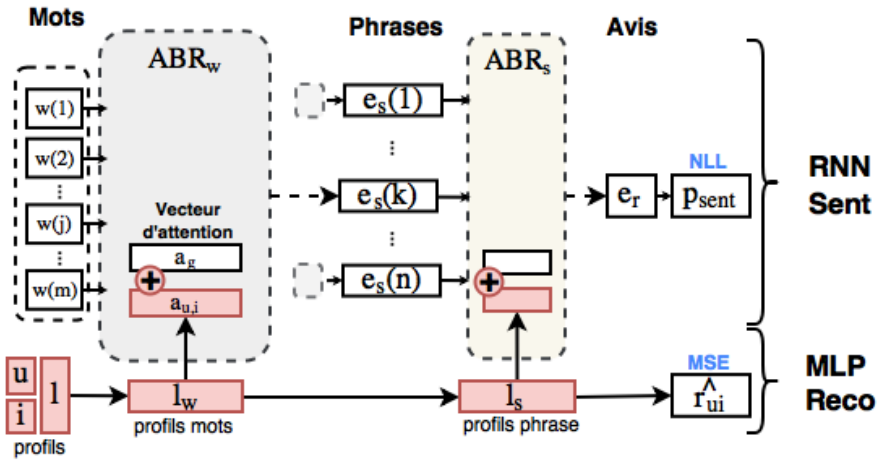

Figure 2 : **Vue détaillée du modèle pour une entrée de n phrases de m mots : (Haut) RNN sentiment, composé de deux RBA, un pour les mots (RBA_w) et un pour les phrases (RBA_s). – (Bas) Perceptron multicouche pour la recommandation.**

est simplement obtenue par régression, en transformant successivement ℓ en ℓ_w, ℓ_w en ℓ_s et finalement ℓ_s en $\hat{r}_{ui}$.

$$\ell_w = \tanh(W^{\ell w}\ell + b_\ell), \quad \ell_s = \tanh(W^{ws}\ell_w + b_w), \quad \hat{r}_{ui} = \tanh(W^{sr}\ell_s + b_s) \qquad [10]$$

L'objectif de ce réseau est de minimiser l'erreur quadratique moyenne sur la prédiction de note.

Le choix d'un perceptron à deux couches cachées permet d'extraire des caractéristiques de bas niveau ℓ_w et de haut niveau ℓ_s. Nous utiliserons ℓ_w pour encoder l'attention sur les mots du couple (utilisateur u, produit i) tandis que ℓ_s encodera l'attention sur les phrases. Le passage d'un réseau à l'autre est effectué en utilisant la projection décrite en eq. 9.

3.2.2. *Réseau hiérarchique d'analyse de sentiments*

Le second réseau prend uniquement les avis en entrée et cherche à en prédire la polarité. Il est composé de deux RBA, utilisés l'un après l'autre afin d'encoder hiérarchiquement les avis –au niveau des mots puis des phrases– et d'une couche de classification. A l'issue du double encodage, la revue est représentée par un vecteur $\mathbf{e}_r$:

$$RBA_w : (\{w\}, \ell_w) \mapsto \mathbf{e}_s \qquad RBA_s : (\{\mathbf{e}_s\}, \ell_s) \mapsto \mathbf{e}_r \qquad [11]$$

Les mots w des phrases sont encodés en $\mathbf{e}_s$ puis les phrases sont agrégées en $\mathbf{e}_r$. Enfin, cette représentation e_r passe par une couche de classification W^{pred}. Nous utilisons un *softmax* pour obtenir une distribution dans l'espace de notes possibles :

$$p_{sent} = \text{softmax}((W^{pred} e_r) + b_p) \quad [12]$$

L'objectif de ce réseau est de minimiser la log-vraisemblance négative (NLL) sur la prédiction de note, ce qui revient à essayer de prédire la polarité de la revue.

3.2.3. *Hyper-paramètres, Objectif et Entraînement*

Nos modèles sont implémentés à l'aide de *Pytorch*[1]. Nous minimisons conjointement deux fonctions objectifs : l'erreur quadratique moyenne (pour le perceptron multicouche de recommandation) et la log-vraisemblance négative (pour le réseau de classification de sentiments). L'intégralité des paramètres du réseau sont entraînés par descente de gradient, en utilisant l'algorithme d'optimisation Adam (Kingma et Ba, 2014). Les hyper-paramètres sont les mêmes pour toutes les expériences qui suivent. Les couches cachées sont de la même taille que les embeddings des mots et des phrases : 200 pour les RBAs et les mots et 40 pour le perceptron multicouche (20 pour les utilisateurs, 20 pour les produits).

4. Expériences

Dans un premier temps nous présentons les données utilisées pour évaluer notre double modèle. La première série d'évaluations proposée est d'ordre quantitative, il s'agit d'évaluer les performances de nos modèles en analyse de sentiments (classification de notes) et en recommandation (écart de prédiction). Ensuite, afin d'évaluer qualitativement le modèle, nous proposons d'observer les représentations apprises par celui-ci.

4.1. *Données et pré-traitements*

Pour nos expériences, nous nous appuyons sur des avis consommateurs extraits d'Amazon (McAuley *et al.*, 2015). Nous avons retenu cinq bases d'avis issues de différents domaines thématiques dont les statistiques sont détaillées dans le tableau 1. Le problème de démarrage à froid n'est pas abordé dans cet article et les bases d'avis originales sont expurgées : seuls les avis des utilisateurs et des produits avec au minimum cinq avis sont conservés.

Pour séparer les avis en listes de phrases et les phrases en listes de mots, nous utilisons la librairie de TALN *spacy*[2]. Les mots sont encodés en représentations apprises lors de l'entraînement. Seuls les 10.000 mots les plus utilisés sont conservés,

1. `http://pytorch.org/`
2. `https://spacy.io/`

Dataset	#Avis	#Util.	#Prod.	p(1)	p(2)	p(3)	p(4)	p(5)
Instant Video	37 126	5130	1685	4,6	5,1	11,3	22,7	56,3
Digital Music	64 706	5541	3568	4,3	4,7	10,5	25,6	55,0
Video Games	231 780	24 303	10 672	6,4	5,9	12,2	23,6	51,9
Clothes S.J.	278 677	39 387	23 033	4,0	5,5	10,9	20,9	58,6
Movies	1 697 533	123 960	50 052	6,1	6,0	11,9	22,6	53,4

Tableau 1 : Statistiques des différentes notes (en pourcentage) pour les bases de données Amazon utilisées.

les autres sont remplacés par une représentation dédiée aux mots inconnus. Enfin, les données sont séparées en cinq ensembles égaux pour la procédure de validation croisée. Pour chaque campagne d'évaluation, nous en utilisons quatre (80% des données) pour l'entraînement, et un –divisé en deux– pour la validation (10%) et l'évaluation (10%).

4.2. *Évaluation en analyse de sentiments :*

Notre première tâche est l'analyse de sentiments, qui consiste à prédire la polarité d'un texte. Nous proposons de comparer les performances de notre modèle par rapport à trois modèles de référence récents utilisant également des représentations de texte :

– **FastText** (Joulin *et al.*, 2017). Issue des travaux autour de *word2vec*, l'idée est d'apprendre des représentations de mots spécifiques à la tâche de classification. Ces représentations sont moyennées (sur la revue) et classifiées par régression logistique.

– **HAN** *Hierarchical Attention Networks for Document Classification* (Yang *et al.*, 2016) : Ce modèle est équivalent à notre réseau de neurones hiérarchique récurrent, sans l'attention personnalisée.

– **NSUPA** *Neural Sentiment Classification with User & Product Attention* (Chen *et al.*, 2016) : Ce modèle est une évolution de **HAN** visant à prendre en compte les biais utilisateurs/items –comme dans notre approche–. Pour ce faire, les auteurs proposent de projeter le texte dans un espace d'attention paramétré par des représentations utilisateur et item. Dans leur formulation, cela revient à remplacer t_i dans l'équation 8 par $\mathbf{t}_i = \tanh(W^{tu}[h_i; i; u] + b_u)$. Là où (Chen *et al.*, 2016) proposent d'injecter les paramètres de profils directement dans le modèle de classification de sentiments, nous proposons une approche plus souple, reposant sur deux modèles indépendants reliés par les paramètres W^ℓ (eq. 9).

– **SVM** *Support Vector Machines*. Les SVM ont donné des résultats très intéressants en classification de sentiments dans un contexte binaire (fusion des revues négatives d'une part, des revues positives d'autre part et élimination des revues ambiguës –3 étoiles–) (Pang *et al.*, 2008). Les articles précédents (Yang *et al.*, 2016 ; Chen *et al.*, 2016) rapportent de mauvaises performances dans le cadre d'une classification

fine sur les cinq étoiles. Nous n'avons effectivement pas obtenu de résultats satisfaisants dans ce cadre, quel que soient les pré-traitements considérés. Pour cette raison, les résultats des SVM ne sont pas présents dans le tableau de résultat suivant.

Nous évaluons les capacités d'analyse de sentiments des modèles en taux de bonne classification. Les résultats sont reportés dans le tableau 2. Notre premier point de référence –FastText– est largement dépassé par les modèles hiérarchiques qui prennent en compte la structure des phrases. La richesse de l'espace de représentation des mots et la pertinence de cet espace pour l'analyse d'opinion ne suffisent pas à compenser la perte de structure du document et la faiblesse de la fonction d'agrégation des mots (une moyenne simple).

La comparaison des trois modèles hiérarchiques montre l'importance de modéliser les biais utilisateurs/produits pour passer un palier de performances : ces biais sont présents dans NSUPA et dans notre approche (RHAR) mais pas dans HAN. Entre notre proposition et NSUPA, les performances sont très proches. Aussi, l'enjeu de notre formulation était d'atteindre les performances de l'état de l'art en classification de sentiments et de mieux transférer les informations pertinentes vers les profils de recommandation.

	FastText	HAN	NSUPA	RHAR
Instant Video	62.60	64.50	65.88	**66.60**
Digital Music	63.58	68.03	**70.08**	68.80
Video Games	62.51	67.67	68.60	**69.11**
CSJ	67.83	71.96	**71.99**	71.49
Movies	64.56	68.95	71.20	**71.62**

Tableau 2 : Taux de bonne classification en analyse de sentiments - les valeurs sont des moyennes sur 5 splits (en %)

4.3. *Évaluation en recommandation :*

Le second objectif de notre approche est la prédiction de notes, une tâche classique en filtrage collaboratif. Nous avons opté pour une référence standard du domaine : la factorisation matricielle (FM), qui infère les notes uniquement à partir des profils utilisateurs et items appris sur les notes, sans tenir compte des avis écrits. A l'inverse, la seconde référence, TransNet (Catherine et Cohen, 2017), ne prend en compte que le texte : TransNet adopte une approche basée sur le *matching* entre les textes du passé de l'utilisateur et les textes associés à l'item cible, à la manière ce qui avait été proposé dans (Dias *et al.*, 2016). La note $\hat{r}_{ui}$ est directement prédite à partir des profils textuels existants, sans passer par une factorisation des observations. Comme habituellement en recommandation, nous mettons ces performances en perspective par rapport aux modèles triviaux prédisant respectivement la moyenne générale du jeu de données (μ) et la moyenne corrigée des biais utilisateur et item (*w/offset*). Les résultats

Dataset (#reviews)	Mean (μ)	w/offset	FM	TransNet	RHAR
Instant Video (37.126)	1.25	1.137	1.024	1.526	**0.937**
Digital Music (64,706)	1.19	0.965	0.903	1.522	**0.838**
Video Games (231,780)	1.45	1.281	1.267	1.313	**1.076**
CSJ (278,677)	1.215	1.323	1.365	1.285	**1.081**
Movie (1,697,533)	1.436	1.148	1.118	1.359	**1.058**

Tableau 3 : Erreur quadratique moyenne en prédiction de notes. Les valeurs de références sont la moyenne globale μ, le biais global (eq.5), une factorisation matricielle et le modèle Transnet. Les valeurs présentées correspondent à l'erreur quadratique moyenne sur les cinq ensembles.

sont présentés dans le tableau 3. Nous utilisons l'erreur quadratique moyenne comme mesure d'évaluation [3].

En terme de prédiction de notes, notre modèle est systématiquement meilleur que les modèles de références. Le modèle TransNet obtient des résultats décevants puisque quasi-systématiquement au dessus de la moyenne –sauf sur le dataset le plus fourni– : ne se basant que sur le texte il nécessite beaucoup de données pour être compétitif. Notre modèle est très largement supérieur à la factorisation matricielle, cela démontre la capacité de notre architecture à extraire des informations de profil pertinentes à partir des données textuelles.

4.4. *Visualisation de l'attention*

Un des avantages majeurs de notre modèle est la possibilité d'utiliser les modules d'attention pour l'introspection dans le cadre de la tâche de classification de sentiments. En effet, les vecteurs d'attention permettent d'isoler les éléments discriminants (mots et phrases) dans le corpus. Nous avons procédé de la manière suivante : l'ensemble des vecteurs d'attentions du module RBA_w de l'intégralité des n phrases d'évaluation est récupéré : $att_w = \{\alpha^{\mathbf{0}}, ..., \alpha^{\mathbf{n}}\}$. L'indice de la valeur maximum de chacun de ces vecteurs indique quel mot représente le plus d'intérêt dans chaque phrase ; ces mots sont donc les plus discriminants. Pour le corpus *Video Games*, ces mots issus de l'introspection du modèle sont représentés sous la forme d'un nuage de mots en figure 3. Globalement, le nuage fait apparaitre des mots de sentiments (souvent très positifs, en lien avec la distribution déséquilibrée des données présentée en tableau 1). Quelques mots liés au domaine (e.g. *game*, *playing*) apparaissent également.

3. Nos expériences intégraient initialement le modèle hybride de factorisation thématique & profiling de (McAuley et Leskovec, 2013b). Cependant, le code fourni par les auteurs donne systématiquement des résultats inférieurs à la factorisation matricielle standard. Pour cette raison, les résultats issus de cette approche ne sont pas présentés dans le tableau ci-dessous.

Figure 3 : Nuage des mots les plus discriminants du jeu de données *Video Games*. La taille est proportionnelle au nombre de fois où le mot est le plus important dans une phrase du corpus.

L'exercice d'introspection devient plus intéressant lorsque nous considérons séparément les deux composantes de l'attention $\mathbf{a}_g$ –terme général– et $\mathbf{a}_{u,i}$ –attention personnalisée– (cf eq. 9). Pour l'attention générale, comme dans l'expérience précédente, les mots extraits sont principalement des mots polarisants génériques comme *good, worst, love*(fig.4 - gauche). La figure 4 (droite) illustre au contraire les mots mis en avant par la personnalisation. En considérant toute la population, nous avons mis en évidence les mots revenant le plus souvent : la personnalisation permet au modèle de se focaliser sur des mots bien plus spécifiques que ceux issus de l'attention généraliste ; ces mots s'apparentent à des attributs dans le domaine du jeu vidéo. Nous observons des entités caractéristiques comme *Wolfenstein, Lara (Croft), Zelda* ou des éditeurs comme *Sony, Blizzard, Capcom*. Ce sont ces informations qui, intégrées aux profils des utilisateurs et des produits, nous ont permis de franchir un palier de performances sur la tâche de recommandation.

4.5. *Explications des suggestions par l'attention*

Un des enjeux qui nous pousse à traiter la recommandation dans un espace textuel, au-delà de l'amélioration dans la prédiction des notes, réside dans la capacité d'expliquer automatiquement à l'utilisateur du système les suggestions qui lui sont proposées. Notre modèle d'analyse de sentiments personnalisé doit se comporter comme un filtre à même de remplacer l'utilisateur dans la lecture des revues d'un produit : nous souhaitons isoler automatiquement les phrases et les mots qui sont susceptibles d'attirer son attention.

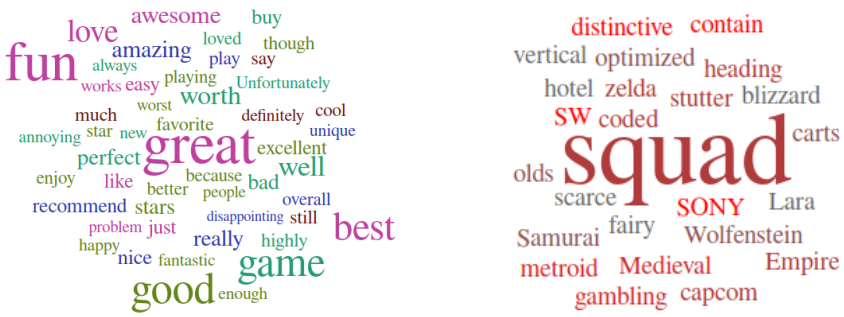

Figure 4 : Nuages de mots : **(à gauche)** des mots discriminants pour le modèle en utilisant seulement la composante généraliste de l'attention dans l'équation (eq. 9) – **(à droite)** des mots uniquement discriminants au sens de la composante personnalisée de l'attention dans l'équation (eq. 9)

Pour étudier cette possibilité, nous avons fait fonctionner notre architecture sur un couple utilisateur produit (figure 5) : la figure de droite rassemble toutes les sorties de notre modèle bicéphale, une prédiction de note, une extraction des mots importants, et enfin grâce au module d'attention sur les phrases, une extraction des phrases importantes pour l'utilisateur parmi les revues sur l'item cible. Les phrases présentées sont ordonnées par rapport aux scores d'attention (en bleu). La figure de gauche montre ce que l'utilisateur a effectivement écrit sur cet item[4].

Les systèmes de recommandation commerciaux commencent à intégrer des mécanismes d'explication pour sortir de la logique de boite noire. Ces explications sont généralement centrées sur l'item cible. Nous démontrons ici la possibilité de personnaliser les explications par rapport à chacun des utilisateurs du système.

5. Discussion

Dans cet article, nous nous sommes intéressés à la modélisation des utilisateurs et des produits via les corpus d'avis en ligne. Notre objectif était de mieux intégrer le texte dans le processus de recommandation. En nous basant sur des travaux récents en analyse de sentiments nous avons présenté une nouvelle architecture composée de deux réseaux de neurones opérant en parallèle et reliés par un mécanisme d'attention. Nous avons montré qu'un tel modèle permet d'améliorer les performances en recommandation –précision des notes prédites– tout en offrant une manière élégante d'expliquer les suggestions du système.

4. Les attentions sur les phrases écrites par l'utilisateur lui-même sont plus hautes que sur les phrases des autres auteurs, ce qui semble logique.

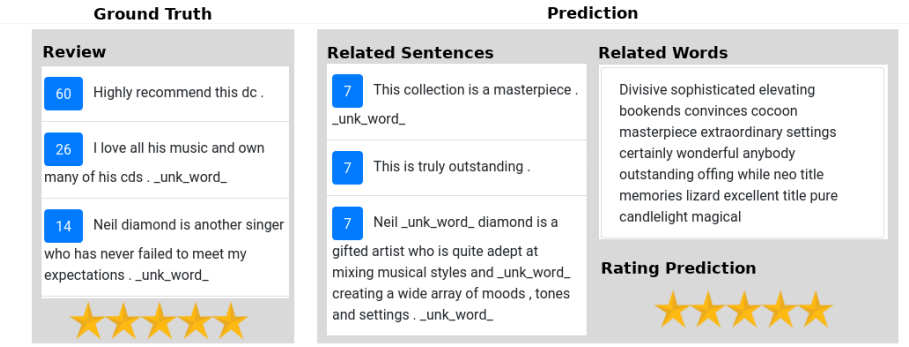

Figure 5 : Exemple d'analyse de l'attention pour l'explication de la recommandation. A gauche, la revue écrite par l'utilisateur et sa note. A droite, les différentes sorties de notre système : la note prédite, les phrases que nous jugeons utiles pour l'utilisateur –extraites des revues sur l'item ciblé–, les mots clés associés à la cible.

Remerciements

Ce travail a été réalisé en partie avec le soutien du FUI-BIND

6. Bibliographie

Almahairi A., Kastner K., Cho K., Courville A., « Learning distributed representations from reviews for collaborative filtering », *Proceedings of the 9th ACM Conference on Recommender Systems*, ACM, p. 147-154, 2015.

Bahdanau D., Cho K., Bengio Y., « Neural machine translation by jointly learning to align and translate », *arXiv preprint arXiv :1409.0473*, 2014.

Ben-Elazar S., Koenigstein N., « A Hybrid Explanations Framework for Collaborative Filtering Recommender Systems. », *RecSys Posters*, Citeseer, 2014.

Bennett J., Lanning S. *et al.*, « The netflix prize », *Proceedings of KDD cup and workshop*, vol. 2007, New York, NY, USA, p. 35, 2007.

Catherine R., Cohen W., « TransNets : Learning to Transform for Recommendation », *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, ACM, New York, NY, USA, p. 288-296, 2017.

Chen H., Sun M., Tu C., Lin Y., Liu Z., « Neural sentiment classification with user and product attention », *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, p. 1650-1659, 2016.

Cho K., van Merrienboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., Bengio Y., « Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation », *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1724-1734, 2014.

Dias C.-E., Guigue V., Gallinari P., « Recommandation et analyse de sentiments dans un espace latent textuel », *CORIA*, 2016.

Dias C.-E., Guigue V., Gallinari P., « Text-based collaborative filtering for cold-start soothing and recommendation enrichment », *AISR2017*, 2017.

Elman J. L., « Finding structure in time », *Cognitive science*, vol. 14, n° 2, p. 179-211, 1990.

Guy I., « Social recommender systems », *Recommender Systems Handbook*, Springer, p. 511-543, 2015.

Hochreiter S., Schmidhuber J., « Long short-term memory », *Neural computation*, vol. 9, n° 8, p. 1735-1780, 1997.

Honnibal M., « Embed, encode, attend, predict : The new deep learning formula for state-of-the-art NLP models », , https ://explosion.ai/blog/deep-learning-formula-nlp, 2016.

Joulin A., Grave E., Bojanowski P., Mikolov T., « Bag of Tricks for Efficient Text Classification », *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics : Volume 2, Short Papers*, vol. 2, p. 427-431, 2017.

Kingma D. P., Ba J., « Adam : A method for stochastic optimization », *arXiv preprint arXiv :1412.6980*, 2014.

Koren Y., « Collaborative filtering with temporal dynamics », *Communications of the ACM*, vol. 53, n° 4, p. 89-97, 2010.

Koren Y., Bell R., Volinsky C., « Matrix factorization techniques for recommender systems », *Computer*, 2009.

Ling G., Lyu M. R., King I., « Ratings meet reviews, a combined approach to recommend », *Proceedings of the 8th ACM Conference on Recommender systems*, ACM, p. 105-112, 2014.

McAuley J. J., Leskovec J., « From amateurs to connoisseurs : modeling the evolution of user expertise through online reviews », *Proceedings of the 22nd international conference on World Wide Web*, ACM, p. 897-908, 2013a.

McAuley J., Leskovec J., « Hidden factors and hidden topics : understanding rating dimensions with review text », *Proceedings of the 7th ACM conference on Recommender systems*, ACM, p. 165-172, 2013b.

McAuley J., Pandey R., Leskovec J., « Inferring networks of substitutable and complementary products », *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, p. 785-794, 2015.

Pang B., Lee L. *et al.*, « Opinion mining and sentiment analysis », *Foundations and Trends® in Information Retrieval*, vol. 2, n° 1–2, p. 1-135, 2008.

Parikh A. P., Täckström O., Das D., Uszkoreit J., « A decomposable attention model for natural language inference », *arXiv preprint arXiv :1606.01933*, 2016.

Poussevin M., Guigue V., Gallinari P., « Extended recommendation framework : Generating the text of a user review as a personalized summary », *Workshop on New Trends in Content-Based Recommender Systems, RecSys*, 2015.

Tintarev N., Masthoff J., « A survey of explanations in recommender systems », *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*, IEEE, p. 801-810, 2007.

Yang Z., Yang D., Dyer C., He X., Smola A., Hovy E., « Hierarchical attention networks for document classification », *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 1480-1489, 2016.