



République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique



Université des Sciences et de la Technologie Houari Boumediene

Faculté d'Informatique

Département d'Intelligence Artificielle et Science des Données

Mémoire de Licence

Filière : Informatique

Spécialité : Informatique Académique (ACAD)

Recommandation de produits e-commerce : Modèles hybrides d'analyse de sentiments par apprentissage profond

Sujet Proposé par :

Dr. Lamia BERKANI

Soutenu le : 21/06/2022

Présenté par :

M. Mouaadh HAMED ABDELOUAHAB

M. Ilyes DJERFAF

Devant le jury composé de :

Dr. Sadjia BABA ALI (Présidente)

Dr. Redouane BOUCHEKIR (Membre)

Projet ACAD_I_034/2022

Sommaire

1	ETAT DE L'ART	2
1.1	Introduction	2
1.2	Systèmes de recommandation	2
1.2.1	Définition	2
1.2.2	Approches de recommandation	2
1.3	Réseau de neurones et apprentissage profond	3
1.3.1	Apprentissage automatique (Machine Learning)	3
1.3.2	Réseau de neurones (Neural Networks)	3
1.3.3	Recommandation et apprentissage profond (Deep Learning)	5
1.4	Recommandation et Analyse des sentiments	5
1.4.1	Analyse des sentiments	5
1.4.2	Techniques d'analyse des sentiments	6
1.5	Travaux liés	7
1.5.1	Recommandation basée sur l'Analyse de Sentiments	7
1.5.2	Recommandation basée sur l'Apprentissage Profond	8
1.6	Conclusion	9
2	Conception	10
2.1	Introduction	10
2.2	Approche proposée	10
2.2.1	Motivation	10
2.2.2	Notre modèle SB-HDLM	11
2.3	Entraînement	19
2.3.1	Gestion du jeu de données (Dataset)	19
2.3.2	Entraînement	20
2.4	Conclusion	20
3	Implémentation et expérimentation	21
3.1	Introduction	21
3.2	Environnement de développement	21
3.2.1	Environnement logiciel	21
3.2.2	Présentation de l'application	22
3.3	Expérimentations	24
3.3.1	Datasets	24
3.3.2	Métriques d'évaluation	25
3.3.3	Résultats des évaluations	26
3.4	Discussion générale	30
3.5	Conclusion	31

Table des figures

1.1	Structure d'un neurone artificiel [9]	4
1.2	Réseau de neurones simple VS réseau de neurones profond [10]	5
1.3	Hiérarchie des différentes méthodes d'analyse d'opinion [12]	6
1.4	Schéma général représentant l'architecture du système proposé par Abbasi-Moud et al [15]	7
1.5	Schéma représentant l'architecture de l'approche NCTR [16]	9
2.1	Architecture générale du système de recommandation [1]	10
2.2	Méthodologie d'analyse des sentiments [1].	11
2.3	Architecture générale de notre approche SB-HDLM.	12
2.4	Exemple illustratif des avis et notes de la couche d'entrée extraite de Reviews_Movies_and_TV_5 Dataset [18]	12
2.5	Les opérations de prétraitement	12
2.6	Exemple de la mise en minuscule	13
2.7	Exemple de l'opération de suppression des ponctuations	13
2.8	Exemple de l'opération d'élimination des mots vides	13
2.9	Liste des mots vides [19].	14
2.10	Exemple de l'opération d'extraction du nom radical.	14
2.11	Aperçu du mécanisme de Word2Vec	15
2.12	Architecture du CNN	15
2.13	Représentation d'une convolution.	16
2.14	Visualisation avec Max Pooling.	16
2.15	Représentation de RNN à la fois sous forme pliée et dépliée.	17
2.16	Représentation d'une cellule RNN.	17
3.1	Interface de l'application – Fenêtre principale	23
3.2	Interface de l'application – Sélection du dataset	23
3.3	Interface de l'application – Sélection du modèle et des métriques	24
3.4	Figure représentant l'évaluation des variantes du système SB-HDLM en fonction du batch size sur le dataset Movies and TV -GloVe	27
3.5	Évaluation du modèle hybride avec une variation du poids d'importance α sur les deux datasets	29

Liste des tableaux

3.1	Statistiques relatives aux jeux de données.	25
3.2	Évaluation des modèles sur le dataset Movies and TV -GloVe.	26
3.3	Tableau représentant l'évaluation des modèles sur les différents datasets en fonction de la variation du nombre d'epochs -GloVe.	27
3.4	Tableau représentant l'évaluation des modèles sur les différents datasets en fonction de la variation de la taille de batch -GloVe.	28
3.5	Tableau représentant l'évaluation des modèles sur les différents datasets en fonction de la variation du nombre des epochs -Word2Vec.	28
3.6	Évaluation des modèles sur les deux datasets en fonction de la variation de la taille du batch -Word2Vec.	28
3.7	Comparaisons entre l'utilisation des GloVe et Word2Vec embeddings.	29
3.8	Comparaison des résultats des prédictions des modèles sur les deux datasets. . .	30

Remerciements

Louange à Allah, le tout miséricordieux, le très miséricordieux et le tout-puissant, d'avoir guidé nos pas vers les portes du savoir tout en illuminant notre chemin, et de nous avoir accordé la patience, la force, la volonté et le courage nécessaires à l'achèvement de notre travail dans de bonnes conditions.

La réalisation de ce travail a été possible grâce à la contribution de plusieurs personnes à qui nous voudrions témoigner notre profonde et sincère gratitude. Nous aimerions, d'abord, remercier notre promotrice Dr. BERKANI Lamia qui nous a permis de bénéficier d'un encadrement digne de ce nom. Les conseils qu'elle nous a prodigués, la patience, la confiance qu'elle nous a témoignée ont été déterminants dans la réalisation de notre travail.

Nous sommes également reconnaissants envers les membres du jury qui nous font l'honneur d'examiner notre modeste travail.

Nos vifs remerciements vont également envers nos parents, nos familles et nos amis pour leur soutien constant et leurs encouragements.

Enfin, nous remercions vivement tous nos enseignants de la Faculté d'Informatique, qui, durant nos années d'études, ont fourni un travail énorme pour notre formation.

Résumé

Le présent projet de licence propose une approche de recommandation d'items basée sur les techniques d'apprentissage profond et d'analyse de sentiments. Notre approche, intitulée SB-HDLM (Sentiment Based - Hybrid Deep Learning Models), combine les évaluations données par l'utilisateur avec les notes prédites par l'analyse de sentiments, où deux variations des modèles d'apprentissage profond hybrides CNN-LSTM et RNN-LSTM ont été développés avec deux word embedding différents Word2Vec et Glove.

Les résultats des expérimentations sur deux bases du domaine e-commerce, « Amazon Movies and TV » et « Amazon Video Games » ont montré que GloVe a généré de meilleures performances par rapport à Word2Vec et que le modèle RNN-LSTM a été plus performant que le modèle CNN-LSTM. De plus, notre modèle a surpassé plusieurs modèles existants en termes des métriques d'évaluation MAE et RMSE.

Mots clés : Recommandation Hybride, Analyse de Sentiments, Deep Learning, CNN, RNN, LSTM, word embedding, Word2Vec, Glove.

Abstract

This project proposes item recommendation approach based on deep learning and sentiment analysis techniques. Our approach entitled SB-HDLM (Sentiment Based – Hybrid Deep Learning Models), combines user- ratings with scores predicted by sentiment analysis, where two variations of the hybrid deep learning models have been used CNN-LSTM and RNN-LSTM, with two different word embeddings Word2Vec and GloVe.

Extensive experiments have been conducted on two e-commerce datasets, “Amazon Movies and TV” and “Amazon Video Games” showed that GloVe generated better performance compared to Word2Vec and that the RNN-LSTM model outperformed the CNN-LSTM model. Moreover, our model surpassed several existing models in terms of MAE and RMSE evaluation metrics.

Keywords : Hybrid Recommendation, Sentiment Analysis, Deep Learning, CNN, RNN, LSTM, word embedding, Word2Vec, GloVe.

Introduction générale

Avec le développement de l'Internet et des technologies de l'information et de communication, les systèmes de recommandation deviennent une solution incontournable pour surmonter le problème de surcharge d'information. En effet, il devient difficile de retrouver facilement les données appropriées à un utilisateur donné en un temps réduit vu l'évolution exponentielle des contenus ajoutés chaque jour. Cependant, malgré la capacité des systèmes de recommandation à surmonter cette difficulté, ces derniers souffrent toujours de quelques limitations liées aux problèmes de démarrage à froid et de rareté des évaluations.

Récemment, la croissance massive des médias sociaux et des sites e-commerce a conduit les utilisateurs à publier des avis ou des commentaires textuels, exprimant leurs préférences ou émettant leurs sentiments envers les items proposés. L'analyse de sentiments est une technique couramment utilisée pour déduire les préférences des utilisateurs.

Dans le cadre de notre projet de fin d'études de licence, nous proposons une approche nommée SB-HDLM (Sentiment Based - Hybrid Deep Learning Models), qui se base sur des variations des modèles d'apprentissage profond hybrides pour l'analyse de sentiments, à savoir : une variation combinant les réseaux de neurones convolutifs (CNN) avec LSTM (Long Short-Term Memory) et une variation combinant les réseaux de neurones récurrents (RNN) et LST. Notre modèle s'inspire principalement du travail : Dang et al [1], proposant des approches de recommandation basées sur l'analyse de sentiments.

Notre mémoire sera structuré en trois chapitres comme suit :

Chapitre 1 – État de l'art : présente les concepts de base de la recommandation, d'apprentissage automatique / profond et d'analyse de sentiments, ainsi que quelques travaux liés à l'utilisation de l'analyse de sentiments dans la recommandation.

Chapitre 2 – Conception : propose une nouvelle de recommandation qui se base sur des variations des modèles d'apprentissage profond hybrides pour l'analyse de sentiments.

Chapitre 3 – Implémentation et expérimentation : décrit les détails de l'implémentation de notre approche de recommandation ainsi que les expérimentations effectuées en utilisant deux datasets d'Amazon (la base la plus utilisée du domaine e-commerce).

Nous concluons ce mémoire par une conclusion avec quelques perspectives futures.

Chapitre 1

ETAT DE L'ART

1.1 Introduction

Ce chapitre présente des généralités sur les systèmes de recommandation, incluant les approches utilisées ainsi que les techniques d'apprentissage automatique et profond. Ensuite, l'analyse de sentiments et ses techniques sont introduites. Finalement, quelques travaux relatifs aux techniques de deep learning et d'analyse de sentiments sont exposés et analysés.

1.2 Systèmes de recommandation

1.2.1 Définition

Les systèmes de recommandation ou en anglais Recommender systems (RS) sont des algorithmes qui permettent de filtrer les données pour offrir une meilleure expérience aux utilisateurs en suggérant des items susceptibles de les intéresser selon leurs préférences. RS aide les utilisateurs à trouver le contenu, les produits ou les services appropriés dans un grand nombre de données en moins de temps.[2]

1.2.2 Approches de recommandation

Trois approches peuvent être distinguées : le filtrage à base de contenu, collaboratif et hybride.

- a. **Filtrage à base de contenu** : Un système à base de contenu, aussi appelé filtrage cognitif ou en anglais content-based filtering (CBF) sélectionne les éléments, appelés aussi items, en fonction de la comparaison entre le contenu des items et les préférences de l'utilisateur.[3]

Cette méthode présente quelques inconvénients tels que le manque de nouveauté et de diversité, c'est-à-dire que l'utilisateur ne verra que ce qui est indiqué sur son profil et que les items avec dont l'utilisateur n'a pas évalué ne seront jamais recommandés.

- b. **Filtrage collaboratif** : Alors que le CBF dépend du contenu des éléments pour générer des recommandations, le filtrage collaboratif, en anglais collaborative filtering (CF), recommande un item à un utilisateur en fonction de ce que les autres utilisateurs pensent de cet élément.

Bien que le CF soit l'un des systèmes de recommandation les plus populaires et les plus fréquemment utilisés, il présente plusieurs inconvénients :

- Problème de démarrage à froid : ce problème survient lorsque le système ne dispose pas de suffisamment de données pour générer des recommandations, que ce soit sur un utilisateur (user cold start), ou quand un nouvel item est ajouté au catalogue (item cold start).[4]
 - La rareté d'information (Data Sparsity) : ce problème est dû au manque d'évaluations des utilisateurs sur les items.[5]
- c. **Filtrage hybride** : Le principe de l'approche hybride est de combiner des algorithmes de filtrage collaboratif et basé contenu dans le but d'optimiser leurs performances et minimiser leurs inconvénients, à savoir le problème du démarrage à froid et de data sparsity.

L'état de l'art montre que les modèles basés sur les techniques d'apprentissage profond ont montré leur efficacité en termes de performances, nous allons présenter dans ce qui suit ces techniques ainsi que quelques travaux liés.

1.3 Réseau de neurones et apprentissage profond

1.3.1 Apprentissage automatique (Machine Learning)

L'apprentissage automatique, en anglais Machine learning (ML), est un domaine d'étude issu du domaine de l'intelligence artificielle (IA). ML est la technologie de développement d'algorithmes informatiques capables d'émuler l'intelligence humaine. C'est une science qui permet aux ordinateurs d'apprendre sans être explicitement programmés.[6]

Deux types d'apprentissage peuvent être distingués :

- a. **Apprentissage supervisé** : L'apprentissage supervisé, ou supervised learning (SL), est utilisé pour classer les données ou prédire les résultats. SL est basé sur un ensemble de données donné comme entrées et ses sorties correctes, ayant l'idée qu'il existe une relation entre l'entrée et la sortie.[6]
- b. **Apprentissage non supervisé** : L'apprentissage non supervisé (unsupervised learning) nous permet d'aborder des problèmes avec peu ou pas d'idée à quoi devraient ressembler nos résultats. Il peut dériver lui-même la structure des données. Les algorithmes d'apprentissage non supervisé peuvent effectuer des tâches de traitement plus complexes que celles de SL. Cependant, l'apprentissage non supervisé a tendance à être plus imprévisible qu'un modèle d'apprentissage supervisé.[6]

Au fil du temps, de nombreux algorithmes d'apprentissage automatique ont émergé, incluant les réseaux de neurones (NN) qui peuvent améliorer les performances des systèmes de recommandation.[7]

1.3.2 Réseau de neurones (Neural Networks)

A. Définition et fonctionnement

Les réseaux de neurones, en anglais Neural Networks (NN) ou réseaux de neurones artificiels (RNA) s'inspirent du cerveau humain, en remplissant les mêmes fonctions que les neurones biologiques : recevoir des signaux, les traiter puis envoyer des résultats.[8]

Chaque neurone artificiel a des entrées ou inputs (x_1 et x_2 sur la figure 1.1), avec des poids associés (w_1 et w_2 respectivement) et un biais b (ou seuil), et une fonction d'activation

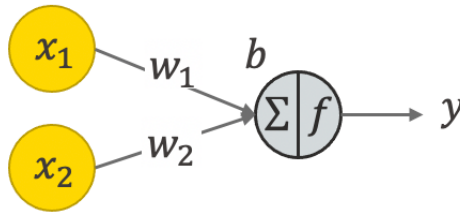


FIGURE 1.1 – Structure d'un neurone artificiel [9]

f. Pour calculer la sortie de chaque neurone, chaque entrée est multipliée par son poids, puis la somme de tous les termes plus le biais est calculée comme suit :

$$\sum_{i=0}^n x_i w_i + b = x_1 w_1 + x_2 w_2 + b \quad (1.1)$$

Ensuite, une fonction d'activation convertit le résultat en sortie de neurone :

$$sortie = y = f\left(\sum_{i=0}^n x_i w_i + b\right) = f(x_1 w_1 + x_2 w_2 + b) \quad (1.2)$$

Il existe de nombreuses fonctions d'activation, notamment : la fonction sigmoïde, la fonction Tanh, la fonction ReLU pour Rectified Linear Unit qui est une fonction d'activation (non linéaire) dont la formule est la suivante $g(x) = \max[0, x]$.

Les réseaux de neurones sont composés de couches de nœuds : une couche d'entrée (input layer), une ou plusieurs couches cachées (hidden layers) et une couche de sortie (output layer) (comme le montre la Figure 1.2 suivante). Des couches cachées peuvent être ajoutées au réseau pour plus de complexité ce qui le rend plus profond, on parle donc d'un réseau de neurones profond (Deep Neural Network) qui est exploité dans l'apprentissage profond (Deep Learning).

Le nœud est activé (ou déclenché) lorsque sa sortie est supérieure à la valeur de seuil spécifiée à l'aide de la fonction d'activation, envoyant ainsi des données à la couche suivante. Ce processus de transmission de données d'une couche à la suivante fait référence au réseau de neurones en tant que réseau de neurones à propagation directe (feedforward neural network - FFNN).

Il existe plusieurs types de réseaux de neurones (profonds), parmi lesquels : perception multicouches, réseau de neurones convolutifs et réseau de neurones récurrents.

B. Types de réseaux de neurones

- **Perception multicouches - Multilayer Perceptron (MLP)** : C'est un réseau de neurones à propagation directe (FFNN), qui est constitué de plusieurs couches, voir une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie. Les MLP sont souvent utilisés pour les problèmes de prédiction de classification où les entrées sont généralement étiquetées et pour les problèmes de régression

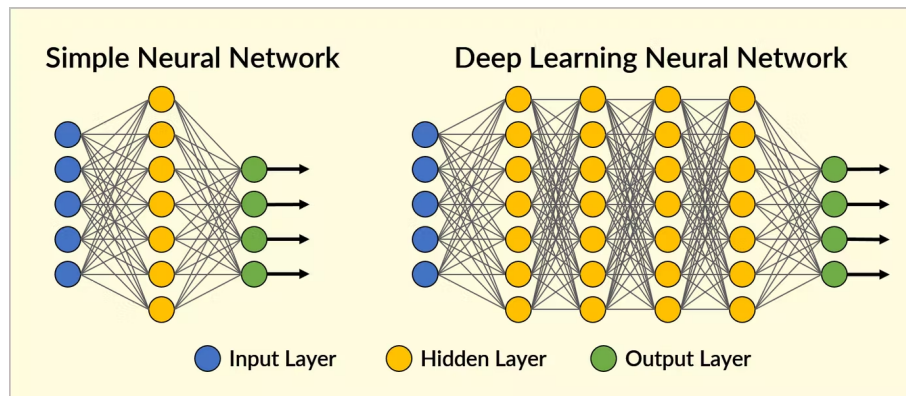


FIGURE 1.2 – Réseau de neurones simple VS réseau de neurones profond [10]

- **Réseau de neurones convolutifs - Convolutional Neural Network (CNN) :** Les réseaux de neurones convolutifs sont une technique de réseau de neurones artificiels puissantes développées pour des tâches de reconnaissance d'objets telles que la reconnaissance de chiffres manuscrits. On trouve trois types de couches dans un réseau de neurones convolutifs : couches convolutives, couches de pooling et des couches entièrement connectées ou couches de classification.
- **Réseau de neurones récurrents - Recurrent Neural Network (RNN) :** Les réseaux de neurones récurrents sont principalement utilisés dans la reconnaissance vocale et le traitement automatique du langage naturel (NLP) et la modélisation des données de séquence. Dérivés des réseaux à anticipation, les RNN présentent un comportement similaire à celui du cerveau humain. En termes simples, les réseaux neuronaux récurrents produisent des résultats prédictifs dans les données séquentielles.

1.3.3 Recommandation et apprentissage profond (Deep Learning)

L'apprentissage profond est un sous-domaine de l'apprentissage automatique concerné par des algorithmes à base de réseaux de neurones artificiels. Il est utilisé dans de nombreux domaines de l'intelligence artificielle, tels que la vision par ordinateur (CV) et le traitement du langage naturel (TAL).

Le deep learning commence par attirer un large intérêt dans le domaine des systèmes de recommandation, en améliorant l'efficacité de la recommandation vu qu'il porte plusieurs avantages, tels que : l'apprentissage profond peut atteindre des relations user-item complexes, De plus, en ayant plusieurs couches de traitement, le deep learning peut traiter un très grand nombre de données de façon précise.[11]

1.4 Recommandation et Analyse des sentiments

1.4.1 Analyse des sentiments

L'analyse des sentiments (sentiment analysis) ou le fouille d'opinions (opinion mining) indique un domaine du traitement du langage naturel (Natural Language Processing NLP), et de l'exploration de texte (Text Mining) qui vise à déterminer les émotions, l'appréciation ou l'avis d'un utilisateur à des sujets spécifiques. L'analyse des sentiments se concentre sur la polarité

d'un texte (positif, négatif, neutre) et détecte des sentiments et des émotions spécifiques (en colère, heureux, triste, etc.), l'urgence (urgent, pas urgent) et même des intentions (intéressé, pas intéressé).

1.4.2 Techniques d'analyse des sentiments

Plusieurs techniques ont été conçues pour l'analyse d'opinion comme nous le montre la figure ci-dessous :

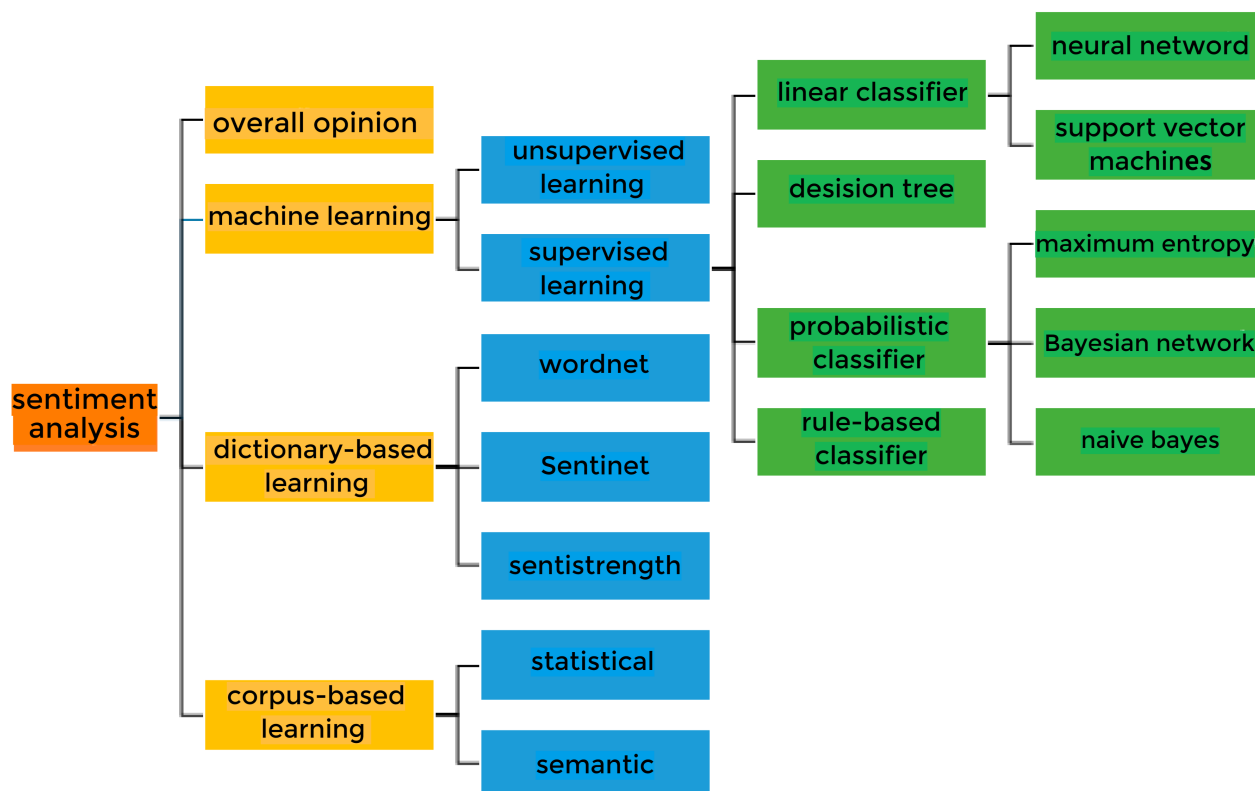


FIGURE 1.3 – Hiérarchie des différentes méthodes d'analyse d'opinion [12]

Parmi les techniques utilisées, nous nous intéressons aux méthodes à base de dictionnaires et à base d'apprentissage automatique.

- **Méthodes à base de dictionnaires** : Il s'agit d'approches basées sur des dictionnaires qui contiennent des synonymes et antonymes tel que WordNet. Dans ce type de dictionnaires, les mots sont organisés sous formes d'arbres qui relient tous les synonymes d'un mot de polarité préalablement définie.
- **Méthodes à base d'apprentissage automatique** : Ces approches utilisent les techniques d'apprentissage automatique. Tout d'abord, les ensembles de données sont formés et une analyse prédictive est effectuée. Le processus suivant consiste à extraire les mots du texte. Cette extraction de texte peut être effectuée à l'aide de différentes techniques telles que les algorithmes : Naive Bayes, Linear Regression, Support Vector Machine et Deep Learning.

1.5 Travaux liés

Récemment, des travaux ont été réalisés dans le domaine des systèmes de recommandation. Notre travail est lié aux études de recommandation basée sur l'analyse des sentiments et l'apprentissage profond.

1.5.1 Recommandation basée sur l'Analyse de Sentiments

Ces dernières années, de nombreux chercheurs ont proposé divers modèles d'analyse des sentiments sur divers sujets. Dans le domaine de la restauration, Asani et al.[13], ont proposé un système de recommandation sensible au contexte qui extrait les préférences alimentaires des individus de leurs commentaires et suggère des restaurants en fonction de ces préférences. Dans le contexte d'e-commerce, Karthik et Ganapathy [14], proposent un nouveau système de recommandation de produits basé sur la fuzzy logique qui prédit dynamiquement les produits les plus pertinents pour les clients dans les achats en ligne en fonction des intérêts actuels des utilisateurs en proposant un nouvel algorithme pour calculer le score sentimental du produit avec la catégorie cible d'utilisateur final associée. Nous allons détailler le travail de Abbasi-Moud et al.[15], qui propose un système de recommandation touristique qui exploite l'analyse des sentiments et l'exploration de texte pour identifier les préférences des utilisateurs ainsi que les principales caractéristiques des attractions touristiques. La figure 1.4 montre le schéma général du système :

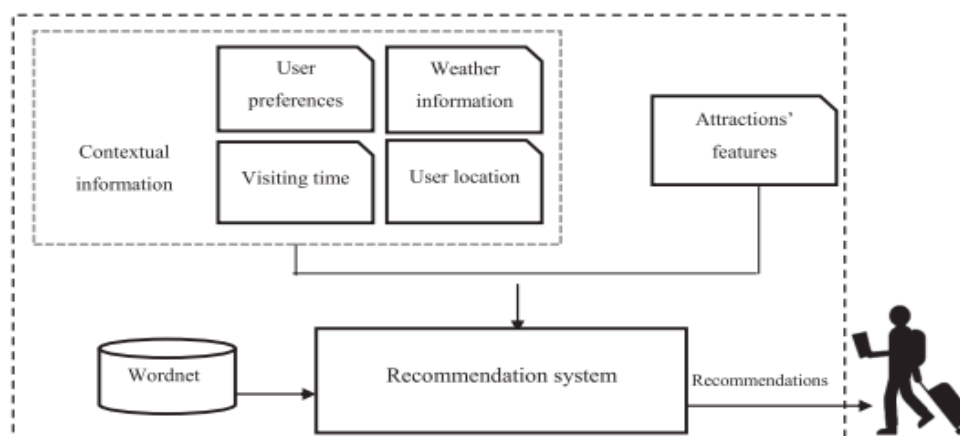


FIGURE 1.4 – Schéma général représentant l'architecture du système proposé par Abbasi-Moud et al [15]

Pour l'extraction des préférences contextuelles (Contextual information), quatre étapes principales comprenant le prétraitement, la formation de graphes sémantiques, le regroupement (clustering) et l'extraction des préférences sont effectuées. Le prétraitement est effectué pour convertir un document initial en une forme de traitement appropriée en extrayant des noms. Après pré-traitement, la matrice de similarité des noms est constituée à l'aide de WordNet et de la similarité sémantique de Wu-Palmer. C'est une matrice symétrique dans laquelle les lignes et les colonnes correspondent aux noms extraits. Ensuite, le graphe de cette matrice est créé et peut être converti en plusieurs sous-graphes connexes. Chaque sous-graphe est considéré comme un cluster dans lequel chaque nom a une similarité sémantique raisonnable avec au moins un autre nom. Chaque phrase est transférée aux clusters qui contiennent l'un de ses mots constitutifs. En continu, chaque groupe est noté en fonction du résultat de l'analyse des sentiments ainsi que de la fréquence des noms. Dans cette recherche, l'analyse des sentiments est effectuée

sémantiquement à l'aide de Sentiwordnet. Enfin, le cluster qui obtient le score le plus élevé est considéré comme le cluster sélectionné, représentant les préférences de l'utilisateur.

Après avoir extrait les préférences des utilisateurs, les caractéristiques des attractions touristiques doivent être extraites. Afin de tenir compte de la qualité des recommandations, les attractions qui ont reçu moins de 3 étoiles par les touristes sont ignorées. Ainsi, pour chaque attraction touristique, l'ensemble des avis des touristes qui ont visité l'attraction sont collectés et prétraités dans cinq conditions météorologiques différentes, notamment neigeuses, pluvieuses, ensoleillées, orageuses et partiellement nuageuses. Ensuite, les cinq mots les plus répétés de chaque attraction dans chacune de ces cinq conditions météorologiques sont extraites et sont considérés comme les caractéristiques de l'attraction.

L'idée principale du système de recommandation proposé est de comparer les préférences d'un touriste avec les caractéristiques des attractions à proximité et de renvoyer les attractions les plus similaires. Pour calculer la similarité des préférences d'un utilisateur avec les caractéristiques d'une attraction, la similarité maximale des éléments de préférence de l'utilisateur concernant toutes les caractéristiques de l'attraction est calculée. Par conséquent, ces similitudes maximales sont moyennées pour tous les éléments de préférence de l'utilisateur.

1.5.2 Recommandation basée sur l'Apprentissage Profond

Liu et al.[16], ont proposé un modèle de réseau neuronal hybride pour combiner des informations textuelles et des informations des évaluations (Neural network to Combine Textual information and Rating information-NCTR) pour la recommandation des items. L'objectif du NCTR proposé est d'exploiter le réseau de neurones pour apprendre les caractéristiques des avis et des évaluations, puis de coupler étroitement ces caractéristiques extraites pour effectuer des prédictions. Comme le montre la figure 1.5, le modèle NCTR proposé se compose de deux composants : **le composant d'apprentissage des caractéristiques** (feature learning component) et **le composant d'interaction des caractéristiques** (feature interaction component).

Le but du composant d'apprentissage des caractéristiques est d'apprendre les fonctionnalités contextuelles de l'élément et les fonctionnalités d'interaction user-item. Il est divisé en deux sections : Apprentissage contextuel des fonctionnalités (Contextual feature learning) et Apprentissage des fonctionnalités d'interaction (Interaction feature learning).

Dans l'apprentissage des caractéristiques contextuelles, CNN est utilisé pour extraire les caractéristiques des informations de description des éléments avec les trois composants suivants : Embedding layer (couche d'intégration) transforme les informations textuelles en une matrice vectorielle de mots, Convolution layer (couche de convolution) génère des caractéristiques latentes.

Après l'opération de convolution, le nombre des vecteurs de caractéristiques extraits peuvent être volumineux et la majorité d'entre eux ne peuvent pas révéler de manière significative les caractéristiques contextuelles, donc Pooling layer (la couche de regroupement) sélectionne les caractéristiques représentatives de la sortie de la couche précédente en utilisant la méthode du max pooling.

Pour l'Apprentissage des fonctionnalités d'interaction, le vecteur de caractéristique d'utilisateur et le vecteur de caractéristique d'élément codés à chaud sont mappés sur des vecteurs de facteurs latents denses de faible dimension (low-dimension) par l'intermédiaire des matrices de facteurs latents dans la couche d'incorporation. Ensuite, les fonctionnalités extraites seront exploitées pour capturer les préférences des utilisateurs. Les tâches principales des composants d'interaction de caractéristiques sont la fusion de caractéristiques et l'interaction de caractéristiques. L'intégration des caractéristiques contextuelles et des caractéristiques latentes d'éléments est

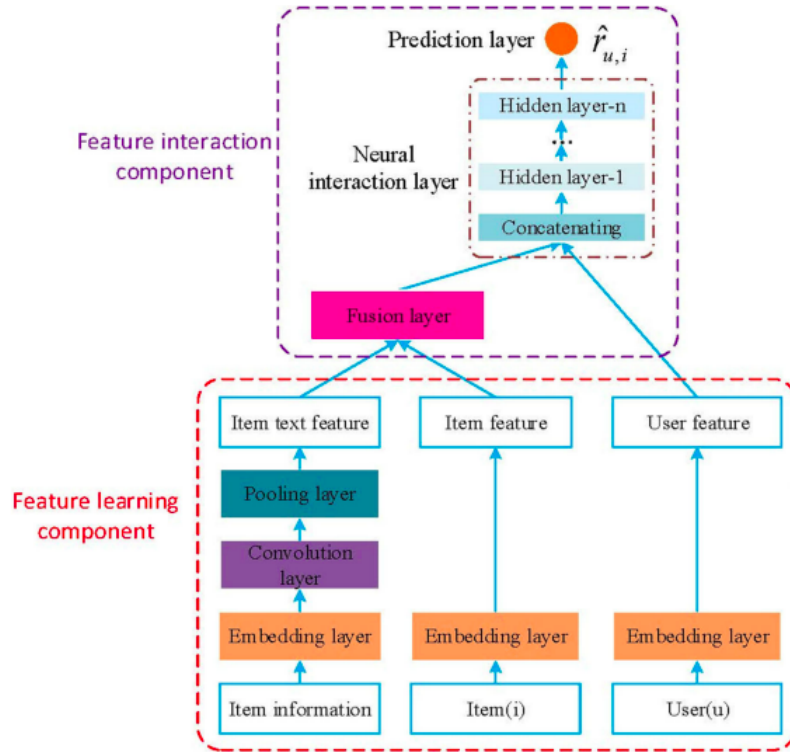


FIGURE 1.5 – Schéma représentant l'architecture de l'approche NCTR [16]

réalisée via la couche de fusion (fusion layer), qui peut implémenter les caractéristiques étroitement couplées. La couche d'interaction neuronale (Neural Interaction Layer) réalise l'interaction non linéaire des caractéristiques. Pour réaliser plusieurs interactions user-item, MLP est empilé sur la couche de connexion pour découvrir la structure complexe des interactions de fonctionnalités.

Enfin, La sortie de la couche précédente est considérée comme l'entrée de la couche de prédiction et la sortie de la couche de prédiction correspond aux prédictions des évaluations des utilisateurs.

1.6 Conclusion

Nous avons abordé dans ce chapitre quelques concepts liés aux systèmes de recommandation, passant par les principes de base de l'apprentissage automatique, de l'apprentissage profond, de l'analyse de sentiments ainsi quelques travaux liés. Cette étude nous a permis de constater l'apport du deep learning et de l'analyse des sentiments dans l'amélioration des performances des systèmes de recommandation. L'intégration des commentaires permet une meilleure interprétation des évaluations des utilisateurs et donc offre la possibilité de connaître les préférences des utilisateurs. Dans le chapitre suivant, nous proposerons une approche hybride de recommandation tout en s'inspirant des principes et travaux précédemment cités.

Chapitre 2

Conception

2.1 Introduction

Dans ce chapitre, nous proposons notre propre méthode de recommandation basée sur l'analyse des sentiments des commentaires des utilisateurs à l'aide de techniques d'apprentissage profond.

Ce chapitre est composé de deux parties : (1) une présentation détaillée de l'approche proposée ; et (2) les détails liés à l'entraînement de notre modèle.

2.2 Approche proposée

2.2.1 Motivation

Notre travail s'inspire de celui de Dang et al. [1], qui ont présenté une approche de recommandation qui combine le filtrage collaboratif et l'analyse de sentiments. L'architecture de leur système est illustrée par la figure 2.1.

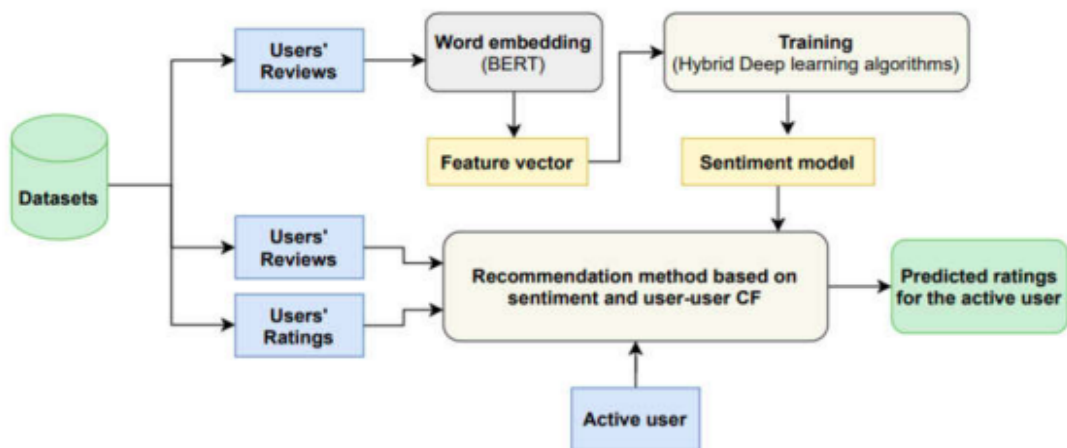


FIGURE 2.1 – Architecture générale du système de recommandation [1]

L'architecture comporte deux parties distinctes, une partie chargée de générer l'analyse de sentiments et l'autre partie de fournir des recommandations à un utilisateur donné. La figure 2.2 montre le modèle utilisé pour l'analyse de sentiments.

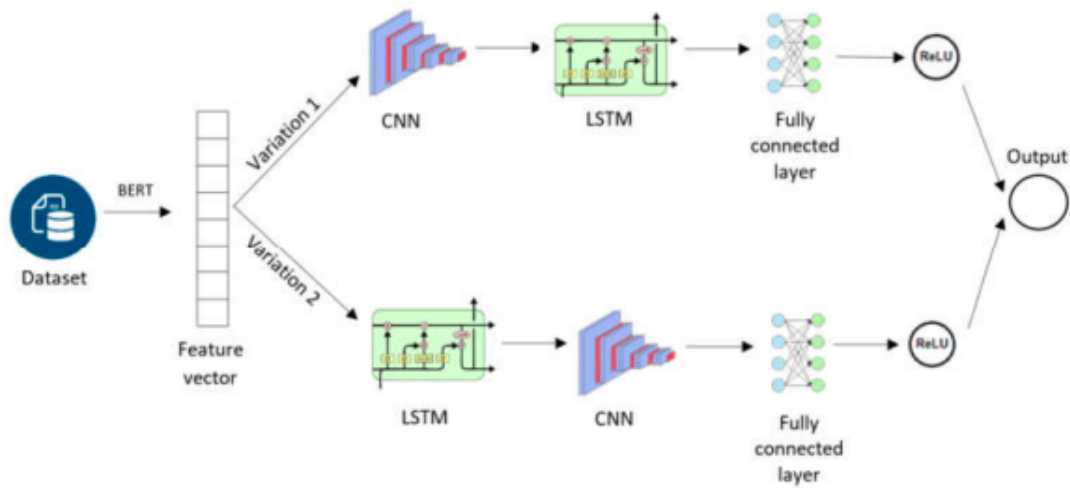


FIGURE 2.2 – Méthodologie d'analyse des sentiments [1].

Dans ce modèle, les données des commentaires ont été introduites dans le modèle BERT [17], pour générer les vecteurs de caractéristiques (feature vectors), qui sont ensuite entrés dans les modèles hybrides pour la classification. La prochaine étape combine les modèles d'apprentissage profond CNN et LSTM. La dernière étape est la classification à l'aide de la fonction d'activation ReLu.

Dans le but d'améliorer ce modèle d'analyse des sentiments, nous introduisons une nouvelle variation qui exploite les réseaux de neurones récurrents (RNN) et LSTM. De plus, nous intégrons une couche de combinaison afin de combiner les notes prédites par la couche d'apprentissage profond avec les notes réelles des items pour achever des meilleures prédictions.

2.2.2 Notre modèle SB-HDLM

A. Description générale du modèle et son architecture

Le système de recommandation basé sur le modèle SB-HDLM (Sentiment Based - Hybrid Deep Learning Models) se base sur des variations des modèles d'apprentissage profond hybrides pour l'analyse de sentiments. Les avis/commentaires des utilisateurs U sur des items I ainsi que leurs notes sont passées en entrée. En sortie, le système devra retourner une prédiction de la note d'un item donné i après avoir analysé l'avis d'un utilisateur donné u sur cet item i , tout en exploitant les différents modèles d'apprentissage automatique et en combinant enfin la note prédite avec la note réelle donnée en entrée par le même utilisateur u .

Le schéma représenté par la figure 2.3 décrit l'architecture de l'approche proposée.

B. Description des couches du modèle

- **Couche d'entrée (Input layer)** : La première couche représente les différentes interactions entre les utilisateurs et les items. Elle comporte deux entrées principales :
 - Les avis/commentaires (reviews) des utilisateurs sur les items.
 - Les notes (ratings) des utilisateurs sur les items sur une échelle de 1 à 5.
 La figure 2.4 illustre le contenu de la couche d'entrée.
- **Couche de prétraitement (Preprocessing layer)** : Le processus de conversion en données compréhensibles et interprétables par un ordinateur peut comprendre est appelé

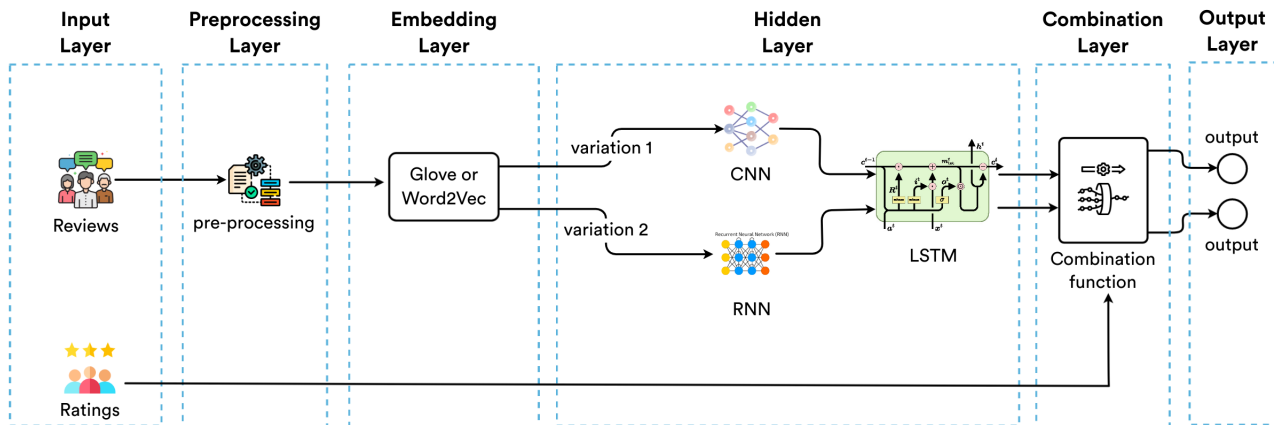


FIGURE 2.3 – Architecture générale de notre approche SB-HDLM.

index	reviewText	rating
31418	I liked this film somewhat and thought the premise was very interesting. The cast is excellent, but the pacing is slow and everyone involved seems to be too calm considering what's going on. The film suffers from being too long and a good 20 minutes trimmed off would have helped a great deal. Denzel Washington holds it all together in this ultimately predictable thriller, but I doubt I would sit through it a second time. I rate it 2 1/2 stars for a good premise and cast.	2
77169	This has always been my favorite of the Pink Panther movies. There was no one like Peter Sellers, he was the best!!	5
47099	He loved this movie back ago he still loves it Adam Sandler is awesome he still watches this on a weekly basis years later	5

FIGURE 2.4 – Exemple illustratif des avis et notes de la couche d'entrée extraite de Reviews__ - Movies_and_TV_5 Dataset [18]

prétraitement. L'une des principales formes de prétraitement consiste à filtrer les données inutiles.

Dans notre système, le prétraitement comprend la mise en minuscules de tous les mots, la suppression des ponctuations, la suppression des mots vides et l'extraction du nom radical, comme indiqué par la figure 2.5.

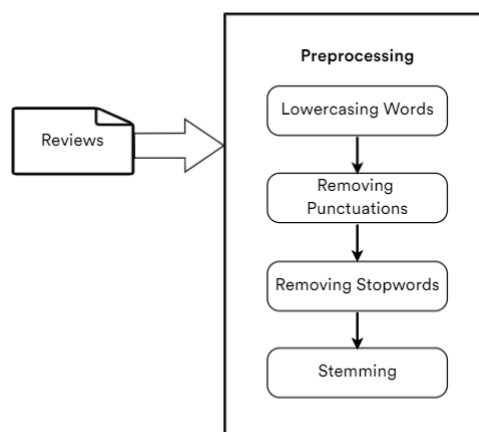


FIGURE 2.5 – Les opérations de prétraitement

- Mise en minuscules de tous les mots (Lowercasing words) : Les mêmes mots écrits dans des casses différents sont considérés comme des entités différentes par l'ordinateur. Par exemple : 'Games' et 'games' sont considérés comme deux mots distincts par l'ordinateur même s'ils représentent le même mot. Afin de résoudre ce problème, nous devons convertir tous les mots en minuscules. Cela permet d'uniformiser le texte.

index	reviewText	rating	clean_review
34432	King Kong is the classic monster movie. In 1932, special effects in film was a lot more work than todays computer graphics. Kong was an 18 inch model filmed by single framing. The sets were hand made and painted for a look that has never been equaled. The beauty and the beast story is one where you can feel sorry for the monster.	5	king kong is the classic monster movie. in 1932, special effects in film was a lot more work than todays computer graphics. kong was an 18 inch model filmed by single framing. the sets were hand made and painted for a look that has never been equaled. the beauty and the beast story is one where you can feel sorry for the monster.

FIGURE 2.6 – Exemple de la mise en minuscule

- Suppression des ponctuations (Removing Punctuations) : En travaillant sur des données textuelles générées par l'utilisateur, comme dans les avis sur le commerce électronique, il y aura trop de ponctuation dans les phrases, ce qui peut ne pas être utile pour la tâche à accomplir. Par conséquent, il est nécessaire de supprimer ces caractères.

index	reviewText	rating	clean_review
67983	This is the French original; it was later remade (badly) as Dinner for Schmucks in English. GAME is remarkably well-written and well-acted, and is staged almost like a play, with nearly the entire action taking place in one apartment. For anyone who can stand reading a movie -- or who speaks French -- it's well worth seeing; it's a great comedy with some real depth to it.	5	this is the french original it was later remade badly as dinner for schmucks in english game is remarkably wellwritten and wellacted and is staged almost like a play with nearly the entire action taking place in one apartment for anyone who can stand reading a movie or who speaks french its well worth seeing its a great comedy with some real depth to it

FIGURE 2.7 – Exemple de l'opération de suppression des ponctuations

- Élimination des mots vides (Removing Stopwords) : Dans une phrase, les mots qui n'ont pas de sens précis sont appelés mots vides (stopwords). En éliminant les mots vides, seuls les mots avec des informations utiles restent. La figure suivante illustre le processus d'élimination des mots vides :

index	reviewText	rating	clean_review
29631	5 star film, excellent presentation but owners of multiregion players should be aware that the US disk contains fewer scenes than the European, and exactly the same number of extras.	4	5 star film excellent presentation owners multiregion players aware us disk contains fewer scenes european exactly number extras

FIGURE 2.8 – Exemple de l'opération d'élimination des mots vides

Nous pouvons visualiser la liste des mots vides utilisée comme suit :

```
import nltk
from nltk.corpus import stopwords
print(stopwords.words('english'))
```

{‘ourselves’, ‘hers’, ‘between’, ‘yourself’, ‘but’, ‘again’, ‘there’, ‘about’, ‘once’, ‘during’, ‘out’, ‘very’, ‘having’, ‘with’, ‘they’, ‘own’, ‘an’, ‘be’, ‘some’, ‘for’, ‘do’, ‘its’, ‘yours’, ‘such’, ‘into’, ‘of’, ‘most’, ‘itself’, ‘other’, ‘off’, ‘is’, ‘s’, ‘am’, ‘or’, ‘who’, ‘as’, ‘from’, ‘him’, ‘each’, ‘the’, ‘themselves’, ‘until’, ‘below’, ‘are’, ‘we’, ‘these’, ‘your’, ‘his’, ‘through’, ‘don’, ‘nor’, ‘me’, ‘were’, ‘her’, ‘more’, ‘himself’, ‘this’, ‘down’, ‘should’, ‘our’, ‘their’, ‘while’, ‘above’, ‘both’, ‘up’, ‘to’, ‘ours’, ‘had’, ‘she’, ‘all’, ‘no’, ‘when’, ‘at’, ‘any’, ‘before’, ‘them’, ‘same’, ‘and’, ‘been’, ‘have’, ‘in’, ‘will’, ‘on’, ‘does’, ‘yourselves’, ‘then’, ‘that’, ‘because’, ‘what’, ‘over’, ‘why’, ‘so’, ‘can’, ‘did’, ‘not’, ‘now’, ‘under’, ‘he’, ‘you’, ‘herself’, ‘has’, ‘just’, ‘where’, ‘too’, ‘only’, ‘myself’, ‘which’, ‘those’, ‘i’, ‘after’, ‘few’, ‘whom’, ‘t’, ‘being’, ‘if’, ‘theirs’, ‘my’, ‘against’, ‘a’, ‘by’, ‘doing’, ‘it’, ‘how’, ‘further’, ‘was’, ‘here’, ‘than’}

FIGURE 2.9 – Liste des mots vides [19].

- Extraction du nom radical (Stemming) : La radicalisation fait référence au processus de conversion d’un mot en sa base ou sa racine. Dans cette sous-étape, des mots tels que ”player” et ”playing” sont transformés en leur racine, c’est-à-dire ”play”. À cette fin, l’algorithme de stemming Snowball Stemmer est utilisé.

index	reviewText	rating	clean_review	stemmed_review
9951	Great Movie, excellant color transfer, Bogart at his best. Superb cast and acting. The story line holds up even today.	5	great movi excel color transfer bogart best superb cast act stori line hold even today	great,movi,excel,color,transfer,bogart,best,superb,cast,act,ston, line,hold,even,today

FIGURE 2.10 – Exemple de l’opération d’extraction du nom radical.

- **Couche d’embedding (Embedding layer)** : Comme nous le savons, lorsque nous traitons des données textuelles, nous devons les convertir en nombres avant d’alimenter tout modèle d’apprentissage automatique, y compris les réseaux de neurones. Pour ce faire, il existe plusieurs méthodes. Dans la méthode classique, nous utilisons un encodage one-hot pour convertir les données catégoriques en nombres en créant des entités fictives (dummy features) pour chacune des catégories et en les remplissant avec des 0 et des 1. Mais de nos jours, l’utilisation de word embedding est considérée comme une meilleure manière de représenter des données textuelles.

La couche d’embedding nous permet de convertir chaque mot en un vecteur de longueur fixe de taille définie. Le vecteur résultant est dense avec des valeurs réelles au lieu d’avoir seulement des 0 et 1. La longueur fixe des vecteurs de mots nous aide à mieux représenter les mots avec des dimensions réduites. De plus, il est capable de comprendre le contexte d’un mot afin que des mots similaires aient des embeddings similaires. Ce vecteur de sortie est l’entrée de la couche cachée du réseau neuronal.

Dans notre travail, nous avons opté pour développer la couche d’embedding en utilisant deux méthodes différentes du word embedding : Word2Vec et GloVe, en vue de déterminer la plus adaptée à notre contexte d’études.

- **Word2Vec** : Word2vec est un groupe de modèles associés qui sont formés pour reconstruire les contextes linguistiques des mots. Word2Vec est basé sur des architectures de Continuous Bag-of-Words (CBOW) et Skip-gram. CBOW prédit un mot en fonction de son contexte et Skip-gram peut prédire le contexte en fonction d'un mot. Après l'entraînement, les modèles word2vec peuvent être utilisés pour mapper chaque mot à un vecteur de plusieurs centaines d'éléments, qui représentent la relation de ce mot avec d'autres mots.[20]

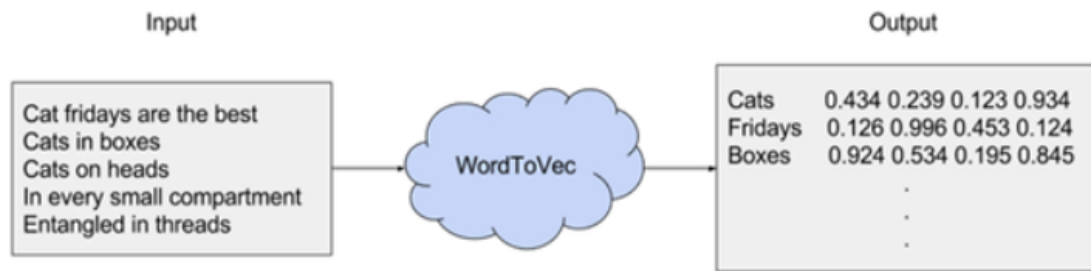


FIGURE 2.11 – Aperçu du mécanisme de Word2Vec

- **GloVe (Global Vectors)** : Le modèle GloVe est entraîné sur les entrées non nulles d'une matrice globale de cooccurrence mot-mot, qui indique la fréquence avec laquelle les mots coexistent les uns avec les autres dans un corpus donné. Le remplissage de cette matrice nécessite un seul passage dans l'ensemble du corpus pour collecter les statistiques. Les itérations des entraînements suivants sont beaucoup plus rapides, car le nombre d'entrées de matrice non nulles est généralement beaucoup plus petit que le nombre total de mots dans le corpus.
- **Couche cachée (Hidden layer)** : Dans la quatrième couche de notre modèle, le but est de générer des prédictions des notes en sortie. Nous avons utilisé la combinaison de plusieurs approches. Nous commençons par faire varier les modèles CNN (variante 1) et RNN (variante 2) qui ont comme entrée les vecteurs d'embeddings, puis nous combinons chaque variation avec le modèle LSTM. La dernière étape du modèle utilise une fonction d'activation «Softmax».
- **Variante 1 (CNN)** : Les vecteurs de mots de la couche d'embedding sont transmis en entrée à notre modèle CNN qui contient une couche convolutive avec une fonction d'activation ReLU et une couche de regroupement (pooling).

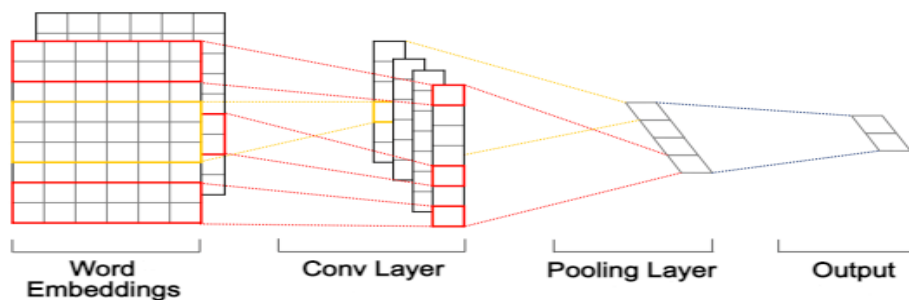


FIGURE 2.12 – Architecture du CNN

Une convolution est l'application d'un ou plusieurs filtres à une entrée. L'application répétée du même filtre à une entrée donne une carte de caractéristique (feature map). Nous utilisons généralement des filtres qui glissent sur les embedding des mots - les lignes de matrice. En analysant les séquences, on peut mieux comprendre le sens d'une phrase.

La figure 2.13 montre le fonctionnement d'un filtre d'une taille de 3x3. Après avoir être créé, il commence à effectuer une multiplication par éléments en partant du coin supérieur gauche de matrix d'entrée. La multiplication par élément signifie que l'on multiplie les éléments ayant le même index. Ces valeurs calculées sont additionnées pour obtenir une valeur de mapping et elles sont stockées dans la nouvelle matrice. Cette matrice de caractéristiques est transmise à la couche suivante comme entrée.

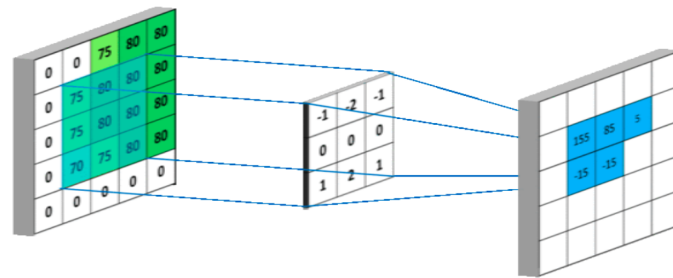


FIGURE 2.13 – Représentation d'une convolution.

La couche de pooling sert à consolider les caractéristiques apprises dans la carte des caractéristiques (feature map) afin de compresser et généraliser les représentations des caractéristiques. Nous avons choisi d'appliquer Max Pooling. L'idée derrière l'utilisation de Max Pooling est de sélectionner la valeur maximale dans la matrice de taille spécifiée. Cette méthode est utile pour extraire les caractéristiques d'une grande importance ou qui sont mises en évidence dans la matrice de mots.

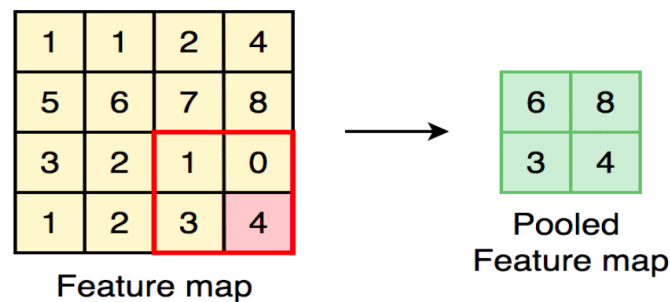


FIGURE 2.14 – Visualisation avec Max Pooling.

- **Variante 2 (RNN) :** Les réseaux de neurones récurrents (RNN) sont utilisés pour les problèmes qui traitent des données séquentielles. Ils sont devenus de plus en plus populaires en raison de leurs excellents résultats dans le traitement du langage naturel (TAL). Les RNN sont généralement affichés comme dans la figure ci-dessous :

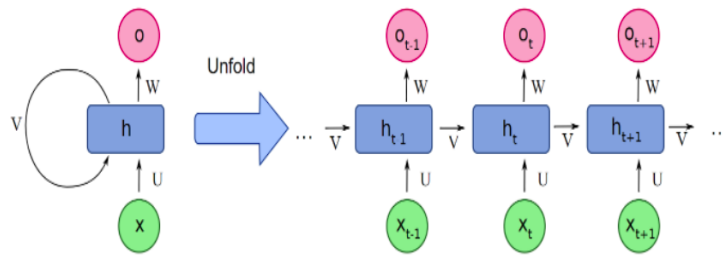


FIGURE 2.15 – Représentation de RNN à la fois sous forme pliée et dépliée.

Où :

x : représente l'entrée concernée du vecteur de mots (données séquentielles).

o : représente la sortie.

h : le bloc principal de la RNN.

v : représente la communication de la sortie comme une entrée elle-même.

Le type de bloc h le plus simple se trouve dans la figure ci-dessous.

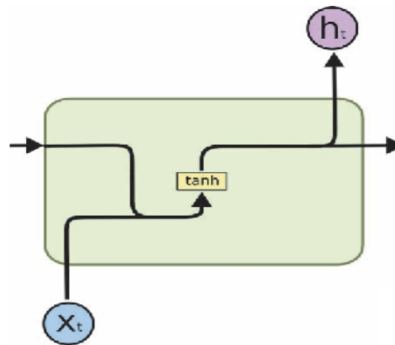


FIGURE 2.16 – Représentation d'une cellule RNN.

Ce bloc combine simplement l'entrée avec sa sortie et le fait passer par une activation \tanh pour obtenir la sortie.

Le bloc de construction de base pour RNN présenté ci-dessus souffre de certains problèmes. L'un des plus importants est l'incapacité à retenir l'information lorsque la séquence donnée est longue, comme les vecteurs de mots dans notre cas. Il oublie les informations qui ont été fournies précédemment. Cela limite les performances d'apprentissage. Les gens ont donc créé des architectures pour résoudre ce problème.

Les plus populaires sont la mémoire longue à court terme (LSTM) et l'unité récurrente fermée (GRU).

Dans notre approche, nous utilisons LSTM dans la variante de RNN qui sera détaillé dans la section prochaine.

- **Passage par LSTM :** (Long Short-Term Memory) est un type de RNN, largement utilisé pour l'apprentissage de problèmes de prédiction de données séquentielles. Le fonctionnement de base de LSTM peut être considéré comme contenant les infor-

mations requises et rejetant les informations qui ne sont pas requises ou utiles pour une prédiction ultérieure.

L'intégration de la couche LSTM, à la fin de chaque variation, peut avoir sa propre particularité. LSTM a une fonctionnalité grâce à laquelle il peut mémoriser la séquence des données. Il a une autre caractéristique qui fonctionne sur l'élimination des informations inutilisées et comme nous le savons, les données textuelles consistent toujours en beaucoup d'informations inutilisées qui peuvent être éliminées par le LSTM afin que le temps et le coût de calcul puissent être réduits.

Nous tenons à mentionner que dans la deuxième variante de notre modèle qui utilise RNN, la couche LSTM peut toujours être considérée comme une sous-couche déjà intégrée vu qu'elle est récurrente.

- **Fonction d'activation** : Après LSTM, une couche dense est ajoutée qui est une couche de sortie avec une fonction d'activation «Softmax».

La fonction d'activation Softmax est utilisée lorsque nous avons deux ou plus de deux classes. Dans notre cas, nous avons un total de cinq classes (des prédictions de 1 à 5), le nombre de neurones dans la couche de sortie sera donc de 5. Chaque neurone représente une classe. Les cinq neurones renverront les probabilités du texte d'entrée pour la classe respective. La classe avec la probabilité la plus élevée sera considérée comme une sortie pour ce texte.

- **Couche de combinaison (Combination layer)** : Cette couche sert à combiner la note prédite par notre module d'analyse de sentiments d'un item avec sa note réelle, pour obtenir la note finale. Pour ce faire, nous avons développé une fonction de combinaison décrite par l'algorithme suivant :

Algorithme 1 : Fonction de combinaison

Entrée : Note prédite d'item i , Note réelle d'item i , pourcentage α ;
Sortie : Note_finale ;
Début
 note_finale = $\alpha \times$ (Note prédite) + $(1 - \alpha) \times$ (Note réelle) ;
 retrun note_finale ;

Où : α est un nombre réel entre 0 et 1.

- **Couche de sortie (Output layer)** : Cette couche présente le résultat de la note finale d'un item i générée à partir de la couche de combinaison.

Les étapes par lesquelles passe notre modèle pour effectuer cette prédiction sont décrites dans l'algorithme ci-dessous :

Algorithme 2 : Fonction de combinaison

Entrée : Avis des utilisateurs A , Notes des utilisateurs N ;
Sortie : Notes prédites pour chaque avis ;
Début

- 1 Nettoyer les avis des utilisateurs A (prétraitement) ;
- 2 Transformer les avis des utilisateurs A sous forme de vecteurs de facteurs latents soit en utilisant : GloVe ou Word2Vec ;
- 3 Entraîner le modèle d'analyse des sentiments ;
 - si** *Variante 1* **alors**
 - _ Entraîner le modèle CNN ;
 - sinon**
 - _ Entraîner le modèle RNN ;
- 4 Appliquer LSTM pour générer une meilleure classification textuelle ;
- 5 **pour chaque** avis $a \in A$ **faire**
 - Note_finale_a = Combinaison de la note prédite de l'avis a avec sa note réelle ;
- return** Note_finale_a ;

2.3 Entraînement

Nous présentons dans cette section le processus d'entraînement de notre modèle.

2.3.1 Gestion du jeu de données (Dataset)

A. Exploitation des données

Les données extraites à partir des jeux de données doivent être traitées et transformées avant de passer par le modèle. Le prétraitement général appliqué aux données se fait comme décrit dans la section 2.2.2. Nous prenons comme exemple le traitement des données de « Amazon Review Data 2014 [18] », qui offre la possibilité d'étudier les évaluations et opinions des utilisateurs, dont le traitement est expliqué à la section ??.

Nous avons opté pour une méthode basée sur l'exploitation des données textuelles fournies par l'utilisateur à travers ses commentaires. Nous avons basé notre approche sur deux variantes hybrides des structures de réseaux neuronaux, CNN-LSTM et RNN-LSTM, s'inspirant des méthodes proposées par Dang et al. [16], s'appuyant sur l'analyse de sentiments des commentaires de l'utilisateur.

B. Répartition des données du dataset

Nous allons répartir (split) le dataset de manière aléatoire entre les données d'entraînement et de test. Le jeu de données est réparti comme suit : 80% pour l'entraînement et 20% pour le test.

- **Données d'entraînement** : Le premier sous-ensemble est connu sous le nom de données d'entraînement, c'est une partie de notre ensemble de données qui est introduite dans

notre approche pour découvrir et apprendre des patrons. De cette façon, il entraîne notre modèle. Les données d'entraînement sont généralement plus volumineuses que les données de test. En effet, nous voulons alimenter le modèle avec autant de données que possible pour trouver et apprendre des patrons significatifs.

- **Données de test :** Une fois notre approche est construit avec les données d'entraînement, nous avons besoin de données inédites pour tester notre modèle. Ces données sont appelées données de test et que nous pouvons utiliser pour évaluer les performances et la progression de notre modèle.

2.3.2 Entraînement

A. Fonction de coût (loss function)

Dans tout projet d'apprentissage profond, la configuration d'une fonction de coût est l'une des étapes les plus importantes pour garantir que le modèle fonctionne comme prévu. Une fonction de coût peut donner à un réseau de neurones une grande flexibilité pratique et définira exactement comment la sortie du réseau est connectée au reste du réseau.

Une fonction de coût peut être définie comme une fonction qui considère deux paramètres : la sortie prévue (Y_pred) et la sortie réelle (Y). Cette fonction calculera essentiellement la mauvaise performance de notre modèle en comparant la valeur prédite par le modèle avec la valeur réelle qu'il devrait produire. Si Y_pred est éloigné de Y , alors la valeur de perte sera élevée. Cependant, si les deux valeurs sont presque similaires, alors la valeur de perte sera faible.

Dans notre approche, nous exploitons la fonction de coût appelée «Categorical Cross-Entropy» qui calcule l'erreur comme suit :

$$LOSS = - \sum_{i=0}^n y_i \times \log(p_i) \quad (2.1)$$

Où : y_i est la note réelle et p_i est la note de prédiction.

B. Algorithme d'optimisation (optimizer)

Dans le but d'améliorer la performance de notre approche, nous allons essayer de minimiser la fonction de coût, car une perte plus faible signifie que notre modèle fonctionnera mieux en utilisant une fonction d'optimisation. Nous utiliserons dans notre modèle l'algorithme d'Adam (Adaptive Moment estimate) vu que c'est l'algorithme d'optimisation le plus utilisé et qui représente une méthode d'optimisation stochastique [21].

2.4 Conclusion

Après avoir expliqué les détails de conception de notre modèle «SB-HDLM» basé sur l'analyse des sentiments et les réseaux de neurones. Nous allons présenter dans le chapitre suivant les aspects techniques liés à l'implémentation et à l'évaluation de notre modèle.

Chapitre 3

Implémentation et expérimentation

3.1 Introduction

Dans ce chapitre, nous introduisons d’abord les aspects liés au développement de notre système de recommandation SB-HDLM. Ensuite, nous montrons également notre application pour illustrer les modèles d’utilisation et les différentes fonctionnalités développées. Enfin, nous détaillerons les métriques utilisées pour évaluer le modèle, puis présenterons les différents résultats des expérimentations réalisées.

3.2 Environnement de développement

3.2.1 Environnement logiciel

A. Langage de programmation

Pour implémenter notre système, nous avons choisi la version 3.10.0 du langage de programmation Python en raison de sa simplicité, lisibilité et flexibilité, de sa documentation riche, sa performance dans l’élaboration des applications de traitement de texte et de ses nombreuses bibliothèques qui facilitent les architectures d’apprentissage automatique et le développement d’apprentissage profond.

B. Bibliothèques

Voici les bibliothèques qui nous ont aidé à créer des modèles et à traiter les données.

- NumPy [22] : un package de calcul scientifique en Python. Il est idéal pour les opérations liées à l’algèbre linéaire et aux opérations sur les nombres aléatoires. Il peut être utilisé comme conteneur de données générique multidimensionnel.
- Pandas [23] : l’une des bibliothèques de science des données les plus populaires. Elle est utilisée pour les structures et l’analyse de données de haut niveau. Elle facilite la fusion et le filtrage des données et également la collection des données à partir d’autres sources externes comme Excel, CSV et JSON.
- Nltk [24] : Natural Language Toolkit est une bibliothèque logicielle Python pour le traitement automatique du langage qui fournit des démonstrations graphiques et des exemples de données. Cette bibliothèque nous fournit toutes les ressources linguistiques nécessaires au prétraitement des données.

- Gensim [25] : est une bibliothèque Python pour la modélisation de sujets, l'indexation de documents et la recherche de similitudes avec de grands corpus. Le public cible est les communautés du traitement du langage naturel (TAL) et de la recherche d'informations.
- Scikit-learn [26] : est la bibliothèque de référence pour l'apprentissage automatique. Elle est utilisée pour gérer les algorithmes ML fondamentaux tels que le clustering, la régression linéaire et logistique, la classification et la régression, entre autres.
- Keras [27] : Une API conçue pour créer des architectures d'apprentissage en profondeur et permettre une expérimentation rapide avec les réseaux de neurones. Elle se concentre sur son ergonomie, sa modularité et son évolutivité. C'est une sous-bibliothèque de TensorFlow.
- Seaborn [28] et Matplotlib [29] : Seaborn est une bibliothèque Python de visualisation de données basée sur matplotlib. Ils fournissent des interfaces de haut niveau pour dessiner des graphiques statistiques attrayants et informatifs, des histogrammes et d'autres formats de visualisation données.
- Surprise [30] : Surprise est une bibliothèque Python facile à utiliser qui nous permet de créer rapidement des systèmes de recommandation. Cela nous donne également accès à des facteurs matriciels lors de l'utilisation de modèles tels que SVD, ce qui nous permet de visualiser les similitudes entre les éléments d'un ensemble de données.
- PyQt [31] : est un ensemble de modules Python pour le framework Qt de C++, qui peuvent être utilisés pour créer des interfaces utilisateur graphiques (GUI) de bureau.

C. Google Colaboratory

C'est un produit de Google Research qui permet de partager des blocs-notes (projets) entre plusieurs utilisateurs qui peuvent écrire et exécuter du code Python à partir d'un navigateur sans avoir à télécharger ou installer un logiciel au préalable. Il s'agit d'un environnement particulièrement adapté à l'apprentissage automatique et à l'analyse de données. La plateforme nous offre d'énormes ressources de stockage mémoire et de vitesse d'exécution pour accélérer la construction de nos modèles complexes, ainsi que nos phases d'expérimentation et d'évaluation. Il est possible de faire le lien entre le notebook et Google Drive ou GitHub pour enregistrer ou importer des fichiers de données.

3.2.2 Présentation de l'application

Afin faciliter la présentation de notre travail, nous avons mis en place une application développée en Python, utilisant PyQt pour la création de la GUI.

Ci-dessous la fenêtre principale de l'application qui comporte une représentation du modèle développé : SB-HDLM.

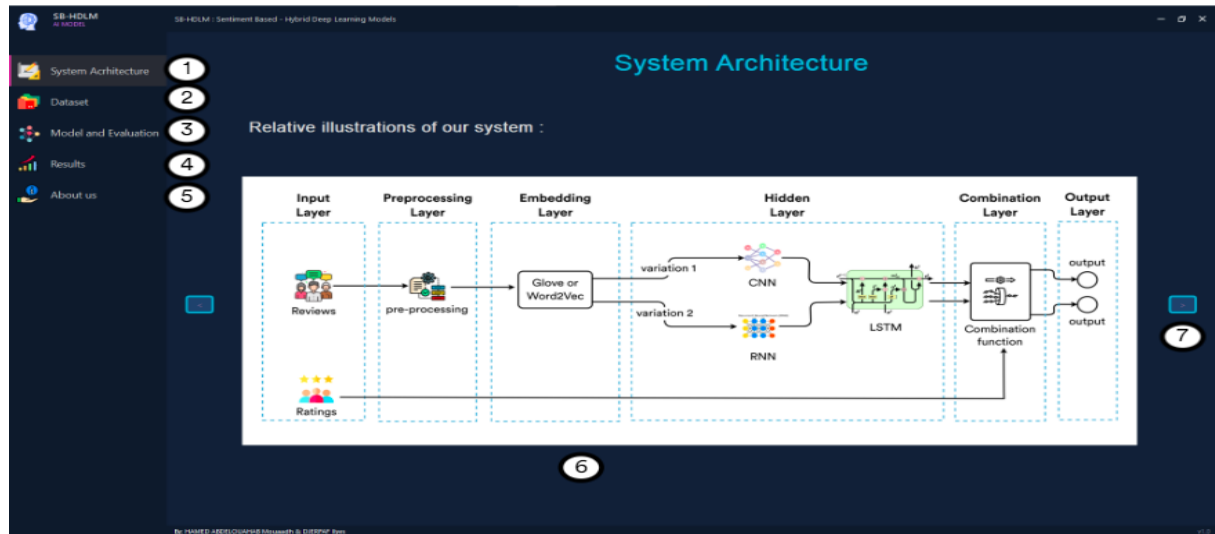


FIGURE 3.1 – Interface de l'application – Fenêtre principale

Description des fonctionnalités décrites dans la figure 3.1 :

1. Présentation de l'architecture de notre système SB-HDLM.
2. Sélection du dataset.
3. Sélection du modèle et type d'évaluation.
4. Affichage des résultats de tests.
5. Informations concernant les développeurs de ce système ainsi que cette interface.
6. Architecture de notre système SB-HDLM.
7. Affichage d'autre illustration du modèle telles que : Visualisation du prétraitement, CNN, etc

Afin de procéder à la sélection du modèle et l'affichage des résultats, il faut d'abord sélectionner un dataset (voir figure 3.2). Deux dataset sont disponibles, il faut donc choisir un.

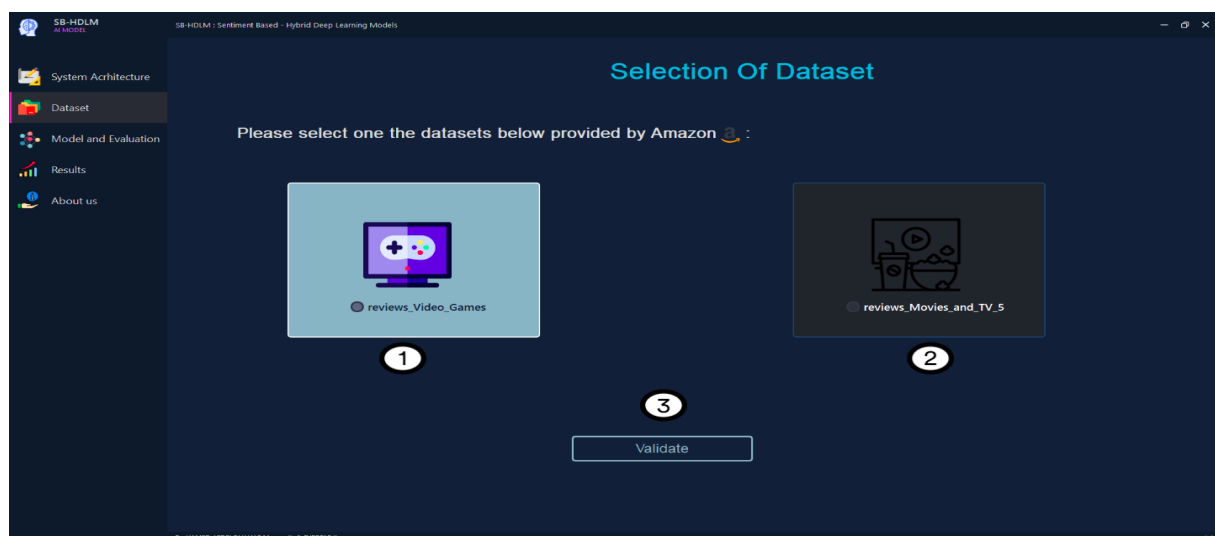


FIGURE 3.2 – Interface de l'application – Sélection du dataset

Description des fonctionnalités décrites dans la figure 3.2 :

1. Sélectionner reviews_Video_Games_5 dataset.
2. Sélectionner reviews_Movies_and_TV_5 dataset.
3. Valider la sélection.

Une fois que le dataset est sélectionné, vient le choix du modèle et les informations propres à l'entraînement de notre système.

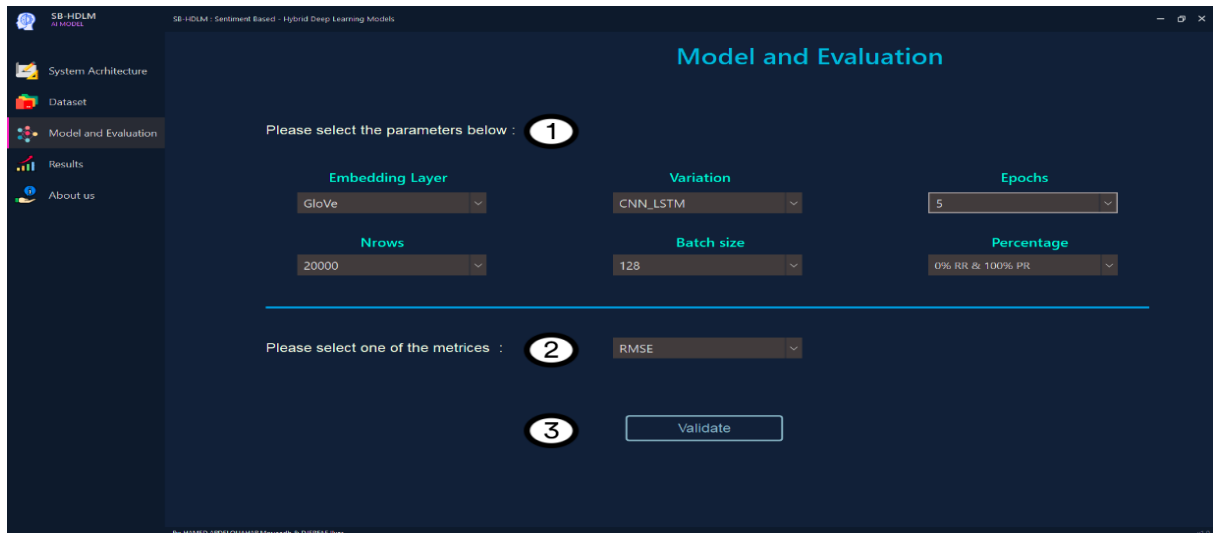


FIGURE 3.3 – Interface de l'application – Sélection du modèle et des métriques

Description des fonctionnalités décrites dans la figure 3.3 :

1. Sélectionner le modèle à évaluer.
2. Sélectionner la métrique d'évaluation souhaitée.
3. Valider la sélection.

3.3 Expérimentations

3.3.1 Datasets

Pour la phase d'entraînement et d'évaluation de notre méthode, nous avons utilisé deux datasets de la base Amazon, qui est la base la plus utilisée dans le domaine de l'e-commerce. Ces datasets sont décrits ci-dessous :

- Amazon Movies and TV : contient 673,342 avis de films d'Amazon [18] ajouté par 14,169 utilisateurs (users) et sur 17,795 items. Chaque avis comprend également des informations sur les produits (productID) et les utilisateurs (userID), des évaluations et des avis en clair. Il couvre une période allant de mai 1996 à juillet 2014.
- Amazon Video Games : contient 231,780 avis/notes de produit des jeux vidéo d'Amazon [18] ajouté par 24,303 utilisateurs (users) et sur 10,627 items. Le même type d'informations qui concernent les utilisateurs et items dans la base Movies and TV sont mis à disposition

Remarque : Les deux datasets sont une base K-cores, de sorte que chacun des utilisateurs et items ait chacun k avis/notes. Dans notre projet, nous utilisons une base 5-cores.

Les statistiques relatives aux jeux de données utilisés sont présentées dans le tableau ci-dessous :

Datasets	#utilisateurs	#items	#notes/avis	#Densité %
Amazon Movies and TV	14169	17795	673342	0.267
Amazon Video Games	24303	10627	231780	0.090

TABLE 3.1 – Statistiques relatives aux jeux de données.

La densité est calculée à partir de la formule suivante :

$$Densité = \frac{\text{nombre d'évaluation contenues dans le dataset}}{\text{nombre d'utilisateurs} \times \text{nombre d'items}} \quad (3.1)$$

3.3.2 Métriques d'évaluation

Pour évaluer les performances de notre système et pour le comparer avec d'autres modèles, nous utilisons les métriques d'évaluation suivantes :

- a. Root Mean Squared Error (RMSE) :** La RMSE est une mesure couramment utilisée pour les problèmes de régression. Elle consiste à calculer la différence entre les résultats prédits par le modèle et les résultats attendus. De ce fait, plus la valeur de cette mesure diminue, plus notre modèle est efficace par ses prédictions. Elle décrit un ajustement parfait aux données lorsqu'il atteint zéro (0). Elle est calculée par la formule suivante :

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (\hat{y}_i - y_i)^2} \quad (3.2)$$

Avec :

- \hat{y}_i : le résultat prédit de l'item i
- y_i : le résultat attendu de l'item i
- n : le nombre total d'évaluations.

- b. Mean Absolute Error (MAE) :** La MAE, ou erreur absolue moyenne, est la moyenne des valeurs absolues des erreurs, définie par la formule :

$$MAE = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} |\hat{y}_i - y_i|} \quad (3.3)$$

Avec :

- \hat{y}_i : le résultat prédit de l'item i
- y_i : le résultat attendu de l'item i
- n : le nombre total d'évaluations.

La MAE est la métrique de régression la plus interprétable, car elle est dans la même unité que la variable à prédire. Cette métrique quantifie l'erreur réalisée par le modèle. Plus elle est élevée, moins le modèle est performant.

3.3.3 Résultats des évaluations

A. Évaluation préliminaires

Dans cette partie, nous présentons les résultats obtenus par notre système SB-HDLM à travers une série d'expérimentations pour déterminer ses paramètres optimaux selon l'utilisation des deux embeddings GloVe et Word2Vec sur les deux différents datasets. Dans chaque embedding, nous varions le nombre d'épochs et la taille du batch pour le deux modèles CNN-LSTM (C_LSTM) et RNN-LSTM (R_LSTM).

a. Utilisation de GloVe embeddings

- **Variation du nombre des epochs :** L'époch est un terme utilisé pour référer à une itération où le modèle voit tout le training set pour mettre à jour ses coefficients (poids).

Au fur et à mesure que le nombre des epochs augmente, le nombre de fois que le poids est modifié dans le réseau neuronal et la courbe du modèle passe d'une courbe de sous-apprentissage (underfitting) à une courbe optimale puis à une courbe de sur-apprentissage (overfitting).

Le tableau suivant illustre la variation du nombre pour chaque modèle sur le dataset de Movies and TV.

Modèle	RMSE	MAE	Graphes d'évaluation
C_LSTM	0.2579	0.1837	
R_LSTM	0.2526	0.1797	

TABLE 3.2 – Évaluation des modèles sur le dataset Movies and TV -GloVe.

Les graphiques de CNN_LSTM et RNN_LSTM montrent que les plus petites valeurs des métriques des données de validation se produisent à l'époch 7 et 10 commençant par 0 respectivement. Après ces points, elle augmente légèrement, cette augmentation est un signe de d'overfitting.

Nous pouvons présenter les résultats obtenus sur les différents datasets avec la variation du nombre des epochs via le tableau ci-dessous :

			Dataset							
			Movies and TV				Video Games			
Nombre d'epochs			5	7	10	20	5	13	15	20
Modèle	CNN LSTM	RMSE	0.2670	0.2579	0.2647	0.2598	0.3019	0.2819	0.2819	0.2825
		MAE	0.1904	0.1837	0.1893	0.1861	0.2217	0.2026	0.2026	0.2051
	RNN LSTM	RMSE	0.2616	0.2638	0.2526	0.2537	0.3019	0.2819	0.2819	0.2857
		MAE	0.1890	0.1886	0.1797	0.1802	0.2217	0.2026	0.2028	0.2051

TABLE 3.3 – Tableau représentant l'évaluation des modèles sur les différents datasets en fonction de la variation du nombre d'epochs -GloVe.

- **Variation de la taille du batch :** La taille du batch ou Batch Size est le nombre d'exemples qui sont montrés au modèle avant que les poids du modèle soient recalculés par chaque epoch. Nous étudions, comme illustré par les figures 3.4, les résultats des modèles selon la variation du Batch Size.

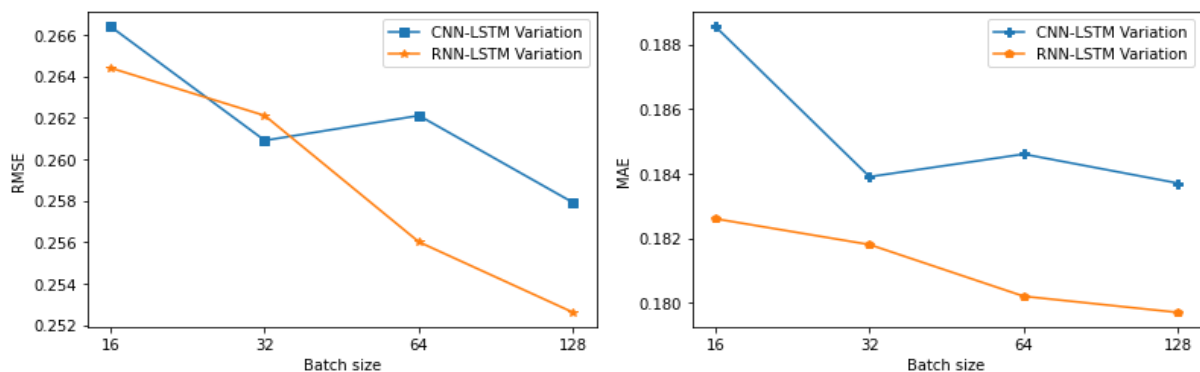


FIGURE 3.4 – Figure représentant l'évaluation des variantes du système SB-HDLM en fonction du batch size sur le dataset Movies and TV -GloVe

Le tableau 3.3 présente les résultats obtenus sur les différents datasets avec une variation de la taille du batch.

			Dataset							
			Movies and TV				Video Games			
Nombre d'épochs			16	32	64	128	16	32	64	128
Modèle	CNN LSTM	RMSE	0.2664	0.2609	0.2621	0.2579	0.2887	0.2866	0.2860	0.2819
		MAE	0.1885	0.1839	0.1846	0.1837	0.2150	0.2115	0.2118	0.2026
	RNN LSTM	RMSE	0.2644	0.2621	0.2560	0.2526	0.3052	0.2880	0.2840	0.2819
		MAE	0.1826	0.1818	0.1802	0.1797	0.2107	0.2075	0.2034	0.2028

TABLE 3.4 – Tableau représentant l'évaluation des modèles sur les différents datasets en fonction de la variation de la taille de batch -GloVe.

Les résultats d'expérimentation illustrés dans le tableau 3.4 démontrent que les deux modèles CNN-LSTM et RNN-LSTM avec un batch size fixé à 128 parvient à réaliser les meilleurs résultats sur les deux datasets.

- b. Utilisation de Word2Vec embeddings :** Pour voir l'impact d'utilisation de Word2Vec, nous expérimentons les mêmes variations du nombre d'épochs et de la taille du batch comme dans la partie précédente

- **Variation du nombre des epochs**

			Dataset							
			Movies and TV				Video Games			
Nombre d'épochs			5	10	11	20	5	6	14	20
Modèle	CNN LSTM	RMSE	0.3410	0.3396	0.3357	0.3871	0.3004	0.2932	0.2944	0.2954
		MAE	0.1893	0.1861	0.2217	0.2427	0.2191	0.2115	0.2120	0.2112
	RNN LSTM	RMSE	0.2941	0.2922	0.2955	0.3167	0.2845	0.2938	0.2820	0.2857
		MAE	0.1965	0.1914	0.1921	0.2103	0.2042	0.2151	0.2007	0.2051

TABLE 3.5 – Tableau représentant l'évaluation des modèles sur les différents datasets en fonction de la variation du nombre des epochs -Word2Vec.

Le tableau 3.5 présente les résultats obtenus sur les différents datasets avec une variation du nombre d'épochs.

- **Variation de la taille du batch :** Les résultats des expérimentations obtenus sur les datasets avec une variation de la taille du batch sont montrés dans le tebleau 3.6.

			Dataset							
			Movies and TV				Video Games			
Batch size			16	32	64	128	16	32	64	128
Modèle	CNN LSTM	RMSE	0.4359	0.3608	0.3386	0.3357	0.3874	0.3521	0.3185	0.2932
		MAE	0.3857	0.2834	0.2485	0.2217	0.4092	0.3584	0.2863	0.2115
	RNN LSTM	RMSE	0.4227	0.3945	0.3231	0.2922	0.3740	0.3433	0.3039	0.2820
		MAE	0.3025	0.2673	0.2136	0.1914	0.3942	0.3458	0.2932	0.2007

TABLE 3.6 – Évaluation des modèles sur les deux datasets en fonction de la variation de la taille du batch -Word2Vec.

- c. **Interprétation des résultats des embeddings :** Les expérimentations précédentes ont montré que Glove a donné de meilleures performances par rapport à Word2Vec. Le tableau suivant présente les résultats de comparaison entre les deux embeddings sur les deux datasets.

		Dataset							
		Movies and TV				Video Games			
Modèle		CNN-LSTM		RNN-LSTM		CNN-LSTM		RNN-LSTM	
Métrique		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
EMBEDD- ING	Glove	0.2579	0.1837	0.2526	0.1797	0.2819	0.2026	0.2819	0.2028
	Word2Vec	0.3357	0.2217	0.2922	0.1914	0.2932	0.2115	0.2820	0.2007

TABLE 3.7 – Comparaisons entre l'utilisation des GloVe et Word2Vec embeddings.

De ce fait, nous prenons en considération les résultats de GloVe embedding pour le reste de nos expérimentations.

B. Évaluation du modèle hybride

- a. **Variation du poids d'importance :** Dans cette section, nous varions le paramètre de la fonction de combinaison « α » qui est un nombre réel entre 0 et 1. La figure ci-dessous montre les expérimentations effectuées :

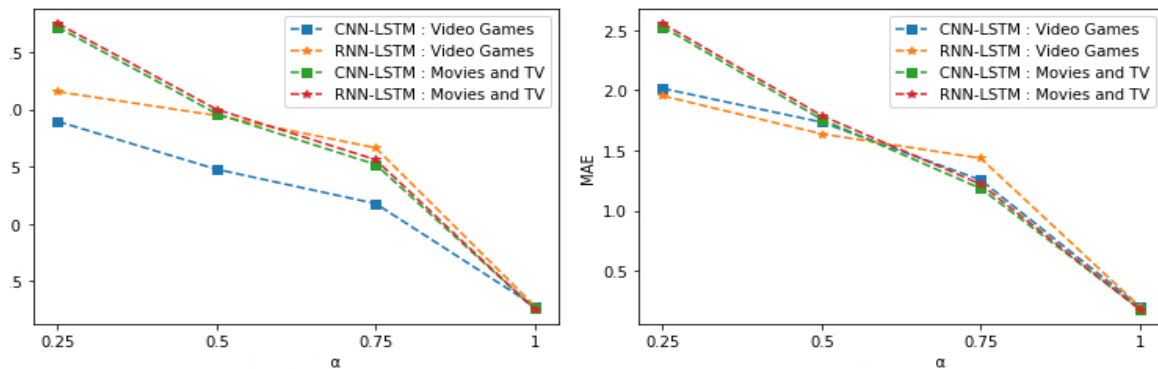


FIGURE 3.5 – Évaluation du modèle hybride avec une variation du poids d'importance α sur les deux datasets

D'après la figure 3.5, nous pouvons remarquer que lorsque α converge vers 0, les modèles génèrent des mauvais résultats. Par contre, lorsque α converge vers 1, les modèles produisent les meilleurs résultats.

- b. **Interprétation des résultats de la variation :** La fonction de combinaison a pour but de combiner, selon un poids d'importance α pour chaque item i , sa note réelle r et sa note p prédite par le modèle d'analyse de sentiments. Les résultats montrent que notre système génère des meilleurs résultats en se basant seulement sur les notes prédites sans prendre en considération la note réelle d'un item donné. (Voir section ??) Ce qui prouve l'efficacité du modèle d'analyse de sentiments.

C. Comparaison avec des modèles de l'état de l'art

Nous avons comparé nos résultats avec trois modèles de l'état de l'art : DeepCoNN [32] qui utilise deux CNNs parallèles pour extraire les facteurs latents des documents d'évaluation des items et des utilisateurs ; NARRE [33] qui adopte un mécanisme d'attention neuronale pour construire des vecteurs de caractéristiques latentes des items et des utilisateurs en sélectionnant les commentaires les plus utiles ; et PMF [34] qui construit les facteurs latents des items et des utilisateurs en adoptant une distribution gaussienne.

Tous les modèles sont testés sur les deux datasets Amazon Video Games et Amazon Movies and TV. Les comparaisons des résultats de la prédiction des notes sont résumées dans le tableau 3.8. Les meilleurs résultats sont mis en évidence en gras.

Modèles	Amazon Movies and TV		Amazon Video Games	
	RMSE	MAE	RMSE	MAE
DeepCoNN [32]	1.0096	0.7323	1.0706	0.8050
NARRE [33]	0.9947	0.7162	1.0607	0.7938
PMF [34]	1.0428	0.7878	1.3965	1.0981
SB-HDLM(C-LSTM)	0.2579	0.1837	0.2819	0.2026
SB-HDLM(R-LSTM)	0.2526	0.1797	0.2819	0.2028

TABLE 3.8 – Comparaison des résultats des prédictions des modèles sur les deux datasets.

3.4 Discussion générale

L'évaluation du modèle proposé SB-HDLM a pour objectif de voir l'impact d'utilisation de word embedding GloVe ou Word2Vec sur l'analyse de sentiments. Les résultats des expérimentations résumés dans le tableau 3.7 montrent que GloVe a généré de meilleures performances par rapport à Word2Vec pour les deux datasets vu que, contrairement à Word2vec, GloVe ne s'appuie pas uniquement sur les informations sur le contexte local des mots, mais intègre les cooccurrences des mots pour obtenir les vecteurs de mots.

Les expérimentations effectuées nous ont permis de fixer les paramètres optimaux pour les deux variantes CNN-LSTM et RNN-LSTM, en faisant varier le nombre d'epochs et la taille du batch. Cela étant un critère essentiel à une bonne prédiction. Les résultats obtenus ont montré que le modèle RNN-LSTM a été plus performant que le modèle CNN-LSTM en termes des métriques MAE et RMSE. Ce résultat peut être justifié par le fait que les réseaux RNN sont formés pour reconnaître des patterns au fil du temps, et ils sont plus adaptés aux données textuelles qui se présentent en séquence. Par contre, le réseau CNN est un réseau neuronal à action directe, généralement utilisé pour la reconnaissance d'images et la classification d'objets, qui prend des entrées de tailles fixes et génère des sorties de tailles fixes.

Notre approche présente l'avantage de pouvoir opérer une analyse de sentiments sans avoir à être supervisée par les notes des utilisateurs. Elle peut être appliquée sur des datasets où les notes ne sont pas renseignées par les utilisateurs. Ce qui permet de résoudre le problème de sparsity (rareté des données dont souffrent les systèmes de recommandation)

Les tests précédents sur les deux datasets, démontrent que nos modèles surpassent plusieurs modèles de l'état de l'art. Cependant, il serait intéressant de tester nos modèles sur d'autres datasets avec une densité plus élevée. De plus, par faute de temps, nous n'avons considéré qu'une

variation du nombre d'époques et à la taille du batch. Afin d'obtenir de meilleurs résultats, il serait intéressant de faire varier d'autres paramètres tels que : le dropout, learning rate, les fonctions de coût et d'optimisation, le nombre de couches neuronale, etc.

Des améliorations supplémentaires peuvent être mises en place en considérant les Fake Reviews (fausses évaluations), détectant ces derniers par une matrice de confiance tel que l'approche proposée par CARM [35] ou en appliquant un mécanisme d'attention neuronale comme NARRE [33].

3.5 Conclusion

Dans ce chapitre, nous avons défini les aspects techniques liés au développement et à l'expérimentation de notre approche de recommandation basée sur l'analyse de sentiments et l'utilisation des techniques d'apprentissages profonds. Les résultats des évaluations ont montré que notre modèle a donné de meilleures performances par rapport aux travaux de l'état de l'art.

Conclusion générale

Nous nous sommes intéressés dans le présent projet de fin d'études de Licence à la problématique de rareté des évaluations des utilisateurs sur les items, ce qui affecte les performances des systèmes de recommandation.

S'inspirant de quelques travaux existants sur l'analyse de sentiments, nous avons proposé une approche appelée SB-HDLM (Sentiment Based - Hybrid Deep Learning Models), qui combine les évaluations données par l'utilisateur avec les notes prédites par analyse de sentiments selon une formule pondérée. Le module d'analyse de sentiments se base sur deux variations des modèles d'apprentissage profond hybrides, CNN-LSTM et RNN-LSTM, où nous avons utilisé deux word embedding Word2Vec et GloVe.

Les résultats des expérimentations de notre approche sur deux datasets du domaine e-commerce « Amazon Movies and TV » et « Amazon Video Games », ont montré que GloVe a généré de meilleures performances par rapport à Word2Vec et que le modèle RNN-LSTM a été plus performant que le modèle CNN-LSTM. De plus, notre modèle a surpassé plusieurs modèles existants, DeepCoNN [32], NARRE [33] et PMF [34] en termes des métriques d'évaluation MAE et RMSE.

Les résultats obtenus nous motivent à enrichir ce travail en testant nos modèles sur d'autres datasets avec une densité plus élevée, faire varier d'autres paramètres (dropout, learning rate, fonctions de coût et d'optimisation, nombre de couches neuronale, etc.), considérer une analyse de sentiments en prenant en charge les Fake Reviews (fausses évaluations).

Bibliographie

- [1] Dang CN, Moreno-García MN et Prieta FD. “An Approach to Integrating Sentiment Analysis into Recommender Systems”. In : *Sensors* 21.16 (2021), p. 5666.
- [2] Frias-Martinez E, Chen SY et Liu X. “Evaluation of a personalized digital library based on cognitive styles : Adaptivity vs. adaptability”. In : *International Journal of Information Management* 29.1 (2009), p. 48-56.
- [3] Cho YH et Kim JK. “Application of Web usage mining and product taxonomy to collaborative recommendations in e-commerce”. In : *Expert systems with Applications* 26.2 (2004), p. 233-246.
- [4] Kumar S et Balakrishnan K. “Development of a Model Recommender System for Agriculture Using Apriori Algorithm”. In : *Cognitive Informatics and Soft Computing*. Springer, 2019, p. 153-163.
- [5] Sahoo AK, Pradhan C, Barik RK et Dubey H. “DeepReco : deep learning based health recommender system using collaborative filtering”. In : *Computation* 7.2 (2019), p. 25.
- [6] Tom MITCHELL. *Machine learning*. T. 1. 9. McGraw-hill New York, 1997.
- [7] El Naqa I et Murphy MJ. “What is machine learning ?” In : *machine learning in radiation oncology*. Springer, 2015, p. 3-11.
- [8] Nielsen MA. *Neural networks and deep learning*. T. 25. Determination press San Francisco, CA, USA, 2015.
- [9] <https://www.knime.com/blog/a-friendly-introduction-to-deep-neural-networks>, site consulté en février,2022.
- [10] <https://www.securityinfowatch.com/videosurveillance/videoanalytics/article/21069937/deep-learning-to-the-rescue>, site consulté en mars,2022.
- [11] Ni J, Huang Z, Cheng J et Gao S. “An effective recommendation model based on deep representation learning”. In : *Information Sciences* 542 (2021), p. 324-342.
- [12] <https://www.mdpi.com/2076-3417/9/24/5462/html>, site consulté en mai,2022.
- [13] Asani E, Vahdat-Nejad H et Sadri J. “Restaurant recommender system based on sentiment analysis”. In : *Machine Learning with Applications* 6 (2021), p. 100-114.
- [14] Karthik RV et Ganapathy S. “A fuzzy recommendation system for predicting the customers interests using sentiment analysis and ontology in e-commerce”. In : *Applied Soft Computing* 108 (2021), p. 107-396.
- [15] Abbasi-Moud Z, Vahdat-Nejad H et Sadri J. “Tourism recommendation system based on semantic clustering and sentiment analysis”. In : *Expert Systems with Applications* 167 (2021), p. 114-324.
- [16] Liu D, Li J, Du B, Chang J, Gao R et Wu Y. “A hybrid neural network approach to combine textual information and rating information for item recommendation”. In : *Knowledge and Information Systems* 63.3 (2021), p. 621-646.

- [17] Alsentzer E, Murphy JR, Boag W, Weng WH, Jin D, Naumann T et McDermott M. "Publicly available clinical BERT embeddings". In : *arXiv preprint arXiv :1904.03323* (2019).
- [18] <http://jmcauley.ucsd.edu/data/amazon/>, site consulté en mai,2022.
- [19] <https://www.geeksforgeeks.org/removing-stop-words-nltk-python/>, site consulté en mai,2022.
- [20] Al-Saqqa S et Awajan A. "The use of word2vec model in sentiment analysis : A survey". In : *Proceedings of the 2019 International Conference on Artificial Intelligence, Robotics and Control* (2019), p. 39-43.
- [21] Kingma DP et Ba J. ADAM. "Adam : A method for stochastic optimization". In : *arXiv preprint arXiv :1412.6980* (2014).
- [22] <https://numpy.org/>, site consulté en mai,2022.
- [23] <https://pandas.pydata.org/>, site consulté en mai,2022.
- [24] <https://www.nltk.org/>, site consulté en mai,2022.
- [25] <https://radimrehurek.com/gensim>, site consulté en mai,2022.
- [26] <https://scikit-learn.org/>, site consulté en mai,2022.
- [27] <https://keras.io/>, site consulté en mai,2022.
- [28] <https://seaborn.pydata.org/> site consulté en mai,2022.
- [29] <https://matplotlib.org/>, site consulté en mai,2022.
- [30] <http://surpriselib.com/>, site consulté en mai,2022.
- [31] <https://pythonpyqt.com/>, site consulté en mai,2022.
- [32] Zheng L, Noroozi V et Yu PS. "Joint deep modeling of users and items using reviews for recommendation". In : *Proceedings of the tenth ACM international conference on web search and data mining*. 2017, p. 425-434.
- [33] Chen C, Zhang M, Liu Y et Ma S. "Neural attentional rating regression with review-level explanations". In : *Proceedings of the 2018 World Wide Web Conference*. 2018, p. 1583-1592.
- [34] Mnih A et Salakhutdinov R. *Probabilistic matrix factorization. in advances in neural information processing systems*. 2007.
- [35] Li D, Liu H, Zhang Z, Lin K, Fang S, Li Z et Xiong NN. "CARM : Confidence-aware recommender model via review representation learning and historical rating behavior in the online platforms". In : *Neurocomputing* 455 (2021), p. 283-296.